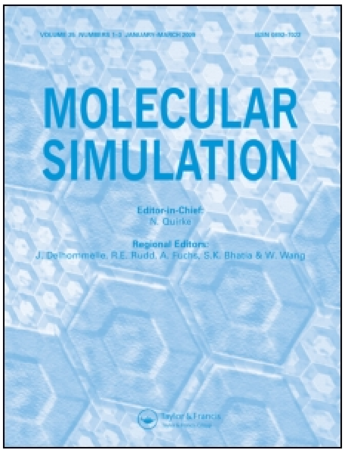


Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

QSAR modeling for thiolactomycin analogues using genetic algorithm optimized artificial neural networks

J. Liu[a]; L. Zhou[a]
[a] Department of Pharmaceutical Engineering, College of Chemical Engineering, Sichuan University, Chengdu, Sichuan, China

To cite this Article Liu, J. and Zhou, L.(2007) 'QSAR modeling for thiolactomycin analogues using genetic algorithm optimized artificial neural networks', Molecular Simulation, 33: 8, 629 — 638

To link to this Article: DOI: 10.1080/08927020701361892

URL: http://dx.doi.org/10.1080/08927020701361892

Molecular Simulation, Vol. 33, No. 8, July 2007, 629–638

Taylor & Francis
Taylor & Francis Group

QSAR modeling for thiolactomycin analogues using genetic algorithm optimized artificial neural networks

J. LIU and L. ZHOU*

Department of Pharmaceutical Engineering, College of Chemical Engineering, Sichuan University, Sichuan, Chengdu 610065, China

(Received January 2007; in final form March 2007)

Fatty acid biosynthesis (FAB) is essential for bacterial survival. Components of this biosynthetic pathway have been identified as attractive targets for the development of new antibacterial agents. Thiolactomycin is an inhibitor of type II fatty acid synthase (FAS). Two-dimension quantitative structure–activity relationship (2D QSAR) such as partial least squares (PLS), quadratic partial least squares (QPLS), artificial neural networks (ANN), genetic algorithm (GA) optimized ANN connection weights (GA-ANN), GA select the most relevant descriptors (GA-ANN-GA) are conducted on a series of potent thiolactomycin analogues. Compare the four methods, the pedictive ability the models is evaluated by the root-mean-square error (RMSE) and R^2 for the training set and the test set. The GA-ANN-GA show the best results, the RMSE for the training set and the test set are 0.0718 and 0.9473, respectively, the R^2 for the training set and the test set are 0.0702 and 0.9504, respectively. GA optimized ANN will provide a superior alternative method 2D QSAR models.

Keywords: FabH inhibitors; Thiolactomycin; QSAR; GA optimized ANN

1. Introduction

Fatty acid biosynthesis (FAB) in bacteria, plants and animals is carried out by the ubiquitous fatty acid synthase (FAS) system [1]. FAB is an essential metabolic process for prokaryotic organisms and is required for cell viability and growth. Targeting this pathway consequent represents a reasonable approach for developing new antibacterial agents. In the type I system of animals, including humans, FAS is a homodimer of two large polypeptides, each composed of several distinct enzyme domains and an integral acyl carrier protein (ACP). In the type II system of bacteria, plants, and protozoa, the FAS components, including the ACP, exist as discrete proteins [2]. The corresponding enzymes of the type I and II FAS are related in structure and function but generally lack overall sequence homology. Large multifunction proteins termed proteins termed type I FAS catalyze these essential reactions in eukaryotes [3]. In contrast, bacterial use multiple enzymes to accomplish the same goal and are referred to as type II, or dissociated, FAS [4]. The bacterial system and proteins bear little homology to the human system and therefore represent a set of attractive target proteins for novel antibacterial development. Since many of today's nosochomial bacterial infections are resistant to several of the available antibiotics, compounds targeting the FAB pathway could fill a serious medical need [5].

A key enzymes responsible for initiation of bacterial FAB has so far escaped serious attention by the drug discovery industry. FabH, a β-ketoacyl–acyl carrier protein synthase, is the bacterial condensing enzyme in Gram-positive and -negative bacteria that initiates the FAB cycle by catalyzing the first condensation step between acetyl-CoA and malonyl-ACP (figure 1) [6].

Thiolactomycin(TLM,1) is a thiotetronic acid-containing natural product that inhibits bacterial and plant type II FAS, which provide essential building blocks for bacterial cell walls (figure 2) [7]. The structure and antibiotic properties of TLM were reported by Noto *et al.* [8]. The compound has moderate *in vitro* activity against a broad spectrum of pathogens, including Gram-positive and -negative bacteria and has also shown encouraging anti-malarial activity. TLM has chemotherapeutic potential, as it is non-toxic to mice and afford significant protection against urinary tract and intraperitoneal bacterial infections [9]. Nowadays there are many synthesized TLM analogues.

Quantitative structure–activity relationship (QSAR) models, mathematical equations relating chemical structure

*Corresponding author. Tel.: + 86-28-81938747. Email: zhouluscu@163.com

Molecular Simulation
ISSN 0892-7022 print/ISSN 1029-0435 online © 2007 Taylor & Francis
http://www.tandf.co.uk/journals
DOI: 10.1080/08927020701361892

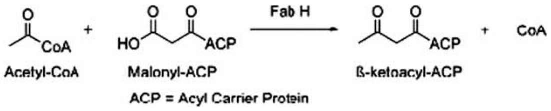


Figure 1. FabH-catalyzed initiation reaction of FAB.

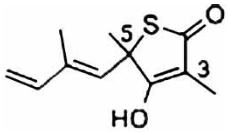


Figure 2. Structure of thiolactomycin.

to their biological activity, give information that is useful for drug design and medicinal chemistry [10]. A challenging problem in QSAR studies is the selection of the suitable modeling method. Different techniques have been used for establishing QSAR models including partial least squares (PLS) [11], quadratic partial least squares (QPLS) [12], and artificial neural networks (ANN) [13]. ANNs have grown in popularity due to their ease of use and success in solving problems where complex nonlinear relationship exist [14–17]. Genetic algorithm (GA) are general evolutionary algorithms that can be used for optimization [18]. Specifically, we integrate GA and the back propagation algorithm to learn the connection weights of an ANN. Following are advantages of combining genetic search with gradient descent (back propagation) algorithm:

(1) global search approach that is less likely to get stuck in local optima,
(2) higher probability of convergence (as a result of global search) to a global optimum,
(3) heuristic global near-optimum (obtained by genetic search) solution is improved by local gradient descent algorithm (back propagation) to obtain the true global optimal solution, and
(4) (4) parallel genetic search approach offers several potential solutions for holdout sample.

