Chapter 8

AutoWeka: Toward an Automated Data Mining Software for QSAR and QSPR Studies

Chanin Nantasenamat, Apilak Worachartcheewan, Saksiri Jamsak, Likit Preeyanon, Watshara Shoombuatong, Saw Simeon, Prasit Mandi, Chartchalerm Isarankura-Na-Ayudhya, and Virapong Prachayasittikul

Abstract

In biology and chemistry, a key goal is to discover novel compounds affording potent biological activity or chemical properties. This could be achieved through a chemical intuition-driven trial-and-error process or via data-driven predictive modeling. The latter is based on the concept of quantitative structure-activity/property relationship (QSAR/QSPR) when applied in modeling the biological activity and chemical properties, respectively, of compounds. Data mining is a powerful technology underlying QSAR/QSPR as it harnesses knowledge from large volumes of high-dimensional data via multivariate analysis. Although extremely useful, the technicalities of data mining may overwhelm potential users, especially those in the life sciences. Herein, we aim to lower the barriers to access and utilization of data mining software for QSAR/QSPR studies. AutoWeka is an automated data mining software tool that is powered by the widely used machine learning package Weka. The software provides a user-friendly graphical interface along with an automated parameter search capability. It employs two robust and popular machine learning methods: artificial neural networks and support vector machines. This chapter describes the practical usage of AutoWeka and relevant tools in the development of predictive QSAR/QSPR models. Availability: The software is freely available at http://www.mt.mahidol.ac.th/autoweka.

Key words Quantitative structure-activity relationship, Quantitative structure-property relationship, QSAR, QSPR, Data mining

1 Introduction

Interactions of drugs with their target proteins are governed by a multitude of molecular forces including electrostatic, hydrophobic, polar, and steric. They can be rationalized through the use of the quantitative structure-activity/property relationship (QSAR/QSPR) paradigm, which has been one of the foremost tools in the arsenal of recent drug discovery efforts and has been almost 100 years in the making. The prelude to the formulation of QSAR/QSPR was the preliminary and independent efforts of Brodin [1]

Hugh Cartwright (ed.), *Artificial Neural Networks*, Methods in Molecular Biology, vol. 1260,
DOI 10.1007/978-1-4939-2239-0_8, © Springer Science+Business Media New York 2015

and Cros [2] in the mid-1850s; they established that there exists an association between chemical constitution and physiological properties. This was at a time prior to the reporting of the aromatic ring structure of benzene in 1865 by Kekulé [3]. In 1868, Brown and Fraser observed changes to physiological action upon methylation of the basic nitrogen atom in alkaloids. In 1869, Richardson [4] showed that aliphatic alcohols of different molecular weight also displayed varied narcotic effects. Similarly, a few decades later in 1893, Richet [5] showed that the water solubility of polar chemicals (such as alcohols, ethers, and ketones) was inversely related to their toxicities, whereby a decrease in the solubility of a polar chemical results in an increase in its toxicity. Toward the 1900s, Meyer and Overton [6, 7] reported that the lipophilicity of a group of organic compounds exhibited a linear relationship with their narcotic potencies. In 1917, Moore [8] investigated the effects of chemical "fumigants" on insects and showed that toxicity of these compounds increased with their boiling point. It can be seen that the majority of these efforts pertained to the establishment of qualitative relationships between chemicals and their respective activity/property.

