Monatsh Chem 139, 197–211 (2008)
DOI 10.1007/s00706-007-0773-4
Printed in The Netherlands

Monatshefte für Chemie
Chemical Monthly

Feature Set Selection in QSAR of 1-[(2-Hydroxyethoxy)methyl]-6-(phenylthio)thymine (*HEPT*) Analogues by Using Swarm Intelligence

Chakguy Prakasvudhisarn[1,*] and **Luckhana Lawtrakul**[2]

[1] School of Technology, Shinawatra University, Bangkok, Thailand
[2] Sirindhorn International Institute of Technology (SIIT), Thammasat University, Pathum Thani, Thailand

Received May 8, 2007; accepted July 23, 2007; published online February 12, 2008

Summary. A binary particle swarm optimization (PSO) is adopted for major influence descriptors selection in quantitative structure-activity relationship (QSAR) of a large data set of 1-[(2-Hydroxyethoxy)methyl]-6-(phenylthio)thymine (*HEPT*) analogues due to its simplicity, speed, and consistency. It has an embedded mechanism, which is based on social behavior of sharing information within a group and experience of each particle that could reach a near-optimal solution. The modified particle swarm optimization is then combined with the neural networks (NNs) for its universal approximating property to generate a QSAR model with the selected features. Since the quality of chosen features also depend on the reliability of the QSAR model, two effective and efficient algorithms, the *Levenberg-Marquardt* backpropagation (PSO–*LM*BP) and the PSO (PSO–PSO), were used instead of the classical backpropagation (gradient descent method). The *Pearson* correlation is employed as the fitness function for the predictive ability of the obtained QSAR model from the selected features. Experimental results reveal that the PSO is a useful tool for feature set selection in QSAR of a large data set of *HEPT* analogues.

Keywords. QSAR; *HEPT*; Feature set selection; Particle swarm optimization; Neural networks.

Introduction

Quantitative structure-activity relationship (QSAR) studies provide interpretability between the physicochemical properties and biological activity of compounds. Chemical structure is represented by a wide variety of descriptors such as mass, surface area, volume, dipoles, molar refractivity, lipophilicity, *Verloop* parameters, connectivity, shape indices, counts of atoms, rings, groups, hydrogen bond donors and acceptors, and electrostatic parameters. There could be quite a number, as many as hundreds or even thousands, of descriptors in QSAR studies. However, the accuracy of the regression model obtained does not monotonically depend on the number of descriptors or features employed. In addition, the number of compounds available with biological values is relatively small compared to the number of these descriptors. Moreover, the presence of irrelevant or redundant features can cause interference with the key relationships that are essential for generalization of the model for unseen test sets. Consequently, the resultant model may be unstable and cannot describe meaningful relationships between the descriptors and the biological activity. The so-called feature selection process would alleviate this situation by identifying a small subset of influencing factors. They can then be used to construct model representing relationships between the chemical parameters and the biological activity. Therefore, the feature selection process would play an important role for the development of a reliable QSAR model. This would, in turn, lead to new drug designs.

Various techniques such as multiple linear regressions (MLR), partial least squares (PLS), and neural networks (NNs) have been used to develop QSAR models with excellent results [1–5]. The application

* Corresponding author. E-mail: chakguy@shinawatra.ac.th

of a novel technique like NNs has gained wide acceptance in QSAR studies. Multilayer feed-forward neural networks have shown to produce impressively effective QSAR models, especially for capturing nonlinear relationships between descriptors and activity [6]. Many researchers have studied QSAR of 1-[(2-hydroxyethoxy)methyl]-6-(phenylthio)thymine (*HEPT*) analogues, a non-nucleoside HIV-1 reverse transcriptase inhibitor (NNRTI) [7], by using a wide variety of statistical techniques and computational intelligence methods [8–19], because the available number of known biological activity compounds and the inhibition mechanism of this NNRTI were clearly identified. In this QSAR study, however, NNs are used to map the key features to the biological activity due to their excellent function approximation, regardless of distribution assumptions on data. A large set of *HEPT* is investigated. As a result, the constructions of QSAR models are evaluated.

The major shortcoming of these regression techniques, that has been largely ignored, is that they can never reach their full potential, no matter how good they are, if they are applied to poor data collected by a data measurement process and a variable selection process. The former normally depends on the operator's skill and the reliability of the instrument used. These can be taken care of rather easily *via* proper training and calibration. The latter, on the other hand, requires deeper understanding of chemical structures, their correlations, and statistics to suitably select key parameters. This widely used manual procedure is very time-consuming and laborious, especially when a large dataset is to be tackled [8, 10, 11].

Recently, some effective and efficient heuristic searches such as simulated annealing (SA), genetic algorithms (GAs), and particle swarm optimization (PSO) have been investigated for feature set selection [1, 20–24]. They provide a means to automatically select influential features without the need of the user interference. The selection process must be formulated as an optimization problem. Excellent outcomes can be obtained because feature set selection can be viewed as a search procedure where each state in the search space represents a particular subset of the available features. The selected descriptors are then used as inputs in the QSAR modeling. As opposed to these heuristic searches, an exhaustive search of the state space is possible, but not practical, since huge possible combinations are computationally expensive too. In summary, not only a good modeling is desired in developing a reliable QSAR model, but an effective and efficient method for key features identification of QSAR study is also desired over a common practice of manually applying chemical structure correlations and trial and error experiment, and a computationally exhaustive search. This method would speed up the development of a reliable QSAR model. Therefore, the objective of this research is to investigate an integrative approach of both issues combined by applying PSO to feature set selection and NNs to QSAR modeling.

Theories and Methodologies

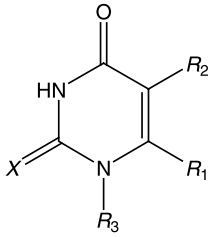
HEPT Data Set and Descriptors Generation

The chemical structures of *HEPT* derivatives are given in Table 1, together with their biological activity which has been defined as $\log(1/EC_{50})$, and is used as the dependent variable in QSAR models. EC_{50} is the effective molar concentration of compound required to achieve 50% protection of MT-4 cells against the cytopathic effect of HIV-1. These values are taken from Refs. [25–28]. The starting geometry of *HEPT* is taken from *HEPT*/HIV-1 RT X-ray structure complex (PDB Code: 1RTI [29]). Other *HEPT* analogues are constructed by modifying this *HEPT* crystal structure using the WebLab Viewer program [30]. After full geometric optimizations at *HF*/3-21G level by the Gaussian03 program [31], the atomic charges from *HF* calculations are used as the electronic parameters in QSAR studies. Structure properties, topological, connectivity, shape indices, and counting molecular features are calculated by the TSAR program package [32]. The values of some particular parameters are not available for many of the substituents or molecules in a given data set. This can be handled by replacing them with an available set of strongly correlated descriptors. These forty selected descriptors are provided in Table 2. The *HEPT* data set consists of 110 compounds (**1–110**) for training the models and 22 compounds (**111–132**) for testing the quality of models.