We propose a predictive technique based on the GA optimize the ANNs connection weights and select variable. We refer this technique as GA-ANN and GA-ANN-GA, respectively. In this paper, we build the QSAR models of the TLM analogues by means of five methods (PLS, QPLS, ANN, GA-ANN and GA-ANN-GA), compare the results obtained when modeling QSAR using GA-ANN-GA against PLS, QPLS, ANN and GA-ANN.

2. Method

2.1 Biological activity data

A series of TLM analogues against parasite were taken from the studies reported by Simon *et al.* [19], and Ross *et al.* [20], which inhibits *Plasmodium falciparum* grown *in vitro*. Half-maximal inhibition concentrations ($IC_{50}S$) for 51 TLM analogues were measures. These data were collected in several different literature sources with experimental error of 20–25%. These values were converted to negative logarithm $\log(1/IC_{50})$ (briefly described as PIC50) as the dependent variable representing the biological activity of these compounds. The basic structures of these compounds are shown in figure 2 and the substituent patterns (SPs) of the compounds along with PIC50 were used in this study are given in tables 1–3.

2.2 Quantum chemical descriptors calculation

The molecular modeling calculations of all the TLM derivatives are performed by Hyperchem software for Windows (Hypercube, FL, USA). The MM + molecular mechanics force field was first run to approach to the optimizes geometry. The conformation obtained from molecular mechanics was subjected to a refined geometry optimization using AM1 semiempirical molecular orbital theory. The AM1 Hamiltonian was selected because it gives good estimates of molecular energies and the computation time is much shorter than need by *ab initio* methods. The calculated descriptors for each molecule are summarized in table 4. Local charges(LC), calculated according to Mulliken populations, and electrostatic potential (EP) at each atom, highest occupied molecular orbital (HOMO), and lowest unoccupied molecular orbital (LUMO) energies and molecular dipole moment (MDP) were calculated by Hyperchem. Quantum chemical indices of hardness (η); softness (S); electronegativity (χ); chemical potential (μ); electrophilicity (ω) were calculated according to the method proposed by Thanikaivelan *et al.* [21]. The brief description of the quantum chemical descriptors, calculated for this study, is represented in table 4. Then, the descriptors were collected in an ($n \times m$) data matrix (D), where n and m were the number of compounds and the number of descriptors, respectively.

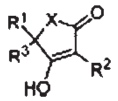
2.3 Partial least squares

PLS is introduced by Wold [12] and is commonly used in chemometrics as modeling alternatives to ordinary least squares (OLS) when the predictor matrix is poorly conditioned. The PLS regression method is well suited for problems with multicollinear predictor and response variables. PLS is explained in detail in literature [22,23], and only a summary of the PLS method is presented. PLS first projects both the predictor (X) and response (Y) variables onto one or more new axes with "outer relations" (factor) yielding scores that contain most of the information in the observed variables.

The PLS model has the form:

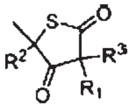
$$\begin{aligned} &X = tp' + E \quad Y = uq' + F \quad u + bt + h \\ &Y = tq'b + f'' \end{aligned} \tag{1}$$

Table 1. Effect of variation of the heteroatom and substituents at C3 and C5 on inhibition of *P. falciparum*



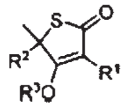
Compound	*R*[1]	*R*[2]	*R*[3]	*X*	*PIC*$_{50}$
1	Me	Me	Isoprenoid	S	3.84
2	Me	Et	H	O	3.45
3	Me	Me	Hexyl	O	3.65
4	Me	Et	Decyl	O	4.03
5	Me	Bu	Decyl	O	4.82
6	Me	Ac	Decyl	NH	4.20
7	Me	Pr	Hexadecyl	S	5.22
8	Me	Bn	Hexadecyl	S	5.15
9	Me	Et	$-(CH_2)_2O(CH_2)_2O(CH_2)_5CH_3$	S	4.22
10	Me	Pr	$-(CH_2)_6OTBS$	S	4.43
11	Me	Pr	$-(CH_2)_6OH$	S	3.80
12	Me	Et	H	S	3.63
13	Me	Pr	H	S	3.54
14	Me	Bn	H	S	3.71
15	Me	Me	Hexyl	S	3.86
16	Me	Me	Octyl	S	3.82
17	Me	Me	Geranyl	S	4.40
18	Me	Me	Decyl	S	4.44
19	Me	Me	Hexadecyl	S	4.60
20	Me	Et	Hexyl	S	4.21
21	Me	Et	Octyl	S	4.27
22	Me	Et	Decyl	S	4.82
23	Me	Et	Hexadecyl	S	4.72
24	Me	Pr	Decyl	S	5.00
25	Me	Bn	Decyl	S	4.30
26	Et	Me	H	S	3.50
27	decyl	Me	H	S	4.14
28	Et	Me	Decyl	S	4.19
29	Et	Me	Hexadecyl	S	4.46
30	Me	Bu	Decyl	S	4.14
31	Me	Me	$-CH_2-CH{=}C(CH_3)_2$	S	4.00
32	Me	Me	$-(CH_2)_2CH(CH_3)_2$	S	3.59
33	Me	Me	$-CH_2CH{=}CCH_3(CH_2)_2CH{=}C(CH_3)_2$	S	5.10
34	Me	Me	$-(CH_2)_2CHCH_3(CH_2)_2CH{=}C(CH_3)_2$	S	4.24
35	Me	Me	$-(CH_2)_2CHCH_3(CH_2)_3CH(CH_3)_2$	S	4.14
36	Me	Me	$-CH_2CH{=}CCH_3(CH_2)_2CH{=}CCH_3(CH_2)_2CH{=}C(CH_3)_2$	S	5.10
37	Me	Me	$-CH_2C_6H_5C{=}O(C6H5)$	S	4.80
38	Me	Me	H	S	3.33