Quantitative relationships between chemical structures and activity/property started to take shape when Hammett [9] introduced a simple equation (later to be known as the Hammett equation) in 1937 that considers the substituent effect caused by electron-withdrawing or electron-donating groups as summarized by the sigma constant, while the rho constant describes the structural class or chemotype under study. Such an equation allows determination of electronic effects caused by the placement of substituent groups at different positions (i.e., ortho-, meta-, and para-) of the benzene ring. The subsequent work of Taft [10] led to the introduction of the steric parameter. It is these two contributions from Hammett and Taft that paved the way for further contributions from Hansch et al. In the mid-1950s to 1960s, Hansch and Muir employed the Hammett substituent constant and partition coefficients in their formulation of equations to correlate the chemical substituents and growth of *Avena* plant using plant growth regulators comprising of phenoxyacetic acids and chloromycetins [11–13]. Finally in 1964, Hansch and Fujita [14] investigated the biological effects of a wide range of chemicals using a combination of physicochemical properties as summarized by the following equation:

$$\log \frac{1}{C} = a \cdot \sigma + b \cdot \pi - c \cdot \pi^2 + d$$

where C is the molar concentration of hormone, σ is the Hammett parameter, and π is a measure of hydrophobicity. In parallel, Free and Wilson [15] analyzed such structure-activity relationships using a binary approach in which the presence or absence of a substituent

was described as 0 and 1, respectively. These landmark reports marked the beginning of classical QSAR/QSPR. Later in 1969, Hansch stressed the importance of computers in analyzing structure-activity relationships [16], which was an observation that is true to this very day, in which the ever-increasing amount of biological and chemical data makes computers indispensable in the analysis of these relationships.

The essence of QSAR/QSPR lies in the predictive ability of these models to relate a set of explanatory variables (**X**) with their response variables (**Y**). Such explanatory variables of compounds are represented by molecular descriptors, which are correlated with functional moieties present in the chemical structure. Descriptors can be calculated directly from the chemical structure, although in some circumstances some form of geometry optimization is needed prior to obtaining the molecular features. Depending on the software used, the number of descriptors can span thousands of collinear and redundant descriptors, a situation that would typically call for the use of feature selection. Structure-activity/property relationships can then be discerned from traditional multiple linear regression as well as from a wide array of available machine learning algorithms. The reliability of constructed QSAR/QSPR models could be assessed from their internal and external validation, statistical measures of predictive performance, Y-randomization experiments, etc. Assessment of the applicability domain of the constructed QSAR/QSPR models may also provide useful knowledge of the range or scope of compounds that are covered by the model, since the model is only as good as the compounds used to train it. The aforementioned procedures are common steps in a typical QSAR/QSPR workflow as summarized in Fig. 1; further details are discussed in previous review articles [17, 18].

As noted previously, the early days of QSAR/QSPR were predominantly concerned with its utilization for investigating toxicity and the narcotic effects of compounds. Over time, this use had expanded to encompass a wide range of biological activities and chemical properties, as summarized in Table 1. In a simplistic system, compounds are isolated in the sense that they act or behave a certain way (e.g., as revealed by their boiling point or melting point) owing directly to their unique chemical structure. To complicate the matter, it may be the case that minor changes to the chemical structure may exert a drastic influence on its observed property. This issue is known as the structure-activity cliff and had been documented previously [19, 20]. The underlying reason for this is that a majority of biological activities arises from the interaction of compounds with target proteins. A minor change in the chemical structure, such as the replacement of a methyl group by an ethyl group, may then cause steric clashes with an amino acid side chain inside the protein's binding pocket and thus radically alter the activity of the compound.