PSO for Feature Selections

Particle swarm optimization (PSO) is a relatively new optimization method for continuous nonlinear functions. It has been introduced in the framework of a simple artificial social model such as bird

Table 1. Chemical structure and biological activity of the *HEPT* data set



cmp.	X	R_1	R_2	R_3	$\log(1/EC_{50})$
1	O	S*Ph*	*Me*	$CH_2OCH_2CH_2OH$	5.1549
2	O	*Ph*	*Me*	$CH_2OCH_2CH_2OH$	3.7799
3	O	CH_2Ph	*Et*	$CH_2OCH_2CH_2OH$	6.4559
4	O	CH_2Ph(3,5-Me_2)	*Et*	$CH_2OCH_2CH_2OH$	7.8861
5	O	CH_2Ph	*Et*	CH_2OEt	7.3872
6	O	CH_2Ph(3,5-Me_2)	*Et*	CH_2OEt	8.7959
7	O	CH_2Ph	*i-Pr*	$CH_2OCH_2CH_2OH$	7.2007
8	O	CH_2Ph(3,5-Ph_2)	*i-Pr*	$CH_2OCH_2CH_2OH$	8.5686
9	O	CH_2Ph	*i-Pr*	CH_2OEt	8.3768
10	O	CH_2Ph(3,5-Me_2)	*i-Pr*	CH_2OEt	9.2218
11	O	CH_2Ph	*i-Pr*	*Bu*	7.3768
12	O	CH_2Ph	*Et*	CH_2CH_2OMe	6.6021
13	O	CH_2Ph	*Ph*	$CH_2OCH_2CH_2OH$	4.6383
14	O	S*Ph*	*Me*	$CH_2OCH_2CH_2OMe$	5.0605
15	O	S*Ph*	*Me*	$CH_2OCH_2CH_2OC_5H_{11}$	4.2596
16	O	S*Ph*	*Me*	$CH_2OCH_2CH_2OCH_2Ph$	4.6990
17	O	S*Ph*	*Me*	CH_2OMe	5.6778
18	O	S*Ph*	*Me*	CH_2OEt	6.4815
19	O	S*Ph*	*Me*	CH_2OBu	5.3279
20	O	S*Ph*	*Me*	CH_2OCH_2Ph	7.0555
21	S	S*Ph*	*Et*	CH_2OEt	7.5850
22	S	S*Ph*(3,5-Me_2)	*Et*	CH_2OEt	8.3565
23	S	S*Ph*(3,5-Cl_2)	*Et*	CH_2OEt	7.8861
24	S	S*Ph*	*Et*	CH_2O-*i-Pr*	6.6576
25	S	S*Ph*	*Et*	CH_2O-*c-Hex*	5.7959
26	S	S*Ph*	*Et*	CH_2OCH_2-*c-Hex*	6.4559
27	S	S*Ph*	*Et*	CH_2OCH_2Ph	8.1079
28	S	S*Ph*(3,5-Me_2)	*Et*	CH_2OCH_2Ph	8.1612
29	S	S*Ph*	*Et*	$CH_2OCH_2C_6H_4$(4-*Me*)	7.1079
30	S	S*Ph*	*i-Pr*	CH_2OEt	7.8539
31	S	S*Ph*	*i-Pr*	CH_2OCH_2Ph	8.1675
32	S	S*Ph*	*c-Pr*	CH_2OEt	7.0223
33	O	S*Ph*	*Et*	CH_2OEt	7.7212
34	O	S*Ph*(3,5-Me_2)	*Et*	CH_2OEt	8.2676
35	O	S*Ph*(3,5-Cl_2)	*Et*	CH_2OEt	8.1308
36	O	S*Ph*	*Et*	CH_2O-*c-Hex*	5.3979
37	O	S*Ph*	*Et*	CH_2OCH_2-*c-Hex*	6.3468
38	O	S*Ph*(3,5-Me_2)	*Et*	CH_2OCH_2Ph	8.4949
39	O	S*Ph*	*Et*	$CH_2OCH_2CH_2Ph$	7.0177
40	O	S*Ph*	*i-Pr*	CH_2OEt	7.9208
41	O	S*Ph*	*i-Pr*	CH_2OCH_2Ph	8.5686
42	O	S*Ph*	*c-Pr*	CH_2OEt	7.0000
43	O	S*Ph*	*Me*	H	3.6021
44	O	S*Ph*	*Me*	*Me*	3.8239
45	O	S*Ph*	*Me*	*Et*	5.6576

(*continued*)

Table 1 (*continued*)

cmp.	*X*	R_1	R_2	R_3	$\log(1/EC_{50})$
46	O	S*Ph*	*Me*	*Bu*	5.9208
47	O	S*Ph*(2-*Me*)	*Me*	$CH_2OCH_2CH_2OH$	4.1487
48	O	S*Ph*(2-NO_2)	*Me*	$CH_2OCH_2CH_2OH$	3.8539
49	O	S*Ph*(2-O*Me*)	*Me*	$CH_2OCH_2CH_2OH$	4.7212
50	O	S*Ph*(3-*Me*)	*Me*	$CH_2OCH_2CH_2OH$	5.5850
51	O	S*Ph*(3-*Et*)	*Me*	$CH_2OCH_2CH_2OH$	5.5686
52	O	S*Ph*(3-*t-Bu*)	*Me*	$CH_2OCH_2CH_2OH$	4.9208
53	O	S*Ph*(3-CH_2OH)	*Me*	$CH_2OCH_2CH_2OH$	3.5346
54	O	S*Ph*(3-F)	*Me*	$CH_2OCH_2CH_2OH$	5.4815
55	O	S*Ph*(3-Cl)	*Me*	$CH_2OCH_2CH_2OH$	4.8861
56	O	S*Ph*(3-Br)	*Me*	$CH_2OCH_2CH_2OH$	5.2441
57	O	S*Ph*(3-NO_2)	*Me*	$CH_2OCH_2CH_2OH$	4.4685
58	O	S*Ph*(3-OH)	*Me*	$CH_2OCH_2CH_2OH$	4.0862
59	O	S*Ph*(3-O*Me*)	*Me*	$CH_2OCH_2CH_2OH$	4.6576
60	O	S*Ph*(4-*Me*)	*Me*	$CH_2OCH_2CH_2OH$	3.6576
61	O	S*Ph*(4-F)	*Me*	$CH_2OCH_2CH_2OH$	3.6021
62	O	S*Ph*(4-Cl)	*Me*	$CH_2OCH_2CH_2OH$	3.6021
63	O	S*Ph*(4-CN)	*Me*	$CH_2OCH_2CH_2OH$	3.6021
64	O	S*Ph*(4-OH)	*Me*	$CH_2OCH_2CH_2OH$	3.5575
65	O	S*Ph*(3,5-Me_2)	*Me*	$CH_2OCH_2CH_2OH$	6.5850
66	O	S*Ph*(3,5-Cl_2)	*Me*	$CH_2OCH_2CH_2OH$	5.8861
67	S	S*Ph*(3,5-Me_2)	*Me*	$CH_2OCH_2CH_2OH$	6.6576
68	O	S*Ph*(3-COO*Me*)	*Me*	$CH_2OCH_2CH_2OH$	5.1024
69	O	S*Ph*(3-CO*Me*)	*Me*	$CH_2OCH_2CH_2OH$	5.1367
70	O	S*Ph*(4-CO*Me*)	*Me*	$CH_2OCH_2CH_2OH$	3.9586
71	O	S*Ph*(3-$CONH_2$)	*Me*	$CH_2OCH_2CH_2OH$	3.5143
72	O	S*Ph*	CONH*Ph*	$CH_2OCH_2CH_2OH$	4.7447
73	S	S*Ph*	*Et*	$CH_2OCH_2CH_2OH$	6.9586
74	S	S*Ph*	*Pr*	$CH_2OCH_2CH_2OH$	5.0000
75	S	S*Ph*	*i-Pr*	$CH_2OCH_2CH_2OH$	7.2291
76	S	S*Ph*(3,5-Me_2)	*Et*	$CH_2OCH_2CH_2OH$	8.1079
77	S	S*Ph*(3,5-Me_2)	*i-Pr*	$CH_2OCH_2CH_2OH$	8.3010
78	S	S*Ph*(3,5-Cl_2)	*Et*	$CH_2OCH_2CH_2OH$	7.3665
79	O	S*Ph*	*Et*	$CH_2OCH_2CH_2OH$	6.9208
80	O	S*Ph*	*Pr*	$CH_2OCH_2CH_2OH$	5.4685
81	O	S*Ph*	*i-Pr*	$CH_2OCH_2CH_2OH$	7.2007
82	O	S*Ph*(3,5-Me_2)	*Et*	$CH_2OCH_2CH_2OH$	7.8861
83	S	S*Ph*	*Me*	$CH_2OCH_2CH_2OH$	6.0088
84	O	S*Me*	*Me*	$CH_2OCH_2CH_2OH$	3.6021
85	O	S*Et*	*Me*	$CH_2OCH_2CH_2OH$	3.6021
86	O	S*Bu*	*Me*	$CH_2OCH_2CH_2OH$	3.8861
87	O	SC_6H_{11}	*Me*	$CH_2OCH_2CH_2OH$	5.0862
88	O	S*Bu*-t	*Me*	$CH_2OCH_2CH_2OH$	3.4365
89	O	O*Me*	*Me*	$CH_2OCH_2CH_2OH$	3.3152
90	O	OC_6H_{11}	*Me*	$CH_2OCH_2CH_2OH$	3.3979
91	O	O*Ph*	*Me*	$CH_2OCH_2CH_2OH$	4.0706
92	O	NHC_6H_{11}	*Me*	$CH_2OCH_2CH_2OH$	3.4237
93	O	NH*Ph*	*Me*	$CH_2OCH_2CH_2OH$	3.4855
94	O	Br	*Me*	$CH_2OCH_2CH_2OH$	3.7447
95	O	Cl	*Me*	$CH_2OCH_2CH_2OH$	3.6021
96	O	CO*Ph*	*Me*	$CH_2OCH_2CH_2OH$	3.4365
97	O	CH(OH)*Ph*	*Me*	$CH_2OCH_2CH_2OH$	3.3979
98	O	CH_2Ph	*Me*	$CH_2OCH_2CH_2OH$	4.6383
99	O	C≡C*Me*	*Me*	$CH_2OCH_2CH_2OH$	3.6021