Table 2. Effect of 3,3-disubstituted compounds on inhibition of *P. falciparum*



Compound	*R*[1]	*R*[2]	*R*[3]	*PIC*$_{50}$
39	Me	H	Allyl	4.21
40	Me	Decyl	Allyl	4.59
41	Pr	Decyl	Allyl	4.72
42	Pr	H	Geranyl	5.05
43	Pr	Decyl	Geranyl	4.11
44	Me	H	Me	4.34
45	Me	Me	$-CH_2CH{=}CCH_3(CH_2)_2CH{=}C(CH_3)_2$	4.52

Table 3. Effect of 4-OH substituted compound on inhibition of *P. falciparum*



Compound	*R¹*	*R²*	*R³*	*PIC₅₀*
46	Me	Decyl	Allyl	5.22
47	Pr	Decyl	Allyl	6.00
48	Pr	Decyl	Bn	5.40
49	Pr	Decyl	Me	4.35
50	Pr	H	$-(\mathrm{CH}_2)_6\mathrm{OTBS}$	5.23
51	Pr	Decyl	—(CH2)6OH	4.51

where X and Y are the matrices of predictors and responses. The matrices on the right-hand side this model are defined by $t = X -$ scores, $u = Y$-scores, $p = X$-loadings, $q = Y$-loadings, $E = X$-residuals, and $F = Y$-residuals. PLS algorithms choose successive orthogonal factors that maximize the covariance between the X- and Y-scores. The correlation usually decreases from one factor to the next.

Table 4. The calculated quantum chemical descriptors used in this study.

Descriptor	*Brief description*
$\mathrm{LC_i}$	The local charges at each atom of the base unit of Thiolactomycin
MPC	Most positive charge
MNC	Most negative charge
SSC	Sum of squares of charges
SSPC	Sum of squares of positive charges
SSNC	Sum of squares of negative charges
AAC	Average of absolute charges
ASC	Average of square of charges
$\mathrm{EP_i}$	The electrostatic potential at each atom of the base unit of Thiolactomycin
MNEP	Most negative electrostatic potential
LNEP	Least negative electrostatic potential
SDEP	Standard deviation of electrostatic potentials
AEP	Average of electrostatic potential
$\mathrm{DM_t}$	Total molecular dipole moment
DM_x	Molecular dipole moment at X-direction
DM_y	Molecular dipole moment at Y-direction
DM_z	Molecular dipole moment at Z-direction
ET	Total molecular energy
EB	Molecular binding energy
EI	Molecular isolated energy
EE	Molecular electronic energy
EC	Molecular core–core interaction energy
HF	Heat of molecular formation
SM	Molecular surface area
VM	Molecular volume
LOGP	n-octanol/water partition
RM	Molecular molar refractivity
PM	Polarzability of molecular
MM	Mass of molecular
HOMO	Energy of the highest occupied molecular orbital
LUMO	Energy of the lowest unoccupied molecular orbital
X	Electronegativity; −0.5(HOMO − LUMO)
H	Hardness; 0.5(HOMO + LUMO)
S	Softness;$1/\eta$
Ω	Electrophilicity; $\chi^2/2\eta$

To obtain the PLS model with the best predictive performance, the number of PLS components that optimizes the predictive ability of the model should be determined [24]. This is typically done by cross-validation, a procedure in which the available data within the training set are split into several subgroups (called validation sets).

In this application, validation sets were used. The prediction residual sum of squares (PRESS) for the test samples is determined as a function of the number PLS components retained in the regression model that was formed with the training data. The procedure is usually repeated several times, with each subset in the training set being part of the test samples at least once. The total PRESS error over all the test sets as a function of the number of PLS components is then used to determine the optimum number of PLS components, i.e. the number of PLS components that produces minimum PRESS error. Predictions with PLS are performed after an estimation of the regression coefficient B that was obtained from the training set. It is done by multiplying a block of independent or predictor variables by the regression coefficient B. The predicted values for the test set can be compared to the actual values by calculating root-mean-square error of prediction (RMSE). RMSE is calculated as in equation (2):

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (y_i' - y_i)^2}{n}} \tag{2}$$

where, y_i', values of the predicted values, and y_i, values of the actual values.

2.4 *Quadratic partial least squares*

Although the original linear PLS (LPLS) regression method provides good remedial measure to the problems of correlated inputs and limited observations, it has the major limitation that only linear information can be extracted from data. Since many practical data are inherently nonlinear, it is desirable to have a robust method that can model any nonlinear relation. A successful step towards nonlinear PLS modeling was the QPLS method proposed by Wold *et al.* [12]. In QPLS, quadratic functions are used for the inner regression in PLS.

The QPLS has the form:

$$X = tp' + E \quad Y = uq' + \mathrm{F}$$

where X and Y are the matrices of predictors and responses. Then, the vectors t and u are related to each other by a quadratic regression model:

$$u = c_0 + c_1 t + c_2 t^2 + h.$$

Since the score vectors t are orthogonal to each other, the first dimension can be peeled-off from X and Y analogously to the LPLS regression method, letting the set of vectors of the next dimension be estimated from the residual matrices: $E = X - tp'$ and $F = Y - (c_0 + c_1 t + = Y - (c_0 + c_1 t + c_2 t^2)q'$ instead of X and Y, respectively. This procedure is repeated until no significant information is left.

2.5 Artificial neural networks

ANNs [13] are typically organized in layers where these layers are made up of a number of interconnected nodes which contain an activation function. Input vectors are presented to the network via the input layer which communicates to one or more "hidden layers" where the actual processing is done via a system of weighted "connections".