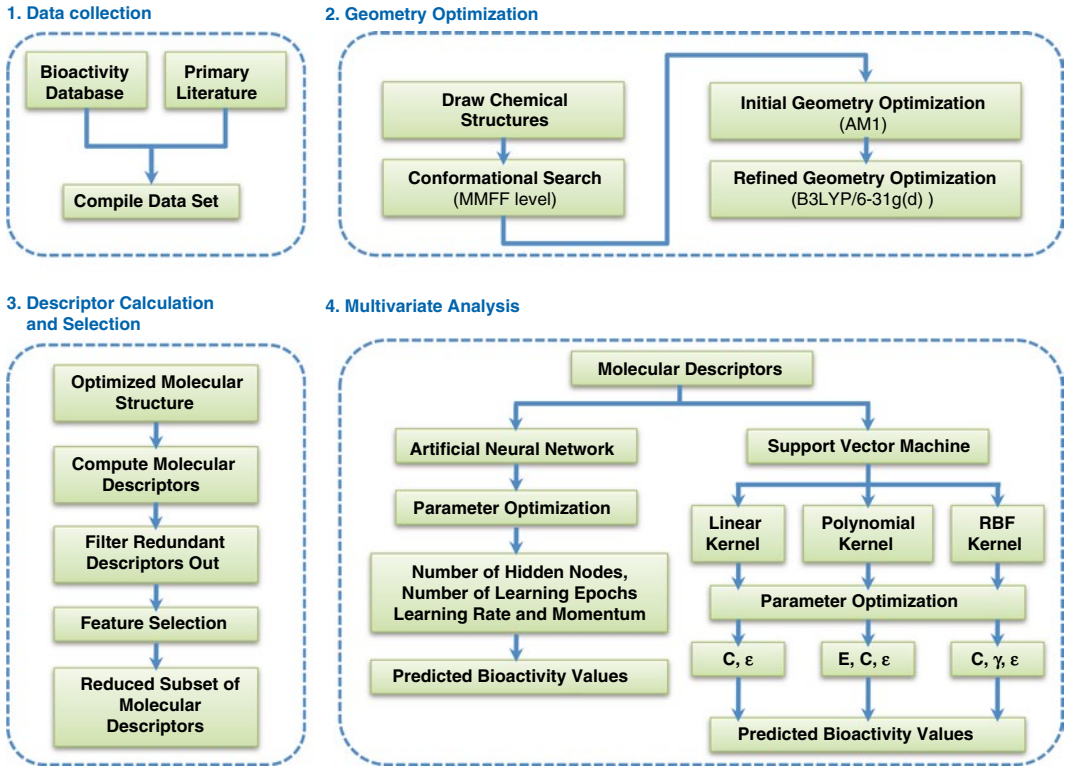


Fig. 1 Schematic representation of the QSAR/QSPR workflow

Although QSAR/QSPR may afford a wide range of utility in the life sciences, there are several potential pitfalls that practitioners need to be aware of. As this topic is beyond the scope of this chapter, readers are directed to previously reported accounts of such flaws [21–25] and solutions [26–29]. In particular, Dearden et al. [30] discuss 21 common types of error encountered in the QSAR/QSPR literature; they also provide recommendations on how to tackle such problems.

Efforts to automate some aspects of the QSAR workflow first appeared in 2001 when Jewell et al. [31] performed automatic generation of alignments for 3D-QSAR studies. A subsequent work from Tervo et al. [32] compared the performance of manual and automatic alignments. The first automated PLS-based feature selection was reported in 2004 by Olah et al. [33] in their exhaustive analysis of biological data from WOMBAT. A year later, Bhonsle et al. [34] reported an automated quasi-4D-QSAR study of CXCR4 inhibitors based on PLS and scripting language. In the same year, Cartmell et al. [35] described an automated QSPR workflow to investigate ADME data sets based on a software architecture called competitive workflow as implemented in the Discovery Bus. Zhang et al. [36] introduced the Automated Lazy

Table 1
Types of biological activities modeled by QSAR and selected examples

Type	Examples
Absorption	Blood-brain barrier penetration Human intestinal absorption P-glycoprotein Skin permeability
Distribution	Aqueous solubility (logS) Octanol-water partition coefficient (logP) Octanol-water distribution coefficient (logD)
Metabolism	Cytochrome P450 induction/inhibition
Excretion	Hepatic microsomal intrinsic clearance
Toxicity	AMES mutagenicity Carcinogenicity Hepatotoxicity Skin sensitivity Target/organ cytotoxicity
Receptor agonist/antagonist	A3 adenosine receptor antagonist Angiotensin II receptor antagonist CCR5 receptor antagonist Glucagon receptor antagonist Peroxisome proliferator-activated receptor gamma agonists
Enzyme activator/inhibitor/modulator	Aromatase inhibitor Dipeptidyl peptidase IV inhibitor Gamma-aminobutyric acid modulator Gamma-secretase modulator Glucokinase activator