(*continued*)

Table 1 (*continued*)

cmp.	X	R_1	R_2	R_3	$\log(1/EC_{50})$
100	O	C≡CH	*Me*	$CH_2OCH_2CH_2OH$	5.2596
101	O	CH=CH*Me* (*Z*)	*Me*	$CH_2OCH_2CH_2OH$	3.6021
102	O	$CH=CH_2$	*Me*	$CH_2OCH_2CH_2OH$	3.6021
103	O	S*Ph*	$COCHMe_2$	$CH_2OCH_2CH_2OH$	4.9208
104	O	S*Ph*	CO*Ph*	$CH_2OCH_2CH_2OH$	4.8861
105	O	S*Ph*	CH_2Ph	$CH_2OCH_2CH_2OH$	4.6383
106	O	S*Ph*	$CH=CPh_2$	$CH_2OCH_2CH_2OH$	6.0757
107	O	S*Ph*	C≡C*Me*	$CH_2OCH_2CH_2OH$	4.7212
108	O	S*Ph*	C≡C*Ph*	$CH_2OCH_2CH_2OH$	5.4685
109	O	S*Ph*	C≡CH	$CH_2OCH_2CH_2OH$	4.7447
110	O	S*Ph*	CH=CH*Ph* (*Z*)	$CH_2OCH_2CH_2OH$	5.2218
111	O	CH_2CH_2Ph	*Me*	$CH_2OCH_2CH_2OH$	3.3526
112	O	CH_2Ph	*Et*	*Bu*	6.6778
113	O	CH_2Ph	*i-Pr*	CH_2CH_2OMe	7.2840
114	O	S*Ph*	*Me*	CH_2OPr	5.4437
115	S	S*Ph*	*Et*	$CH_2OCH_2C_6H_4$(4-Cl)	7.9208
116	S	S*Ph*	*Et*	$CH_2OCH_2CH_2Ph$	7.0410
117	O	S*Ph*	*Et*	CH_2O-*i-Pr*	6.4685
118	O	S*Ph*	*Et*	CH_2OCH_2Ph	8.2291
119	O	S*Ph*(2-Cl)	*Me*	$CH_2OCH_2CH_2OH$	3.8861
120	O	S*Ph*(3-CF_3)	*Me*	$CH_2OCH_2CH_2OH$	4.3468
121	O	S*Ph*(4-NO_2)	*Me*	$CH_2OCH_2CH_2OH$	3.7212
122	O	S*Ph*(4-O*Me*)	*Me*	$CH_2OCH_2CH_2OH$	3.6021
123	O	S*Ph*(3-COOH)	*Me*	$CH_2OCH_2CH_2OH$	3.4535
124	O	S*Ph*(3-CN)	*Me*	$CH_2OCH_2CH_2OH$	5.0000
125	O	S*Ph*	$CH_2CH=CH_2$	$CH_2OCH_2CH_2OH$	5.6021
126	O	S*Ph*	COO*Me*	$CH_2OCH_2CH_2OH$	5.1805
127	O	S*Ph*(3,5-Me_2)	*i-Pr*	$CH_2OCH_2CH_2OH$	8.5686
128	O	S*Ph*(3,5-Cl_2)	*Et*	$CH_2OCH_2CH_2OH$	7.8539
129	O	C≡C*Ph*	*Me*	$CH_2OCH_2CH_2OH$	4.8539
130	O	CH=CH*Ph* (*Z*)	*Me*	$CH_2OCH_2CH_2OH$	3.6021
131	O	S*Ph*	S*Ph*	$CH_2OCH_2CH_2OH$	4.6778
132	O	S*Ph*	$CH=CH_2$	$CH_2OCH_2CH_2OH$	5.9586

flocking, fish schooling, and swarming theory [33]. It exhibits some population based stochastic evolutionary computation and falls somewhere between a genetic algorithm and evolutionary programming. The main advantages of the PSO are that its function does not have to be twice differentiable and it has embedded mechanism that is quite robust to local optima. Further, it has demonstrated superior performance in terms of convergence rate and consistent results, since it can be realized in a small computer program using only primitive mathematical operators [34].

To search for optima, the PSO is initialized with a population or swarm of random solutions called particles. Each particle can keep track of its coordinate which is associated with the stored best value, called *pbest*, it has achieved thus far. The overall best value, *gbest*, and its location obtained so far, by any particle, are also recorded. In other words, each individual remembers its own experience (*pbest*) and also knows the group's publicized knowledge (*gbest*). The particles are then assigned changing velocities to fly through hyperspace with a mechanism to avoid local optima by considering its momentum and the best solutions (*pbest* and *gbest*) obtained in each iteration. The standard PSO consists of the following steps [35]:

1. Initialize a population of I particles with random positions and velocities on D dimensions.
2. Evaluate the desired optimization function in D variables for each particle.

Table 2. Forty descriptors in the QSAR models

Feature	Abbreviation	Descriptor	High correlation coefficient features (>0.8)
1	CN1	Atomic charge of N_1	
2	CC4	Atomic charge of C_4	
3	CC5	Atomic charge of C_5	
4	CR1	Atomic charge of R_1	19
5	CR2	Atomic charge of R_2	
6	CR3	Atomic charge of R_3	
7	MOV	Molecular volume	8, 11, 13, 15, 31, 35–40
8	MOR	Molecular refractivity	7, 9, 11, 13, 31, 35–40
9	LIP	log P	8
10	TLI	Total lipole	
11	TRI	*Randic* topological index	7, 13, 31, 35–40
12	TBI	*Balaban* topological index	
13	TWI	Wiener topological index	7, 11, 31, 34–40
14	ESI	Sum of E-state indices	
15	SFI	Shape flexibility index	7, 31, 35–40
16	VL1	*Verloop* L (R_1)	32
17	VL2	*Verloop* L (R_2)	
18	VL3	*Verloop* L (R_3)	
19	B11	*Verloop* B1 (R_1)	4
20	B12	*Verloop* B1 (R_2)	23
21	B13	*Verloop* B1 (R_3)	
22	B21	*Verloop* B2 (R_1)	
23	B22	*Verloop* B2 (R_2)	20
24	B23	*Verloop* B2 (R_3)	
25	B31	*Verloop* B3 (R_1)	
26	B32	*Verloop* B3 (R_2)	
27	B33	*Verloop* B3 (R_3)	34
28	B51	*Verloop* B5 (R_1)	
29	B52	*Verloop* B5 (R_2)	
30	B53	*Verloop* B5 (R_3)	34
31	KCM	*Kier* Chi0 (atoms) index	7–8, 11, 13, 15, 35–40
32	KC1	*Kier* Chi0 (atoms) index (R_1)	16
33	KC2	*Kier* Chi0 (atoms) index (R_2)	
34	KC3	*Kier* Chi0 (atoms) index (R_3)	7, 13, 27, 30
35	KI1	Kappa1 index	8, 11, 13, 15, 31, 36–40
36	KI2	Kappa2 index	7–8, 11, 13, 15, 31, 34–35, 37–40
37	KI3	Kappa3 index	7–8, 11, 13, 15, 31, 34–36, 38–40
38	KA1	KAlpha1 index	7–8, 11, 13, 15, 31, 35–40
39	KA2	KAlpha2 index	7–8, 11, 13, 15, 31, 34–38, 40
40	KA3	KAlpha3 index	7–8, 11, 13, 15, 31, 34–39