Most ANNs contain some form of "learning rule" which modifies the weighs of the connections according to input patterns that it is presented with. There are many different kinds of learning rules used by neural networks, in this work, back-propagation artificial neural networks (BP-ANN) was used. In BP-ANN, "learning" is a supervised process that occurs with each cycle of "epoch" through a forward activation flow of inputs and the backwards error propagation of weight adjustment. There are many variations of the back-propagation algorithm [25]. The simplest implementation of back-propagation learning updates the network weights and biases in the direction in which performance function decreases most rapidly, the negative of the gradient. One iteration of this algorithm can be written as in equation (3)

$$X_{K+1} = X_K - \alpha_K g_K \tag{3}$$

where X_k is a vector of current weights and biases, g_k is the current gradient and α_k is the learning rate. In this work, gradient descent with momentum is applied and the performance function was RMSE.

For the basic gradient descent algorithm, the weights and biases are moved in the direction of the negative gradient of the performance function. Gradient descent with momentum often provides faster convergence [25] because momentum allows a network to respond not only to the local gradient but also to recent trends in the error surface. Momentum can also help the network to overcome a shallow local minimum in the error surface and settle down at or near the global minimum [26]. Momentum can be added to back-propagation learning by making weight changes equal to the sum of a fraction of the last weight change and the new change suggested by the back-propagation rule.

2.6 Genetic algorithm optimized ANN connection weights

GA, was first introduced in the early 1970s [18], is becoming an important tool for optimizing functions. GA is a searching or optimizing algorithm based on Darwinian biological evolution principle. It is a structured probabilistic algorithm which starts with a population of randomly generated candidates and evolves toward better solutions by applying so-called "genetic operators", modeled based on genetic processes occurring in nature. Genetic operators are used to create offsprings in the next generation that differ from their parents, but maintain characteristics of parents. There are many ways that these operators can be implemented. Generally, the chromosomes in the population are represented as strings of binary digits. The implementation of chromosomes is usually called encoding scheme.

The BP algorithm by which the network is trained begins with a random set of weights. GA use survival of the fittest strategy to learn connection weights in an ANN [18,27,28]. GA are parallel search techniques that start with a set of random potential solutions and use special search operators (evaluation, selection, crossover and mutation) to bias the search towards the promising solutions. At any given time, unlike any optimization approach, GA has several promising potential solutions (equal to population size) as opposed to one optimal solution. Each population member in a GA is a potential solution. A population member (P1) used to learn the strength of connections for ANN shown in figure 3. P1 can be represented as

$$\mathrm{P1} = \langle \mathrm{w14; w15; w16; w24; w25; w26; w31; w35; w36;}$$
$$\mathrm{w47; w57; w67}\rangle;$$

where (w14; w15; w16; w24; w25; w26; w31; w35; w36; w47; w57; w67) $\in R$

Any w $\in$ P1 is called a gene (connection weight) of a given population member P1. A set of several population members is called population Ω. The number of population members Ω is called population size. The number of genes in a population member is called the defining length of the population member ζ. The defining length of all the population members in a given population is constant.

GA starts with a random set of population. An evaluation operators is then applied to evaluate the fitness of each for classification, the evaluation function is number of correctly classified cases. A selection operator is then applied to select the population members with

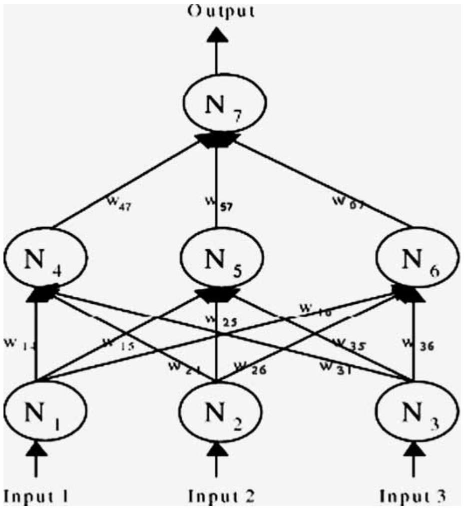


Figure 3. A three-layer ANN.

higher fitness (so that they can be assigned higher probability for survival). Under selection operator, individual population members may be born, allowed to live or die. Several selection operators are reported in literature; the operators are proportionate reproduction, ranking selection, tournament selection and steady state selection [29]. Among the popular selection operators are ranking and tournament selection. Goldberg and Deb [29] showed that both ranking and tournament selection maintain strong population fitness growth potential under normal conditions. Tournament selection operator, however, requires lower computational overhead. The time complexity of ranking selection is $O(n \log n)$ whereas, the time complexity of tournament selection is $O(n)$ where n is number of population members in a population. In tournament selection two random pair of individuals are selected and the member with the better fitness of the two is admitted to the pool of individuals for further genetic processing. The process is repeated in the way that the population size remains constant and the best individual in the population always survives. For our research, we use tournament selection operator.

After selection operator is applied the new population special operators called crossover and mutation is applied with a certain probability. For applying the crossover operator, the status of each population member is determined. Every population member is assigned a status as a survivor or non-survivor. The number of population members equal to survivor status is approximately equal to population size$*$(1 − probability of crossover). The number of non-surviving members are approximately equal to population size$*$probability of crossover. The non-surviving members in a population are then replaced by applying crossover operators to randomly selected surviving members. Several crossover operators exist in the literature, we use arithmetic crossover operators in our research. Arithmetic crossover consists of producing children in a way that every gene in a child is convex combination of genes from its two parents. Given the following two parents P1 and P2 a child C1 can be produces as follows:

$$\mathrm{P1} = \langle \mathrm{w14}, \mathrm{w15}, \mathrm{w16}, \mathrm{w24}; \mathrm{w25}; \mathrm{w26}, \mathrm{w34}, \mathrm{w35}, \mathrm{w36}, \mathrm{w47}, \mathrm{w57}, \mathrm{w67} \rangle;$$

$$\mathrm{P2} = \langle \mathrm{w14}, \mathrm{w15}, \mathrm{w16}, \mathrm{w24}; \mathrm{w25}; \mathrm{w26}, \mathrm{w34}, \mathrm{w35}, \mathrm{w36}, \mathrm{w47}, \mathrm{w57}, \mathrm{w67} \rangle;$$

$$\mathrm{C1} = \langle \mathrm{k14}, \mathrm{k15}, \mathrm{k16}, \mathrm{k24}, \mathrm{w25}, \mathrm{k26}, \mathrm{k34}, \mathrm{k35}, \mathrm{w36}, \mathrm{k47}, \mathrm{k57}, \mathrm{k67} \rangle;$$

where $k_{ij} = \sigma w_{ij} + (1 - \sigma) w_{ij}$, $\sigma \in [0,1]$ is a random number. The bold font is used to represent the genes from parent P2.