Learning QSAR (ALL-QSAR) modeling approach in 2006 as applied to three sets of anticonvulsant agents. Subsequently in 2007, Bhonsle et al. [37] reported a semiautomated QSAR workflow using Tcl-based Cerius2 scripts in their investigation of a set of insect repellants. Furthermore, Obrezanova et al. [38] described an automated QSAR modeling approach of ADME data sets using the Gaussian process. Moreover, Rodgers et al. [39] introduced an approach to automatically update the QSAR model with measured data of new compounds in their human plasma protein binding model. In 2008, in a continuation of their earlier work, Obrezanova et al. compared the results from automated QSAR models with those of manual efforts in their study of blood-brain barrier penetration and aqueous solubility. In the same year, Ma et al. [40] reported an inductive data mining approach in the automatic generation of decision trees for modeling of the ecotoxicity of chemicals as well as in the analysis of a historical data from wastewater

treatment plant. In 2011, Wood et al. [41] presented an automated QSAR procedure employed in AstraZeneca's AutoQSAR system as tested against three properties comprising of log*D*, solubility, and human plasma protein binding. Furthermore, Sushko et al. introduced the Online Chemical Modeling Environment (OCHEM) as a web-based tool for automating the process of QSAR modeling. In 2012, Pérez-Castillo et al. [42] described the genetic-algorithm-(meta)-ensembles approach for binary classification of five data sets from the literature. In 2013, Cox et al. [43] reported the QSAR Workbench, which is based on the Pipeline Pilot workflow tool and evaluated against two public domain data sets. Furthermore, Martins and Ferreira [44] introduced software called QSAR modeling for generating and validating QSAR models.

As we can see, development of QSAR models may not be a straightforward task, especially for the non-bioinformatician, and this is concomitant with the fact that efforts to automate the QSAR/QSPR process are an active area of research. These and many other factors sparked our interests and motivated us to develop AutoWeka starting from late 2009, which we have been coding ever since. The project began as an effort to automate our own data mining workflow as applied to QSAR/QSPR modeling and finally became publicly and freely available in 2012. The data mining capabilities, particularly through the use of artificial neural networks and support vector machines, of AutoWeka are powered by the popular and widely used Weka machine learning package [45]. It should be noted at the outset that the AutoWeka software is used for the multivariate analysis phase of QSAR/QSPR modeling, for which it also performs an extensive parameter optimization, the results of which could be scrutinized and selected for further rounds of computation (*see* **Note 1**). Nevertheless, specific details of chemical structure drawing and molecular descriptor generation are also mentioned in this chapter. It is also worthy of mention that this chapter will not cover the software development of AutoWeka, which will be discussed elsewhere.

In the following sections, we will provide an example of the practical use of AutoWeka in constructing QSAR/QSPR models (*see* **Note 2**). This is demonstrated in a step-by-step manner using a set of curcumin analogs as a case study.

2 Materials

2.1 Data Source

There are several ways in which one can acquire the data that is to be used for QSAR/QSPR modeling. It can be measured data from one's own experiments, compiled from the primary literature, retrieved from curated databases, or even all of the above. In most cases, accessibility to the primary literature is heavily dependent on institutional or personal subscription, while there are a growing number of open access journals that are freely available for readers.