3. Compare the evaluation with the particle's previous best value, *pbest*[*i*]. If the current value is better than *pbest*[*i*], then *pbest*[*i*] = current value and the *pbest* location, *pbest x*[*i*][*d*], is set to the current location in *d*-dimensional space.
4. Compare the evaluation with the swarm's previous best value, (*pbest*[*gbest*]). If the current value is better than *pbest*[*gbest*]), then *gbest* = current particle's array index.
5. Change the velocity and position of the particle according to the following equations, respectively:

$$V[i][d] = V[i][d] + c_1 * rand() * (pbest\ x[i][d] - present\ x[i][d]) + c_2 * rand() * (pbest\ x[gbest][d] - persent\ x[i][d]), \quad (1)$$

$$present\ x[i][d] = present\ x[i][d] + V[i][d]. \quad (2)$$

6. Loop to step 2 until a stopping criterion, a sufficiently good evaluation function value, or a maximum number of iterations, is met.

Particle velocities in each dimension are clamped to a maximum velocity, $V_{\max}$, to control the exploration ability of particles. $V_{\max}$ is a problem-oriented parameter and should be set at about 10–20% of the dynamic range of the variable in each dimension [35]. The acceleration constants c_1 and c_2 represent the weighing of the stochastic terms that pull each particle toward *pbest* and *gbest* positions. They are normally set to 2.0 to give it a mean of 1 for the cognition and social parts, so that the particles thoroughly search the settled regions [36]. Under these conditions, the search space is statistically shrunk through iterations. Hence, this resembles a local search algorithm. In contrast, the first part helps expand the search space so that the particles can explore new areas. Thus, this implies a global search ability of the PSO. *rand*() is a uniform, random number generator within the (0, 1) range. This makes the system less predictable and more flexible.

Furthermore, there should be a balance between the global and local search abilities. An inertia weight w is then introduced into Eq. (1) as shown below:

$$V[i][d] = w * V[i][d] + c_1 * rand() * (pbest\ x[i][d] - present\ x[i][d]) + c_2 * rand() * (pbest\ x[gbest][d] - persent\ x[i][d]). \quad (3)$$

Logically, the PSO should possess more exploration ability at the beginning to find a good region, then more exploitation ability later on to refine search within it. Simply, w could be a positive linear or nonlinear function of time, which has a high value at the beginning and is gradually lower in each iteration. This is also a contribution to improve the convergence rate.

In feature selection, the input presented to the regression modeling is in the form of a table where the rows represent chemical compounds and the columns are the molecular descriptors. Each compound contains a value for each corresponding factor. How accurate a QSAR model can predict the biological activity of the compounds depends on their values in a subset of the selected features. Hence, the selection of each column or feature can be treated as a binary number. A numerical value of zero could be used to represent that the corresponding descriptor is not selected for QSAR modeling. Otherwise, a numerical value of one is assigned. This binary problem calls for some modification of the original PSO proposed to handle continuous problems [24]. Thus, *present* $x[i][d]$ representing the value stored by the i^{th} particle in the d^{th} dimension, can only take on the binary value, instead of a real valued number. This indicates whether the d^{th} feature is selected or not. Note that the D dimensions above are equal to the total number of descriptors.

Normally, *present* $x[i][d]$ stores a real number. After the update step (Eq. (2)), *present* $x[i][d]$ is discretized to a binary value by using probabilistic selection or roulette wheel selection. The fractional values of *present* $x[i][d]$ are treated as probability thresholds to determine subset membership. Each dimension or feature of the particle is assigned a slice of a roulette wheel whose size is proportional to *present* $x[i][d]$. The subset is assembled by spinning the wheel and selecting the features to which the wheel's marker points. This process is repeated k times which is the predefined number of selected features. The chosen descriptors are then set to 1 and the remaining paramters are set to 0. The actual probabilities, p_{id}, are computed as follows:

$$p_{id} = \frac{x_{id}^a}{\sum_{d=1}^{D} x_{id}^a} \quad (4)$$

where x_{id} is the fractional coordinates *present* $x[i][d]$ after the update step (Eq. (2)), and a is a scaling factor or selection pressure and is set to 2 [24]. This binary PSO or BPSO still presents the same advantages as the original PSO. The near-optimal solutions are found much faster, compared with the performance of a random search or an exhaustive search. This allows BPSO to perform feature selection efficiently in datasets with large numbers of descriptors. The objective function evaluated by the BPSO is the *Pearson* correlation coefficient that measures the quality of QSAR model with the selected features:

$$R = \frac{N\sum_{n=1}^{N} y_i\hat{y}_i - \sum_{n=1}^{N} y_i \sum_{n=1}^{N} \hat{y}_i}{\sqrt{\left[N\sum_{n=1}^{N} y_i^2 - \left(\sum_{n=1}^{N} y_i\right)^2\right] \times \left[N\sum_{n=1}^{N} \hat{y}_i^2 - \left(\sum_{n=1}^{N} \hat{y}_i\right)^2\right]}} \quad (5)$$

where N is the number of training compounds for regression and y_i and $\hat{y}_i$ are the measured and the predicted activities of the i^{th} compound, respectively. Since the NNs can be applied to approximate any function and do not require any prior statistical assumption on the data fitted, the NNs are then chosen as the modeling technique for QSAR study and described in the following subsection.

NNs for QSAR Modeling

QSAR modeling is rather complicated because there are many factors involved such as mass, surface area, volume, dipoles, molar refractivity, lipophilicity, *Verloop* parameters, connectivity, shape indices, counts of atoms, rings, groups, hydrogen bond donors and acceptors, and electrostatic parameters. Thus, the analytical approach for their modeling is extremely difficult due to their unknown nonlinearity nature. Feedforward neural networks have been considered a very powerful tool for function approximation and modeling. One of their advantages is the ability to learn from examples. Hence, they can be applied to model relationships between biological activity and relevant factors. A neural network normally has two elementary components; processing elements (or processing nodes) and connection weights. A feedforward architecture specifies that the network has no loops as opposed to feedback architecture. A classical backpropagation (BP) learning algorithm based on the gradient descent method is simply used to train and update the weights on each link of a neural network with training examples. These weights capture the relationship pattern of a multivariable function through learning. In other words, they are used to capture the relationships between chemical descriptors and biological activity. Weight adjustment between processing nodes in backpropagation is carried out according to the difference between the target value and the output value of the neural network. The difference of the error is measured by the sum of squared error as shown below:

$$E = \sum_{p=1}^{P} \sum_{k=1}^{K} (d_{pk} - o_{pk})^2 \tag{7}$$

where d_{pk} is the k^{th} desired value of the p^{th} data and o_{pk} is the actual output.

The weights ($\mathbf{W}$) are adjusted toward the gradient direction that produces a smaller approximation error as follows:

$$\mathbf{W} = \mathbf{W} + \eta\delta\mathbf{y} \tag{8}$$

where η is a positive constant called the learning rate, δ is the gradient of the difference between the desired and actual neuron's response, and $\mathbf{y}$ is the input vector.