Arithmetic crossover ensures that every gene in the child is bounded by the respective genes from the both parents. Arithmetic crossover is a popular crossover operator when GA is used for optimization. Mutation operator randomly picks a gene in a surviving population member (with the probability equal to probability of mutation) and replaces it with a real random number between a given range of maximum and minimum real values.

We use a three layer (of nodes) ANN with 12 input, 7 hidden and 1 output node. For our architecture, we have a population member defining length of $\zeta = 99$((12 inputs + 1 threshold)$*$7 hidden + (7 hidden + 1 threshold)$*$1 output).

2.7 GA select the most relevant descriptors

Recently, GA is also used in drug design [30], a wide range of studies in QSAR gain advantage from the use of GA. QSAR models are typically in several steps. When the number of descriptors exceeds the number of compounds in the data set, so that one is dealing with an undetermined problem where undesirable overfitting can result [31–33]. This problem can be avoided by preprocessing the descriptor set with a feature selection routine that determines which of the descriptors have a significant influence on the activity of a given compound. GAs, which are clearly well-suited to tackle problem of this kind, were introduced to the field of QSARs to address this need [27,28,34,35]. After the most relevant features have been selected, this set serves as input to ANN, and the QSAR model building is executed by ANN. When the neural network is training, we also uses GA optimize connection weights. We name this method GA-ANN-GA.

In GA for variable selection, the chromosome and its fitness in the species represent a set of variables and predictvity of the derives QSAR model, respectively. Each individual of the population define by a chromosome of binary values represented a subset of descriptors. The number of genes at each chromosome is equal to the number of descriptors. The population of the first generation is selected randomly. A gene took a value of 1 if its correspond descriptor is included in the subset; otherwise, it took a value of zero. The number of genes with a value of 1 is kept relatively low to have a small subset of descriptors, that is, the probability of generating 0 for a gene is set greater (at least 60%) than the value of 1. The operator used here are crossover and mutation. The probability of the application of these operators is varied linearly with generation renewal (0–0.1% for mutation and 60–90% for crossover). The fitness score of each member of this new generation is again evaluated, and the reproductive cycle is continued until a desired number of generations or target fitness score is reached. Here the fitness function is the root-mean-square errors of training, validation, and testing set.

Finally, six descriptors, molecular electronic energy (EE), heat of molecular formation (HF), n-octanol/water partition (LOGP), total molecular dipole potential ($\mathrm{DM_t}$), softness (ω), and energy of the lowest unoccupied molecular orbital (LUMO) are selected to build the QSAR model.

2.8 Software

The molecular modeling calculations of all the TLM derivatives are performed by Hyperchem software for Windows (Hypercube). The PLS, QPLS, ANN, GA-ANN, GA-ANN-GA program used were the toolbox of MATLAB 7.0.1 developed by Math Works.

3. Result and discussion

As mentioned in the introduction, the aim of this work is to compare the results obtained when modeling QSAR using GA-ANN-GA against PLS, QPLS, ANN, GA-ANN. In this section, the prediction performances of the method proposed GA-ANN-GA and four other models (PLS, QPLS, ANN and GA-ANN) are evaluated. All the data are mean centered and scaled to unit variance [− 1, 1] before modeling. After the model has been built, the calculated values of PIC50 need to be transferred back to the same units that were used for the original experimental values of PIC50 for comparison purpose. To ensure a fair comparison, the same training and test set are used for each of the models.

To obtain robust and accurate models, the models should be trained by subset of descriptors instead of all generated descriptors. There exist two ways of reducing the descriptors space. One is to select the features with respect for their generalization ability, which is called feature selection. The other alternative is to extract features by building linear and nonlinear combinations of a lower dimension of the input features, which is called feature extraction. The former give models which are simple to interpret, however, in the presence of large number of descriptors, the ANN modeling becomes complex and time consuming. In contrast, the latter extracts the information contents of the original descriptors into new variables by simple algorithms such as PCA [36]. In addition, the extracted variables from the original descriptors should be selected before entering into the model. The potential usefulness of the feature extraction is that the information from large number of descriptors is extracted to a few numbers of new variables by using simple algorithms. To reduce the dimension of the calculated descriptors and consequently to increase the speed of calculation and overall performances of the PLS, QPLS, ANN, GA-ANN and GA-ANN-GA.

PLS analysis was performed by using 24 descriptors and 12 components. The PLS model predicts the training data with $R^2 = 0.6724$ and RMSE = 0.3177. However, the prediction result for the test set gives $Q^2 = 0.7225$ and RMSE = 0.2960, indicating that the LPLS model is inappropriate for fitting this data. The plots of calculated vs. observed values PIC50 are shown in figure 4.

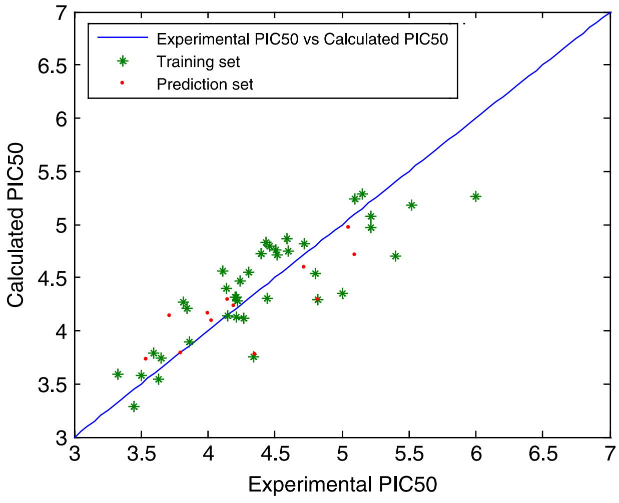


Figure 4. A plot experimental vs. calculated PIC50 from the PLS.