Typical journals describing the implementation of QSAR/QSPR models include but are not limited to the following:

- *Bioinformatics*
- *Bioorganic & Medicinal Chemistry*
- *Bioorganic & Medicinal Chemistry Letters*
- *BMC Bioinformatics*
- *Briefings in Bioinformatics*
- *Chemical Biology & Drug Design*
- *Chemometrics and Intelligent Laboratory Systems*
- *Chemosphere*
- *Computational and Theoretical Chemistry* (formerly *Journal of Molecular Structure: THEOCHEM*)
- *Computational Biology and Chemistry*
- *European Journal of Medicinal Chemistry*
- *International Journal of Quantum Chemistry*
- *Journal of Chemical Information and Modeling*
- *Journal of Cheminformatics*
- *Journal of Chemistry*
- *Journal of Chemometrics*
- *Journal of Computational Biology*
- *Journal of Computational Chemistry*
- *Journal of Computer-Aided Molecular Design*
- *Journal of Enzyme Inhibition and Medicinal Chemistry*
- *Journal of Medicinal Chemistry*
- *Journal of Molecular Graphics and Modelling*
- *Journal of Molecular Modeling*
- *Journal of Theoretical and Computational Chemistry*
- *Letters in Drug Design & Discovery*
- *Medicinal Chemistry Research*
- *Molecular Informatics*
- *Molecular Simulation*
- *PLoS Computational Biology*
- *SAR and QSAR in Environmental Research*

The following databases contain curated bioactivity data that have either been deposited by the laboratory generating the data or compiled from the literature:

- BindingDB—a database containing ~1,000,000 pieces of interaction data for ~6,500 protein targets and ~427,000 small molecules. It is accessible at http://www.bindingdb.org.

- ChEMBL—a database containing ~12,000,000 pieces of interaction data for ~9,000 protein targets and ~1,500,000 small molecules. It is accessible at https://www.ebi.ac.uk/chembldb.
- DrugBank—a database containing ~6,800 drug entries spanning FDA-approved small molecules, FDA-approved peptide/protein drugs, nutraceuticals, and experimental drugs. It is accessible at http://www.drugbank.ca.
- PubChem—is comprised of three linked databases spanning substance, compound, and bioassay. It is accessible at http://pubchem.ncbi.nlm.nih.gov.
- WOMBAT—a database containing ~79,000 in vivo measurement data, ~330,000 compounds, and ~1,900 protein targets. It is commercially available at http://www.sunsetmolecular.com.

2.2 Drawing and Refining Chemical Structures

Chemical structures can be drawn into the computer using any of the following software tools:

- Accelrys Draw (available at http://www.accelrys.com)
- MarvinSketch (available at http://www.chemaxon.com)
- OpenEye VIDA (available at http://www.eyesopen.com)

Chemical structure file format conversion tools of use include:

- Babel (available at http://www.eyesopen.com)
- Open Babel (available at http://www.openbabel.org)

Molecular structures can be refined using online tools:

- CORINA Online Demo (available at http://www.molecular-networks.com/online_demos/corina_demo)
- COSMOS (accessible at http://cosmos.igb.uci.edu)

In certain cases where the molecular structure is relevant or of importance for use in prediction, it can also be subjected to geometry optimization via quantum chemical calculations:

- Gaussian (commercially available at http://www.gaussian.com)
- GAMESS (available at http://www.msg.ameslab.gov/gamess)
- HyperChem (commercially available at http://www.hyper.com)
- MOLCAS (commercially available at http://www.molcas.org)
- MOPAC (available at http://openmopac.net)

2.3 Calculating Molecular Descriptors

The next step is to generate numerical descriptions of compounds to be investigated; a wide range of software is available to perform this task. For example, common compounds may be available from

the MOLE-db website (Accessible at http://michem.disat.unimib.it/mole_db). Moreover, quantum chemical descriptors can be obtained from the aforementioned software such as Gaussian, GAMESS, etc. Several other categories of molecular descriptors could be obtained from the following software:

- CDK Descriptor GUI (available at http://www.rguha.net/code/java/cdkdesc.html)
- ChemAxon JChem Calculator Plugins (available at http://www.chemaxon.com/jchem)
- Dragon (commercially available from http://www.talete.mi.it)
- E-Dragon (accessible at http://www.vcclab.org/lab/edragon)
- MODEL (accessible at http://jing.cz3.nus.edu.sg/cgi-bin/model/model.cgi)
- PaDEL (available at http://padel.nus.edu.sg)

2.4 Data Compilation

Prior to constructing the QSAR/QSPR model, the block of **X** molecular descriptors and **Y** bioactivity values are combined and prepared in ARFF file format for use as input to AutoWeka.