However, the gradient descent method presents some weaknesses associated with long computational time, overfitting, and trapping in local optima. These can alleviated by training the NNs with a more efficient optimization technique like the PSO and the *Levenberg-Marquardt* (*LM*) algorithm. With the aforementioned appealing properties, PSO can be applied to train neural networks by optimizing their weights in place of the classical BP. This PSO based neural networks (PSONNs) should relieve some drawbacks posed by the gradient descent method. PSONNs were used to develop the QSAR model and combined with the feature selection conducted by the PSO. The *Levenberg-Marquardt* optimization algorithm was also selected to train NNs due to its very fast convergence for medium sized problems. It was designed to approach a Hessian, a matrix of second order derivatives of Eq. (7), with an approximation. Such estimation is obtained by averaging outer products of the first order derivative or gradient. Its computation is much less complex than that of the Hessian matrix. As a result, this heuristic algorithm performs much faster than the steepest descent method for training feedforward NNs. It is important to note that this *LM* algorithm is not an alternative to backpropagation. Rather, it is a variant of classical backpropagation. Hence, the *LM* based backpropagation neural networks (*LM*BPNNs) were also used to develop the QSAR model and combined with the feature selection accomplished by the PSO. They were created by using the neural network toolbox in MATLAB 6.5.

The implementation process of feedforward neural networks to model relationship between biological activity and its relevant factors can be roughly divided into four main steps, (1) assembling the data, (2) creating the network, (3) training the network, and (4) simulating the network.

In step (1), chemical structure data points were collected as described in *HEPT* Data Set and Descriptors Generation. Normally, input parameters

of the target function are composed of various magnitudes. The one with higher magnitude may dominate those with lower magnitude. Therefore, preprocessing should be applied to raw data before training. Thus, the raw data points were normalized to [−1, 1] for every factor. Since a large data set of 132 compounds was collected, the holdout method was chosen as a validation technique for model selection and performance estimation of the constructed model. This data set was thus randomly divided into two subsets for training and validating. Training the network was performed by using about 80% of the original data (110 data points) whereas the remaining 22 data points were used for validating.

In step (2), neural network creation, number of inputs or features were selected by the user and they were composed of the features selected by the PSO while the only output was the target, biological activity. Trial and error was used to determine the network architecture, which includes the number of hidden nodes. It was varied as shown in Tables 7 and 8 to find the highest correlation coefficient, *R*. The same procedure was conducted for both *LM*BPNNs and PSONNs. Note that the standard *hyperbolic tangent sigmoid* (*tansig*) function was used in the hidden layer to limit its output to a small range (−1, 1), whereas the linear (*purelin*) function was used in the output layer to allow the network output to be a real number.

In step (3), training the network is an attempt to determine a proper set of weights on each link of the NNs by minimizing the mean squared error performance function (difference between actual output and desired output). The discussed PSONN was implemented in MATLAB 6.5 running on a Pentium IV (2.4 GHz) with the Microsoft Windows XP operating system.

In step (4), both trained networks are then simulated with all data sets to check their predictive abilities.

Results and Discussion

Feature Set Selection by PSO

The computation of PSO depends on a few parameters such as population size, inertial weight, maximum velocity, maximum and minimum positions in each dimension, and the maximum number of iterations. A population size and maximum iterations of 20 and 50 were selected. The inertia weight gradually decreased from 0.9 to 0.4 so as to balance the global and local exploration. To prevent the tendency to explode, the maximum velocity V_{max} was set at 0.5.

The values of correlation coefficient (*R*) and the selected features from the utilization of PSO for feature selection and *LM*BPNNs for QSAR modeling (PSO–*LM*BP) are illustrated in Table 3. Similarly, the results from the combined methods between

Table 3. QSAR models by using PSO–*LM*BP (selected from 40 features)

Architecture	Features	*R*
7-7-1	8, 10, 22, 24, 34, 37, 38	0.994
7-6-1	8, 9, 16, 22, 23, 24, 26	0.989
7-5-1	4, 15, 27, 28, 30, 33, 38	0.985
7-4-1	5, 8, 21, 23, 28, 31, 38	0.976
7-3-1	5, 8, 11, 18, 22, 27, 35	0.961
6-6-1	7, 14, 16, 24, 31, 32	0.989
6-5-1	8, 14, 15, 24, 25, 31	0.980
6-4-1	5, 6, 10, 14, 27, 29	0.968
6-3-1	6, 16, 22, 24, 27, 31	0.950
6-2-1	8, 14, 22, 23, 27, 28	0.940
5-5-1	8, 10, 14, 17, 27	0.961
5-4-1	5, 8, 23, 31, 37	0.956
5-3-1	13, 15, 16, 26, 33	0.939
5-2-1	5, 8, 27, 31, 37	0.928
4-4-1	8, 11, 14, 33	0.938
4-3-1	5, 27, 28, 31	0.924
4-2-1	8, 16, 34, 35	0.927

Table 4. QSAR models by using PSO–PSO* (selected from 40 features)

Architecture	Features	*R*
7-7-1	2, 8, 21, 22, 24, 31, 34	0.888
7-6-1	2, 8, 11, 18, 21, 36, 37	0.872
7-5-1	8, 21, 22, 24, 29, 31, 34	0.881
7-4-1	3, 4, 5, 8, 15, 16, 29	0.865
7-3-1	2, 8, 11, 20, 21, 26, 38	0.876
6-6-1	4, 8, 10, 22, 31, 38	0.860
6-5-1	8, 21, 23, 28, 33, 38	0.859
6-4-1	1, 7, 8, 9, 11, 38	0.869
6-3-1	2, 8, 11, 28, 36, 38	0.858
6-2-1	8, 15, 21, 22, 29, 37	0.864
5-5-1	8, 19, 21, 30, 31	0.852
5-4-1	2, 8, 12, 21, 31	0.853
5-3-1	8, 19, 21, 31, 37	0.858
5-2-1	8, 21, 22, 30, 31	0.851
4-4-1	8, 21, 22, 35	0.859
4-3-1	4, 8, 19, 21	0.833
4-2-1	8, 18, 19, 21	0.842

*Swarm parameters: 20 particles, 50 iterations

PSO for descriptors selection and PSONNs for QSAR modeling (PSO–PSO) are depicted in Table 4. Seventeen various architectures from each approach were used to build the models. Both PSO–*LM*BP and PSO–PSO appeared to successfully generate highly predictive QSAR models. Feature 8 (MOR) was the most frequently selected feature by both PSO–*LM*BP and PSO–PSO methods as shown in Table 5. However, Table 2 indicates that MOR has high covariance with other descriptors. In order to avoid overestimations of the obtained models and hence increase its predictive ability, pairs of descriptors with a correlation coefficient $(R) \geq 0.8$ were classified as cross-correlated. If significant covariance exists between parameters, the combination of them should not be included in the same QSAR

Table 5. Frequency of feature set selections (selected from 40 features)