When QPLS is used to approximate the model, R^2 and Q^2 is greatly increased to 0.8409 and 0.8556, respectively, and RMSE for the training and test set is reduced to 0.2431 and 0.2185, respectively. Compare the prediction qualities of PLS and QPLS, predicted values were plotted against observed data. In such plots, the data will fall on the diagonal $(y = y')$ if the model fits the data perfectly. As shown in figure 5, the PLS model does not fit the data adequately, since the data are distributed almost perpendicularly to the horizontal axis. Evidently, use of QPLS instead of PLS improves the predictive ability. Applying QPLS model cause the data points to fall on the diagonal direction more compactly, indicate that the predictive ability is further enhanced by using QPLS.

The reason for this is the predictor variables of the TLM analogues data are correlated with each other and response variable. The better prediction performance of QPLS compared to PLS suggest that a nonlinear correlation structure should not be modeled using a linear approach due to the risk of including noise in the model while trying

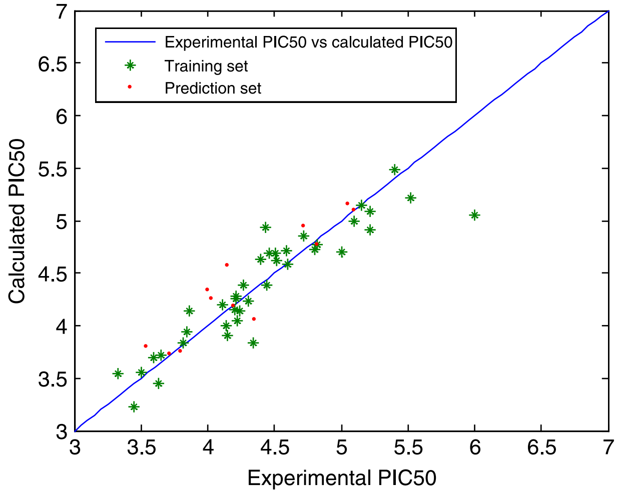


Figure 5. A plot experimental vs. calculated PIC50 from the QPLS.

to account for nonlinearity. As for PLS in this example, for instance, projecting the nonlinear input data onto a linear subspace cannot model the nonlinearity of the relationship properly. In contrast, QPLS aims to model such nonlinear relationships via a nonlinear mapping into a feature space.

The neural-network approach is especially suitable for analyzing complex nonlinear relationships between the outputs and inputs. The number of neurons in the hidden layer is an important factor determining the network's performance. That is, too many nodes cause the network to memorize the dataset (overfitting); network with few nodes may be insufficient to use all the information from the dataset (underfitting). It is desirable to construct the network that generalizes the patterns of the dataset rather than that merely memorizes them [31]. Previous studies conducted to determine the appropriate number of hidden units suggest that ρ, the ratio of number of data points to the number of adjustable weights in the neural network, should have a value between 1.8 and 2.3 [31,32]. The range of ρ was used as a guideline for an acceptable number of neurons in the hidden layer. When the increase of hidden neurons did not improve the model anymore.

As the BP algorithm by which the network is trained begins with a random set of weights, the weights may have an influence of the results of the models. For each different number of neurons in the hidden layer, we build 1000 QSAR models, respectively. The performance of the neural network model was evaluated with the combination of the mean of RMSE and R^2 of the 1000 training set and 1000 test set, the results are summarized in table 5. The architecture of 12–7–1 with minimum RMSE of training and testing set was 0.1876 and 0.1806, respectively, and with maximum R^2 of the training and test set is 0.8761 and 0.8863, respectively. The architecture of 12–7–1 is selected as the best nonlinear model. The plots of calculated vs. observed values PIC50 from the best nonlinear model are shown in figure 6.

However, the ANN as a gradient search algorithm has some limitations, such as overfitting, local optimum problems, and sensitivity to the initial values of weights. GA use survival of the fittest strategy to learn connection weights in an ANN. The GA search process was terminated as the generation number reaches a predefined value of 200. The corresponding RMSE of the training set and test set is 0.1032 and 0.0923, respectively, and R^2 of the training set and Q^2 of the test set is 0.9165 and 0.9289, respectively. In consequence, the BPNN prediction performance is considerably improved by optimizing connection weights. The improvements of GA-ANN models over conventional BPNN, PLS and QPLS could be seen from table 6.

Table 5. The influence of different hidden neurons on the ANNs' performance.

	Training set		Test set	
ANN[†]	*RMSE*	*R^2*	*RMSE*	*Q^2*
12–3–1	0.2401	0.8432	0.2145	0.8578
12–4–1	0.2233	0.8503	0.2098	0.8633
12–5–1	0.2065	0.8632	0.1961	0.8707
12–6–1	0.1973	0.8699	0.1893	0.8797
12–7–1	0.1876	0.8761	0.1806	0.8863
12–8–1	0.2304	0.8489	0.2123	0.8605
12–9–1	0.1953	0.8715	0.1889	0.8804
12–10–1	0.1904	0.8732	0.1853	0.8841
12–11–1	0.2081	0.8603	0.1974	0.8699

[†] Number of inputs-hidden-outputs neurons.

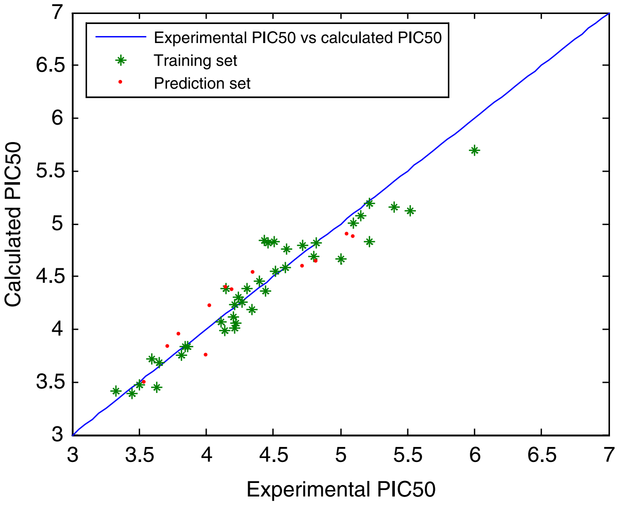


Figure 6. A plot experimental vs. calculated PIC50 from the ANN.