- Spreadsheet program
  - Microsoft Excel (commercially available at http://office.microsoft.com/excel)
  - OpenOffice (freely available at http://www.openoffice.org)
- Text editor
  - Notepad++ (freely available at http://notepad-plus-plus.org) EditPad Lite (freely available at http://www.editpadlite.com)
- CSV to ARFF converter
  - csv2arff (accessible from http://slavnik.fe.uni-lj.si/markot/csv2arff/csv2arff.php)

2.5 Multivariate Analysis

It should be noted here that multivariate analysis or the actual process of constructing the QSAR model is performed using AutoWeka (Freely available at http://www.mt.mahidol.ac.th/autoweka). It is in this phase that parameter optimization takes place in an automated fashion. Completed results will contain information on the best set of parameters for the selected machine learning algorithm.

2.6 Plotting Graphs

After completing AutoWeka calculations, results from parameter optimization could be visualized using the Python scripts available in the Download section at http://www.mt.mahidol.ac.th/autoweka. Alternatively, plots of the results could also be created using statistical graphical software such as SigmaPlot (Commercially available from http://www.sigmaplot.com).

3 Methods

3.1 Data Source

At this phase, the practitioner decides on the data to be used in their QSAR/QSPR project and therefore retrieves the necessary information (i.e., chemical name, SMILES, and bioactivity values) from resources mentioned in Subheading 2.1; the data are then compiled as outlined in Subheading 3.4. In the case study described in this chapter, we will be using a data set of 22 curcumin analogs with DPPH free radical-scavenging activity as previously reported by Venkateswarlu et al. [46] and modeled by Worachartcheewan et al. [47]. The bioactivity values were binned to the binary labels "low" and "high" activity, denoting compounds affording IC_{50} values of greater than and less than 10 μM, respectively. It should be noted that in the previously reported QSAR modeling study described by Worachartcheewan et al. [47], three compounds (nos. 5, 6, and 10) were identified as outliers and subjected to removal from the data set, thereby reducing it to 19 compounds.

3.2 Drawing and Refining Chemical Structures

Chemical structures of curcumin analogs (Fig. 2) are drawn into the computer using software described in Subheading 2.2. Subsequently, chemical structures are either subjected to a rough energy minimization (using a molecular mechanics force field) or a more extensive quantum chemical calculation (using a tool such as HF, B3LYP, etc.) in order to obtain a more refined structure. These structures can be subjected to an initial geometry refinement using the semiempirical Austin Model 1 (AM1) followed by Becke's three-parameter hybrid method using the Lee-Yang-Parr correlation functional (B3LYP) along with 6-31g(d) basis set as performed previously by Worachartcheewan et al. [47].

3.3 Calculating Molecular Descriptors

Molecular descriptors can be derived from unrefined chemical structures, which may be applicable for 1D and 2D descriptors. However, in order to derive 3D descriptors, the chemical structures must first be subjected to geometry optimization using quantum chemical calculations as noted above. Depending on the software used to generate the molecular descriptors, this can lead to the production of hundreds or thousands of descriptors. It is common to observe a high degree of collinearity or redundancy among these descriptors, and this issue is best handled by performing feature selection to select a smaller subset of informative descriptors for further multivariate analysis. As this issue is beyond the scope of this chapter, readers are directed to previous review articles.