Feature		PSO–*LM*BP					PSO–PSO				
		7-x-1	6-x-1	5-x-1	4-x-1	Total	7-x-1	6-x-1	5-x-1	4-x-1	Total
1	CN1	–	–	–	–	–	–	1	–	–	1
2	CC4	–	–	–	–	–	3	1	1	–	5
3	CC5	–	–	–	–	–	1	–	–	–	1
4	CR1	1	–	–	–	1	1	1	–	1	3
5	CR2	2	1	2	1	6	1	–	–	–	1
6	CR3	–	2	–	–	2	–	–	–	–	–
7	MOV	–	1	–	–	1	–	1	–	–	1
8	MOR	4	2	3	2	11	5	5	4	3	17
9	LIP	1	–	–	–	1	–	1	–	–	1
10	TLI	1	1	1	–	3	–	1	–	–	1
11	TRI	1	–	–	1	2	2	2	–	–	4
12	TBI	–	–	–	–	–	–	–	1	–	1
13	TWI	–	–	1	–	1	–	–	–	–	–
14	ESI	–	4	1	1	6	–	–	–	–	–
15	SFI	1	1	1	–	3	1	1	–	–	2
16	VL1	1	2	1	1	5	1	–	–	–	1
17	VL2	–	–	1	–	1	–	–	–	–	–
18	VL3	1	–	–	–	1	1	–	–	1	2
19	B11	–	–	–	–	–	–	–	2	2	4
20	B12	–	–	–	–	–	1	–		–	1
21	B13	1	–	–	–	1	4	2	4	3	13
22	B21	3	2	–	–	5	2	2	1	1	6
23	B22	2	1	1	–	4	–	1	–	–	1
24	B23	2	3	–	–	5	2	–	–	–	2
25	B31	–	1	–	–	1	–	–	–	–	–
26	B32	1	–	1	–	2	1	–	–	–	1
27	B33	2	3	2	1	8	–	–	–	–	–
28	B51	2	1	–	1	4	–	2	–	–	2
29	B52	–	1	–	–	1	2	1	–	–	3
30	B53	1	–	–	–	1	–	–	2	–	2
31	KCM	1	3	2	1	7	2	1	4	–	7
32	KC1	–	1	–	–	1	–	–	–	–	–
33	KC2	1	–	1	1	3	–	1	–	–	1
34	KC3	1	–	–	1	2	2	–	–	–	2
35	KI1	1	–	–	1	2	–	–	–	1	1
36	KI2	–	–	–	–	–	1	1	–	–	2
37	KI3	1	–	2	–	3	1	1	1	–	3
38	KA1	3	–	–	–	3	1	4	–	–	5
39	KA2	–	–	–	–	–	–	–	–	–	–
40	KA3	–	–	–	–	–	–	–	–	–	–
	Total	35	30	20	12	97	35	30	20	12	97

Table 6. Twenty-seven descriptors in the QSAR models

Feature	Abbreviation	Descriptor
1	CN1	Atomic charge of N_1
2	CC4	Atomic charge of C_4
3	CC5	Atomic charge of C_5
4	CR1	Atomic charge of R_1
5	CR2	Atomic charge of R_2
6	CR3	Atomic charge of R_3
7	MOV	Molecular volume
8	LIP	log P
9	TLI	Total lipole
10	TBI	*Balaban* topological index
11	ESI	Sum of E-state indices
12	VL1	*Verloop* L (R_1)
13	VL2	*Verloop* L (R_2)
14	VL3	*Verloop* L (R_3)
15	B11	*Verloop* B1 (R_1)
16	B12	*Verloop* B1 (R_2)
17	B13	*Verloop* B1 (R_3)
18	B21	*Verloop* B2 (R_1)
19	B22	*Verloop* B2 (R_2)
20	B23	*Verloop* B2 (R_3)
21	B51	*Verloop* B5 (R_1)
22	B52	*Verloop* B5 (R_2)
23	B53	*Verloop* B5 (R_3)
24	KCM	*Kier* Chi0 (atoms) index
25	KC1	*Kier* Chi0 (atoms) index (R_1)
26	KC2	*Kier* Chi0 (atoms) index (R_2)
27	KC3	*Kier* Chi0 (atoms) index (R_3)

model. For example MOR has high covariance with log P (LIP). The use of LIP in the model would be sufficient to represent both features. This LIP would also represent other physicochemical properties that might be highly corrrelated with it as well.

Out of 40 parameters, 13 descriptors were eliminated owing to cross-correlations, and hence 27 descriptors shown in Table 6 were submitted to the PSO selection stage. Note that the feature descriptors from here onward follow the feature descriptors in Table 6. Seventeen various architectures from each approach again were used to build the models with selected features from these 27 properties. The R results with varied number of features specified are shown in Tables 7 and 8 for PSO–*LM*BP and PSO–PSO, respectively. Clearly, the obtained results were much higher after the preprocessing step. Feature 8 (LIP) is the most frequently selected feature with or without preprocessing.

Seventeen runs of the QSAR modeling by using PSO–*LM*BP indicated that features 8 (LIP), 12 (VL1), 13 (VL2), 21 (B51), 24 (KCM), and 27 (KC3) were frequently selected as shown in Table 9. Features 8 (LIP), 11 (ESI), 12 (VL1), 13 (VL2), 25 (KC1), and 26 (KC2) were commonly chosen by using the combined PSO–PSO.

Table 7. QSAR models by using PSO–*LM*BP (selected from 27 features)

Architecture	Features	R
7-7-1	7, 8, 10, 11, 12, 24, 26	0.992
7-6-1	12, 14, 15, 21, 24, 26, 27	0.987
7-5-1	8, 13, 20, 21, 24, 26, 27	0.982
7-4-1	8, 13, 15, 21, 24, 25, 27	0.974
7-3-1	8, 12, 13, 14, 20, 21, 24	0.966
6-6-1	9, 11, 12, 19, 22, 27	0.983
6-5-1	8, 10, 13, 19, 25, 27	0.975
6-4-1	3, 11, 12, 18, 19, 23	0.960
6-3-1	8, 11, 12, 13, 21, 27	0.956
6-2-1	8, 13, 21, 24, 25, 27	0.948
5-5-1	13, 19, 21, 24, 25	0.968
5-4-1	7, 8, 22, 24, 25	0.957
5-3-1	4, 8, 10, 13, 26	0.944
5-2-1	11, 12, 13, 17, 27	0.920
4-4-1	21, 24, 26, 27	0.950
4-3-1	8, 22, 25, 27	0.936
4-2-1	7, 8, 13, 14	0.922

Table 8. QSAR models by using PSO–PSO* (selected from 27 features)

Architecture	Features	R
7-7-1	8, 10, 13, 17, 25, 26, 27	0.920
7-6-1	4, 8, 12, 20, 24, 26, 27	0.925
7-5-1	8, 12, 18, 19, 23, 24, 26	0.930
7-4-1	1, 3, 6, 8, 11, 12, 26	0.919
7-3-1	8, 13, 15, 17, 24, 25, 26	0.921
6-6-1	3, 11, 12, 16, 26, 27	0.915
6-5-1	8, 12, 16, 20, 24, 26	0.916
6-4-1	3, 4, 8, 11, 12, 26	0.925
6-3-1	8, 11, 13, 21, 22, 27	0.918
6-2-1	3, 8, 11, 12, 16, 26	0.917
5-5-1	8, 9, 12, 13, 22	0.916
5-4-1	8, 13, 14, 25, 26	0.910
5-3-1	8, 13, 19, 25, 26	0.909
5-2-1	8, 11, 13, 21, 25	0.900
4-4-1	8, 18, 22, 25	0.903
4-3-1	8, 18, 22, 25	0.901
4-2-1	8, 13, 15, 21	0.902

*Swarm parameters: 50 particles, 200 iterations

The selected features can be divided into 4 groups based on structural properties. The first group was lipophilicity (LIP). Most people use the partition coefficient for water/octanol (log P). Lipophilicity is a measure of the ability of molecules to move between

Table 9. Frequency of feature set selections (selected from 27 features)