Recently, some published papers suggest that GA may be useful in data analysis, especially in the task of number of features for regression models [34,35], the advantages are the followings. A way to save CPU time is to minimize the size of the neural network without losing effectiveness. This has the additional advantage that the risk of overfitting due to a limited data set is reduced. Besides faster computations, another advantage in dealing with a small set of input descriptors is that subsequent analysis can be greatly simplified. It can be found that the steric complementarity and the hydrophobic interaction between the inhibitors and the receptor are very important to the biological activity. We can focus on the few key elements that appear to determine the biological activities. The last one is the result of GA-ANN-GA is best, it can be seen from table 6.

For classification, the RMSE and R^2 for the training set is ordered by the following: GA-ANN-GA gave the best results (0.0718 and 0.9473); GA-ANN and ANN showed better results (0.1032 and 0.9165) and (0.1876 and

Table 6. The results of GA-ANN-GA, GA-ANN, BPNN, PLS and QPLS.

	Training set		Testing set	
Model	*RMSE*	*R^2*	*RMSE*	*Q^2*
PLS	0.3177	0.6724	0.2960	0.7225
QPLS	0.2431	0.8409	0.2185	0.8556
ANN	0.1876	0.8761	0.1806	0.8863
GA-ANN	0.1032	0.9165	0.0923	0.9289
GA-ANN-GA	0.0718	0.9473	0.0702	0.9504

0.8761); QPLS show good results (0.2431 and 0.8409) than PLS (0.3177 and 0.6724). For prediction, the RMSE and R^2 for the training set is ordered by the following: GA-ANN-GA gave the best results (0.0702 and 0.9504); GA-ANN and ANN showed better results (0.0923 and 0.9289) and (0.1806 and 0.8863); QPLS show good results (0.2185 and 0.8556); PLS provided a poor result (0.2960 and 0.7225).

The RMSE for the training sets of the linear model is greater than 0.25, and the RMSE for the training set of the nonlinear model are all lower than 0.25, which indicate that the nonlinear model is better than the linear model during the training process. For the prediction ability, the RMSE of best linear and nonlinear model is 0.3177 and 0.0718, respectively, which indicate that the predictive ability of best nonlinear model is better than that of best linear model.

As discussed earlier, the predictor variables of the data are nonlinearly correlated with each other and with the response variable. GA select the most relevant descriptors will increase the speed of computation, reduce the risk of overfitting, and simplify the future drug design. As a result, the prediction ability is best. The GA-ANN optimizing ANN connection weights, the prediction performance was considerably improved, in subsequence the results of GA-ANN is better than ANN. However, the neural-network approach is especially suitable for analyzing complex nonlinear relationships between the outputs and inputs. The results of ANN is better than QPLS and PLS.

4. Conclusion

Many of today's nosochomial bacterial infections are resistant to several of the available antibiotics, compounds targeting the FAB pathway could fill a serious medical need. TLM, is a good FAB inhibitor, which has moderate *in vitro* activity against a broad spectrum of pathogens.

In this paper, we build the QSAR models of the TLM analogues by means of five methods (PLS, QPLS, ANN, GA-ANN and GA-ANN-GA), and we propose use GA-ANN-GA to build QSAR models. The proposed approach GA-combined ANN, which effectively selecting the most relevant descriptors and optimizing ANN connection weights, and ANN, which captures nonlinear relationships among predictor variables as well as with response variables through high-dimension feature mapping. Compare with the QPLS, ANN has a better prediction performance than QPLS. However the predictive ability of nonlinear model was better than that of linear model, PLS provided a poor result. It suggest that nonlinear correlation structures should not be modeled using linear approaches due to the risk of including noise in the model while trying to account for the nonlinearity. Compare with other nonlinear methods, GA-combined ANN has the following main advantages: (1) it can avoid overfitting, local optimum and optimize the ANN connection weights; (2) the prediction performance for the response variables are superior to that of LPLS; and (3) simplify the future drug design.

The GA-optimized ANN approach described in this paper shows great promise but requires further study, such as GA can also optimize the ANN architecture (delete "and select the most relevant descriptors"). We believe that with further development, the proposed method will provide a superior alternative method 2D QSAR models.

Acknowledgements

This work was partially supported by China West Medical College.

References

[1] J.E. Cronan Jr., C.O. Rock. Biosynthesis of membrane lipids. In Escherichia coli *and* Salmonella typhimurium*: Cellular and Molecular Biology*, *et al.* F.C. Neidhardt (Eds.), 2nd ed., pp. 612–636, American Society for Microbiology, Washington, DC (1996).

[2] R.C. Clough, A.L. Matthis, S.R. Barnum, JG Jaworski. Purification and characterization of 3-ketoacyl–acyl carrier protein synthase III from spinach. A condensing enzyme utilizing acetyl-coenzyme A to initiate fatty acid synthesis. *J. Biol. Chem.*, **20992**, 267 (1992).

[3] A.K. Joshi, A. Witkowski, S. Smith. Mapping of functional interactions between domains of the animal fatty acid synthase by mutant complementation *in vitro*. *Biochemistry*, **2316**, 36 (1997).

[4] S. Jackowski, C.M. Murphy, J.E. Cronan Jr., C.O. Rock. Acetoacetyl–acyl carrier protein synthase. A target for the antibiotic thiolactomycin. *J. Biol. Chem.*, **7624**, 264 (1989).

[5] R.J. Heath, S.W. White, C.O. Rock. Inhibitors of fatty acid synthesis as antimicrobial chemotherapeutics. *Microbiol. Biotechnol.*, **695**, 58 (2002).

[6] J.T. Tsay, W. Oh, T.J. Larson, S. Jackowski, C.O. Rock. Isolation and characterization of the β-ketoacyl carrier protein synthase β gene (*fabH*) from *Escherichia coli* K-12. *J. Biol. Chem.*, **6807**, 267 (1992).