3.4 Data Compilation

One of the first phases of a QSAR/QSPR study is compiling the data set of interest, which can be performed by entering essential data into an appropriate spreadsheet program. For example, a typical

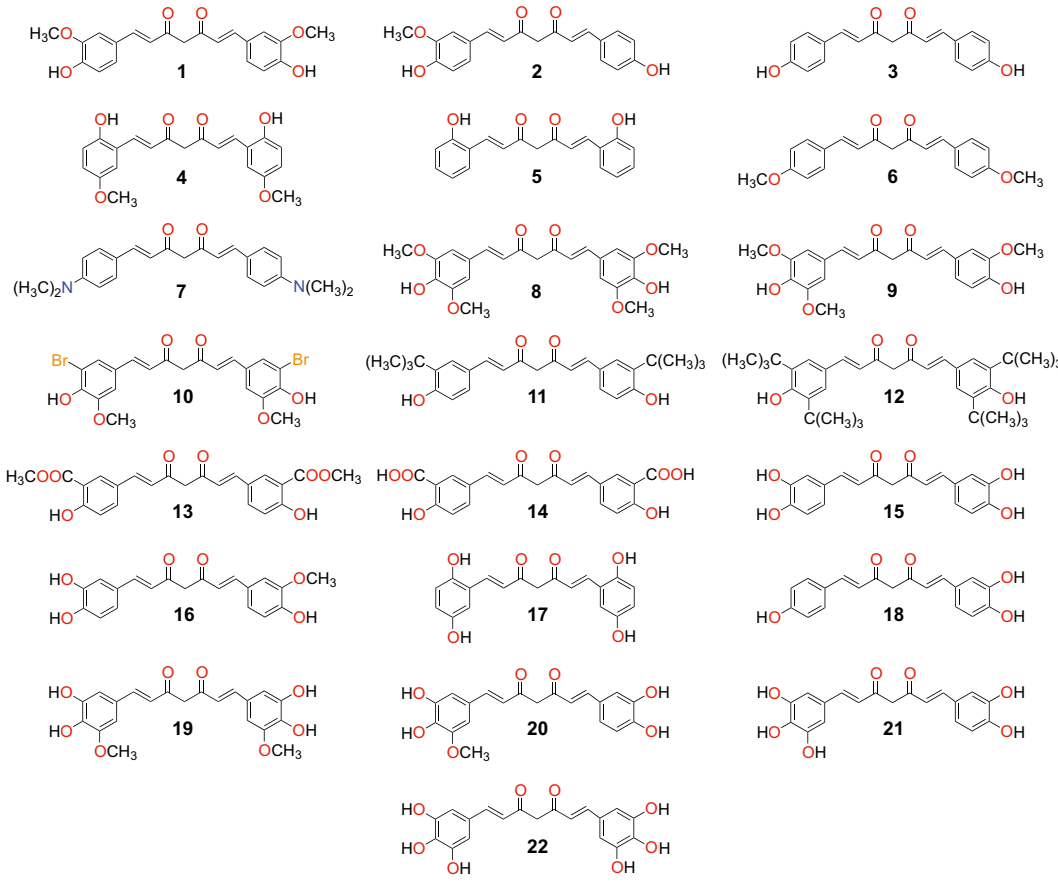
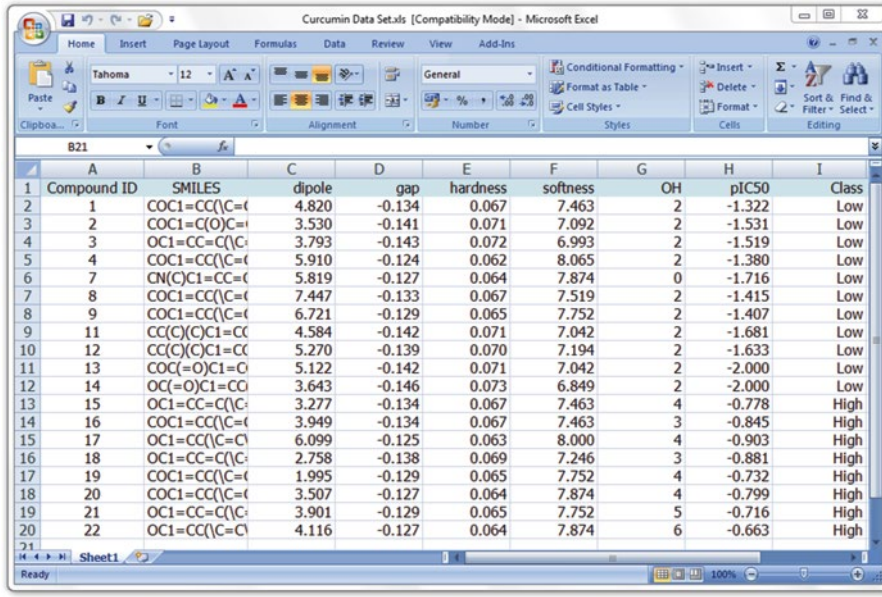