Feature		PSO–*LM*BP					PSO–PSO				
		7-x-1	6-x-1	5-x-1	4-x-1	Total	7-x-1	6-x-1	5-x-1	4-x-1	Total
1	CN1	–	–	–	–	–	1	–	–	–	1
2	CC4	–	–	–	–	–	–	–	–	–	–
3	CC5	–	1	–	–	1	1	3	–	–	4
4	CR1	–	–	1	–	1	1	1	–	–	2
5	CR2	–	–	–	–	–	–	–	–	–	–
6	CR3	–	–	–	–	–	1	–	–	–	1
7	MOV	1	–	1	1	3	–	–	–	–	–
8	LIP	4	3	2	2	11	5	4	4	3	16
9	TLI	–	1	–	–	1	–	–	1	–	1
10	TBI	1	1	1	–	3	1	–	–	–	1
11	ESI	1	3	1	–	5	1	4	1	–	6
12	VL1	3	3	1	–	7	3	4	1	–	8
13	VL2	3	3	3	1	10	2	1	4	1	8
14	VL3	2	–	–	1	3	–	–	1	–	1
15	B11	2	–	–	–	2	1	–	–	1	2
16	B12	–	–	–	–	–	–	3	–	–	3
17	B13	–	–	1	–	1	2	–	–	–	2
18	B21	–	1	–	–	1	1	–	–	2	3
19	B22	–	3	1	–	4	1	–	1	–	2
20	B23	2	–	–	–	2	1	1	–	–	2
21	B51	4	2	1	1	8	–	1	1	1	3
22	B52	–	1	1	1	3	–	1	1	2	4
23	B53	–	1	–	–	1	1	–	–	–	1
24	KCM	5	1	2	1	9	3	1	–	–	4
25	KC1	1	2	2	1	6	2	–	3	2	7
26	KC2	3	–	1	1	5	5	4	2	–	11
27	KC3	3	4	1	2	10	2	2	–	–	4
	Total	35	30	20	12	97	35	30	20	12	97

fat and water. It is often used to indicate how easily a molecule may be transported across membranes. The second group was *Verloop* parameters (VL1, VL2, and B51). They are a set of multi-dimensional steric factors that helps explain the steric influence of substituents in the interaction of inhibitors with RT. The VL1 and VL2 features are defined as the maximum length of the substituents R_1 and R_2, respectively, along the axis of the bond between the first atom of the substituents and the parent molecule. B51 is the maximum width of the substituent R_1 in any direction perpendicular to the bond axis. The next group was connectivity indices (*Kier* Chi0), KCM, KC1, KC2, and KC3, which reflect the immediate bonding environment of atoms of the molecule and the substituents R_1, R_2, and R_3, respectively. The last one was the electrotopological state indices (E-state indices, ESI), based on the electronegativity of an atom and its local topology. Then, the sum of the states of all the individual atoms in the molecule, was calculated.

QSAR Modeling by NNs

The efficacy of both neural network versions can be improved by increasing the number of processing nodes (number of nodes in hidden layers). However, this is not always a solution to obtain high predictive ability even though the value of R is very high due to the over-fitting problem. To overcome this problem, the holdout method was used to determine a proper number of hidden nodes. This procedure can be applied to any number of selected features. Suppose the required features were six. The six most frequently selected descriptors are then used to develop the QSAR model as shown in Table 9. The number of hidden nodes was then varied while computing these selected features. The dataset was split into

Table 10. The prediction ability of PSO–*LM*BP QSAR models

Architecture[a]	R_{training}[b]	R_{testing}[b]
6-7-1	0.977	0.829
6-6-1	0.967	0.796
6-5-1	0.966	0.821
6-4-1	0.953	0.847
6-3-1	0.945	0.828
6-2-1	0.916	0.801

[a] Selected features: 8, 12, 13, 21, 24, and 27 which have the highest frequency selections from 27 features following the PSO–*LM*BP result in Table 9
[b] Average values of 6 calculations

two subsets, for training and unseen testing. Normally, about 80% of the total data set is used for training and the remaining 20% is used for testing. The best intersection point between the *R*s from training and the *R*s from testing are selected to avoid over-fitting. This implies that the *R*s from the training set rise when the number of hidden nodes is increased. Also, the *R*s of the testing set, rise when the number of hidden nodes is increased. However, this trend does not last long for the testing set. As shown in Table 10, the highest value of R_{testing} was 0.847, when the hidden node was only four, which was not the highest one attempted. This procedure was applied to both versions of neural networks as illustrated in both Tables 10 and 11. Also, to avoid a local optima trap, six runs with various initial weights were carried out and their average correlation coefficients were tabulated in these tables. Since the test subsets were not used for training, it can be concluded that both types of neural networks can perform well in determining the biological activity. Data distribution assumptions required in traditional statistical approaches can be discarded in this approach.

Table 11. The prediction ability of PSO–PSO QSAR models

Architecture[a]	R_{training}[b]	R_{testing}[b]
6-7-1	0.879	0.715
6-6-1	0.892	0.701
6-5-1	0.881	0.677
6-4-1	0.869	0.765
6-3-1	0.837	0.739
6-2-1	0.875	0.716

[a] Selected features: 8, 11, 12, 13, 25, and 26 which have the highest frequency selections from 27 features following the PSO–PSO result in Table 9
[b] Average values of 6 calculations

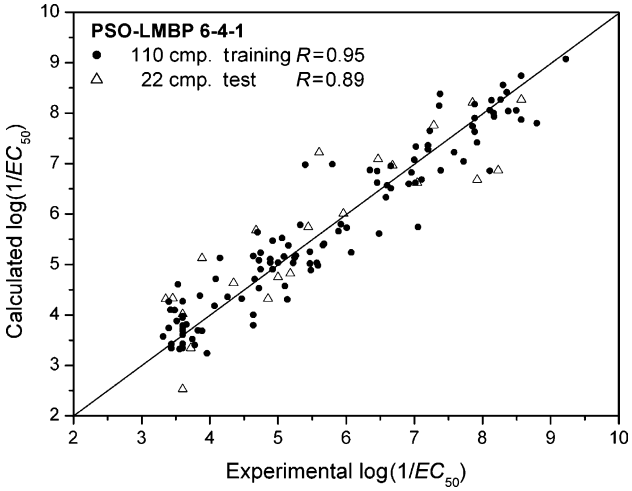


Fig. 1. Plot of calculated $\log(1/EC_{50})$ by PSO–*LM*BP 6-4-1 *versus* the experimental values

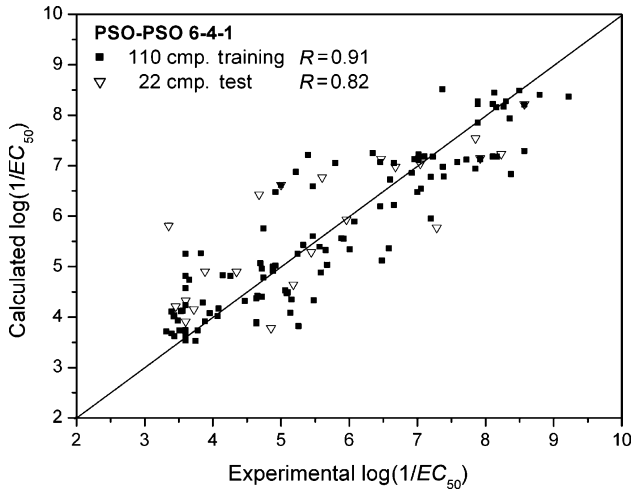


Fig. 2. Plot of calculated $\log(1/EC_{50})$ by PSO–PSO 6-4-1 *versus* the experimental values

The plots of the selected 6-4-1 architecture calculated $\log(1/\text{EC}_{50})$ by these approaches with the experimental values are shown in Figs. 1 and 2. The R_{testing} of PSO–*LM*BP (Fig. 1, R_{testing} 0.89) is higher than that of PSO–PSO (Fig. 2, R_{testing} 0.82). In the PSO–PSO method, there is one compound (**111**) for which the experimental and the calculated values differ by 2.5. The selected model consists of 6 features (LIP, ESI, VL1, VL2, KC1, and KC2) of which two indicate the importance of R_1 functional group (VL1 and KC1). The R_1 (CH_2CH_2Ph) of this compound appears only in the test set. Therefore **111** can be classified as an outlier. If it is removed from

the testing compounds, the value of R_{testing} of the selected PSO–PSO model will increase to be 0.85.