[7] R.A. Slayden, R.E. Lee, J.W. Armour, A.M. Cooper, I.M. Orme. Antimycobacterial action of thiolactomycin: an inhibitor of fatty acid and mycolic acid synthesis. *Antimicrob. Agents. Chemother.*, **2813**, 40 (1996).

[8] T. Noto, S. Miyakawa, H. Oishi, H. Endo, H. Okazaki. Thiolactomycin, a new antibiotic. III. *In vitro* antibacterial activity. *J. Antibiot.*, **401**, 35 (1982).

[9] S. Miyakawa, K. Suzuki, T. Noto, Y. Harada, H. Okazaki. Thiolactomycin, a new antibiotic. IV. Biological properties and chemotherapeutic activity in mice. *J. Antibiot.*, **411**, 35 (1982).

[10] D. Hadjipavlou-Litina. Review, revaluation, and new results in quantitative structure–activity studies of anticonvulsants. *Med. Res. Rev.*, **91**, 18 (1998).

[11] S. Wold. PLS for multivariate linear modeling. In *Chemometric Methods in Molecular Design*, H. van de Waterbeemd (Ed.), pp. 195–218, VCH, Weinheim (1995).

[12] S. Wold, N. Kettaneh-Wold, B. Skagerberg. Nonlinear PLS modeling. *Chemom. Intell. Lab. Syst.*, **53**, 7 (1989).

[13] G. Schneider, P. Wrede. Artificial neural networks for computer-based molecular design. *Prog. Biophys. Mol. Biol.*, **175**, 70 (1998).

[14] M. Shamsipur, B. Hemmateenejad, M. Akhond. Multicomponent acidbase titration by principal component-artificial neural network calibration. *Anal. Chim. Acta*, **147**, 461 (2002).

[15] J. Hunger, G. Huttner. Optimization and analysis of force field parameters by combination of genetic algorithms and neural networks. *J. Comput. Chem.*, **455**, 20 (1999).

[16] S. Ahmad, M.M. Gromiha. Design and training of a neural network for predicting the solvent accessibility of proteins. *J. Comput. Chem.*, **1313**, 24 (2003).

[17] C.L. Waller, M.P. Bradley. Development and validation of a novel variable selection technique with application to multidimensional quantitative structure–activity relationship studies. *J. Chem. Inf. Comput. Sci.*, **345**, 39 (1999).

[18] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*, pp. 40–64, Addison-Wesley, Reading, MA (1989).

[19] M.J. Simon, E.U. Jonathan, K. Marcel, B. Reto, L.H. John, B. Colin, H.G. Ian. Analogues of thiolactomycin as potential antimalarial agents. *J. Med. Chem.*, **5932**, 48 (2005).

[20] F.W. Ross, A.R. Stuart, B.R. Michael, S. Vanessa, D.D. James, E.M. David, F.C. Alan, S.B. Gurdyal, I.M. Geoffrey. A type II pathway for fatty acid biosynthesis presents drug. *Antimicrob. Agents Chemother.*, **297**, 47 (2003).

[21] P. Thanikaivelan, V. Subramanian, J.R. Rao, B.U. Nair. Application of quantum chemical descriptor in quantitative structure activity and structure property relationship. *Chem. Phys. Lett.*, **59**, 323 (2000).

[22] A. Lorber, L.E. Wangen, B.R. Kowalski. A theoretical foundation for the PLS algorithm. *J. Chemometrics*, **19**, 1 (1987).

[23] A. Hoskuldsson. PLS regression methods. *J. Chemometrics*, **211**, 2 (1998).

[24] H. Swierenga, A.P. deWeijer, R.J. vanWijk, L.M.C. Buydens. Strategy for constructing robust multivariate calibration models. *Chemom. Intell. Lab. Syst.*, **1**, 49 (1999).

[25] H. Demuth, M. Beale. *Neural Network Toolbox User's Guide*, pp. 137–194, The MathWorks, Natick, MA (1998).

[26] M.T. Hagan, H.B. Demuth, M.H. Beale. *Neural Netw. Des.*, 45 (1996).

[27] M. Mitchell. *An Introduction to Genetic Algorithms*, pp. 205–218, MIT Press, Cambridge, MA (1996).

[28] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd ed., pp. 178–197, Springer-Verlag, Berlin, Heidelberg, NY (1996).

[29] D.E. Goldberg, K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*, G. Rawlins (Ed.), pp. 69–93, MorganKaufmann, San Mateo, CA (1991).

[30] R. Judson. Genetic algorithms and their use in chemistry. *Reviews in Computational Chemistry*, pp. 1–73, VCH Publishers, New York (1997).

[31] S.S. So, W.G. Richards. Application of neural networks:quantitative structure–activity relationships of the derivatives of 2,4-diamino-5-(substituted-benzyl) phrimidines as DHFR inhibitors. *J. Med. Chem.*, **3201**, 35 (1992).

[32] T.A. Andrea, H. Kalayeh. Applications of neural networks in quantitative structure–activity relationships of dihydrofolate reductase inhibitors. *J. Med. Chem.*, **2824**, 34 (1991).

[33] D.T. Manallack, D.D. Ellis, D.J. Livingstone. Analysis of linear and nonlinear QSAR data using neural networks. *J. Med. Chem.*, **3758**, 37 (1994).

[34] B.T. Hoffman, T. Kopajtic, J.L. Katz, A.H. Newman. 2D QSAR modeling and preliminary database searching for dopamine transporter inhibitors using genetic algorithm variable selection of Molconn Z descriptors. *J. Med. Chem.*, **4151**, 43 (2000).

[35] D.B. Turner, P. Willett. Evaluation of the EVA descriptor for QSAR studies: 3. The use of a genetic algorithm to search for models with enhanced predictive properties (EVA_GA). *J. Comput. Aided Mol. Des.*, **1**, 14 (2000).

[36] H. Gohlke, F. Dullweber, W. Kamm, J. Marz, T. Kissel, G. Klebe. Prediction of human intestinal absorption using a combined simulated annealing/backpropagation neural network approach. In *Rational Approaches to Drug Design*, H.D. Hfltje, M. Sippl (Eds.), pp. 261–270 (2001).