The PSO–*LM*BP performed apparently better than the PSO–PSO as depicted in Tables 7 and 8. This implies that the *LM*BP was more effective and more efficient than the PSO for neural network training. The PSO produced less stable results due to its partial random nature and larger solution space exploration. This was also dependent on the number of particles and number of maximum iterations utilized. Even though the PSO was implicitly built for speed due to its simple concept and primitive mathematical operators employed, its major drawback for training NNs was the long computational time, since its performance relied heavily on a swarm of particles. Each cycle of each particle was quite fast, but a group of particles would take some time compared with the *LM*BP, which was invented to speed up the optimization process. Moreover, the evaluated functions of NNs were not complicated and can be solved analytically.

Conclusions

In this study, particle swarm optimization (PSO) was adopted for major feature set selection in QSAR modeling of *HEPT* analogues. The method was based on a discrete binary modification. The fitness function was the *Pearson* correlation which was curve fitted by both *LM*BPNNs and PSONNs. The adopted PSO showed satisfactory performance with the large dataset attempted. Enhanced performance could be accomplished by choosing only one representative from a group of highly correlated descriptors. The obtained outcomes of QSAR study were also dependent on the QSAR model developed. Therefore, two different training algorithms, *LM*BP and PSO, were investigated for constructing QSAR models. Both PSO–*LM*BP and PSO–PSO produced highly predictive QSAR models. The PSO–*LM*BP performed more effectively than the PSO–PSO due to the PSO's larger solution space exploration and partial random nature. Even though the PSO was built for speed, its population-based characteristics still required longer computational time, especially when compared with the speed-up gradient method, such as the *Levenberg-Marquardt* algorithm.

The number of selected descriptors for QSAR models was fixed and specified by the user. The determination of the appropriate number of features should be studied next. Since the feature selection depends on the reliability of QSAR modeling, a regression technique that can generalize well should also be the subject of future work.

Acknowledgement

The first author was partially supported by the Thailand Research Fund (TRF) grant MRG4980170. We are grateful to the Institute of Theoretical Chemistry, University of Vienna for program support, and *Paul V. Neilson* for carefully reading through the manuscript.

References

[1] Rogers D, Hopfinger AJ (1994) J Chem Inf Comput Sci **34**: 854
[2] Glen WD, Dunn WJ, Scott RD (1989) Tetrahedron Comput Methodol **2**: 349
[3] Wikel J, Dow E (1988) Bioorg Med Chem Soc **110**: 5959
[4] Hemmateenejad B, Miri R, Akhond M, Shamsipur M (2002) Chemom Intell Lab Syst **64**: 91
[5] Tang K, Li T (2002) Chemom Intell Lab Syst **64**: 55
[6] Lu Q, Shen G, Yu R (2002) J Comput Chem **23**: 1357
[7] Miyasaka T, Tanaka H, Baba M, Hayakawa H, Walker RT, Balzarini J, de Clercq E (1989) J Med Chem **32**: 2507
[8] Hannongbua S, Lawtrakul L, Limtrakul J (1996) J Comput-Aided Mol Des **10**: 145
[9] Hannongbua S, Lawtrakul L, Sotriffer CA, Rode BM (1996) Quant Struct-Act Relat **15**: 389
[10] Lawtrakul L, Hannongbua S (1999) Sci Pharm **67**: 43
[11] Klein CT, Lawtrakul L, Hannongbua S, Wolschann P (2000) Sci Pharm **68**: 25
[12] Hannongbua S, Nivesanond K, Lawtrakul L, Pungpo P, Wolschann P (2001) J Chem Inf Comput Sci **41**: 848
[13] Lawtrakul L, Prakasvudhisarn C (2005) Monatsh Chem **136**: 1681
[14] Luco JM, Ferretti FH (1997) J Chem Inf Comput Sci **37**: 392
[15] Jalali-Heravi M, Parastar F (2000) J Chem Inf Comput Sci **40**: 147
[16] Gaudio AC, Montanari CA (2002) J Comput-Aided Mol Des **16**: 287
[17] Douali L, Villemin D, Cherqaoui D (2003) J Chem Inf Comput Sci **43**: 1200
[18] Weekes D, Fogel GB (2003) BioSystems **72**: 149
[19] Arakawa M, Hasegawa K, Funatsu K (2006) Chemometr Intell Lab **83**: 91
[20] Sutter JM, Dixon SL, Jurs PC (1995) J Chem Inf Comput Sci **35**: 77
[21] So SS, Karplus M (1996) J Med Chem **39**: 1521
[22] Yasri A, Hartsough D (2001) J Chem Inf Comput Sci **41**: 1218
[23] Hasegawa K, Miyashita Y, Funatsu K (1997) J Chem Inf Comput Sci **37**: 306
[24] Agrafiotis DK, Cedeño W (2002) J Med Chem **45**: 1098

[25] Tanaka H, Baba M, Hayakawa H, Sakamaki T, Miyasaka T, Ubasawa M, Takashima H, Sekiya K, Nitta I, Shigeta S, Walker RT, Balzarini J, de Clercq E (1991) J Med Chem **34**: 349
[26] Tanaka H, Takashima H, Ubasawa M, Sekiya K, Nitta I, Baba M, Shigeta S, Walker RT, de Clercq E, Miyasaka T (1992) J Med Chem **35**: 337
[27] Tanaka H, Takashima H, Ubasawa M, Sekiya K, Nitta I, Baba M, Shigeta S, Walker RT, de Clercq E, Miyasaka T (1992) J Med Chem **35**: 4713
[28] Tanaka H, Takashima H, Ubasawa M, Sekiya K, Inouye N, Baba M, Shigeta S, Walker RT, de Clercq E, Miyasaka T (1995) J Med Chem **38**: 2860
[29] Ren J, Esnouf R, Garman E, Somers D, Ross C, Kirby I, Keeling J, Darby G, Jones Y, Stuart D, Stammers D (1995) Nat Struct Biol **2**: 293
[30] WebLab ViewerPro 4.0, Molecular Simulations Inc., San Diego, 2000
[31] Gaussian 03, Revision C02, Frisch MJ, Trucks GW, Schlegel HB, Scuseria GE, Robb MA, Cheeseman JR, Montgomery JA Jr, Vreven T, Kudin KN, Burant JC, Millam JM, Iyengar SS, Tomasi J, Barone V, Mennucci B, Cossi M, Scalmani G, Rega N, Petersson GA, Nakatsuji H, Hada M, Ehara M, Toyota K, Fukuda R, Hasegawa J, Ishida M, Nakajima T, Honda Y, Kitao O, Nakai H, Klene M, Li X, Knox JE, Hratchian HP, Cross JB, Bakken V, Adamo C, Jaramillo J, Gomperts R, Stratmann RE, Yazyev O, Austin AJ, Cammi R, Pomelli C, Ochterski JW, Ayala PY, Morokuma K, Voth GA, Salvador P, Dannenberg JJ, Zakrzewski VG, Dapprich S, Daniels AD, Strain MC, Farkas O, Malick DK, Rabuck AD, Raghavachari K, Foresman JB, Ortiz JV, Cui Q, Baboul AG, Clifford S, Cioslowski J, Stefanov BB, Liu G, Liashenko A, Piskorz P, Komaromi I, Martin RL, Fox DJ, Keith T, Al-Laham MA, Peng CY, Nanayakkara A, Challacombe M, Gill PMW, Johnson B, Chen W, Wong MW, Gonzalez C, Pople JA (2004) Gaussian Inc., Wallingford CT
[32] Tsar^{TM} 3.3, Oxford Molecular Ltd., Oxford, 2000
[33] Eberhart R, Kennedy J (1995) Proc. of the 6th Int. Symp. On Micro Machine and Human Science, IEEE Service Center, Piscataway, NJ, p 39
[34] Shi Y, Eberhart RC (1999) Proc IEEE Cong Evol Comp, IEEE Service Center, Piscataway, NJ, p 1945
[35] Eberhart RC, Shi Y (2001) Proc IEEE Cong Evol Comp, IEEE Service Center, Piscataway, NJ, p 81
[36] Kennedy J, Eberhart R (1995) Proc. of IEEE Int. Conf. on Neural Networks, Piscataway, NJ, p 1942