Fig. 2 Chemical structures of 22 curcumin analogs

molecule or compound will occupy a row in the spreadsheet where the columns can contain the following information: compound name, compound ID, XC_{50} of bioactivity (where X refers to the type of bioactivity such as inhibition concentration, effective concentration, etc.), SMILES notation, molecular descriptors (molecular weight, logP, number of hydrogen bond donor atoms, number of hydrogen bond acceptor atoms, etc.), and the reference or data source from which the compounds were obtained (Fig. 3). Therefore, as there are 19 curcumin analogs (after removal of three outliers) for this case study, this would correspond to 19 rows plus the header row (containing the descriptor labels of each column) resulting in a total of 20 rows.

The next step is to transform our data set from the spreadsheet format to an ARFF file format that is compatible with Weka, since this is the machine learning package that powers the software developed in AutoWeka. This spreadsheet to ARFF transformation can be carried out using a text editor or alternatively using the online csv2arff tool. Examples of ARFF input files are shown in



	A	B	C	D	E	F	G	H	I
1	Compound ID	SMILES	dipole	gap	hardness	softness	OH	pIC50	Class
2	1	COC1=CC(\C=	4.820	-0.134	0.067	7.463	2	-1.322	Low
3	2	COC1=C(O)C=	3.530	-0.141	0.071	7.092	2	-1.531	Low
4	3	OC1=CC=C(\C	3.793	-0.143	0.072	6.993	2	-1.519	Low
5	4	COC1=CC(\C=	5.910	-0.124	0.062	8.065	2	-1.380	Low
6	7	CN(C)C1=CC=	5.819	-0.127	0.064	7.874	0	-1.716	Low
7	8	COC1=CC(\C=	7.447	-0.133	0.067	7.519	2	-1.415	Low
8	9	COC1=CC(\C=	6.721	-0.129	0.065	7.752	2	-1.407	Low
9	11	CC(C)(C)C1=CC	4.584	-0.142	0.071	7.042	2	-1.681	Low
10	12	CC(C)(C)C1=CC	5.270	-0.139	0.070	7.194	2	-1.633	Low
11	13	COC(=O)C1=C	5.122	-0.142	0.071	7.042	2	-2.000	Low
12	14	OC(=O)C1=CC	3.643	-0.146	0.073	6.849	2	-2.000	Low
13	15	OC1=CC=C(\C	3.277	-0.134	0.067	7.463	4	-0.778	High
14	16	COC1=CC(\C=	3.949	-0.134	0.067	7.463	3	-0.845	High
15	17	OC1=CC(\C=C	6.099	-0.125	0.063	8.000	4	-0.903	High
16	18	OC1=CC=C(\C	2.758	-0.138	0.069	7.246	3	-0.881	High
17	19	COC1=CC(\C=	1.995	-0.129	0.065	7.752	4	-0.732	High
18	20	COC1=CC(\C=	3.507	-0.127	0.064	7.874	4	-0.799	High
19	21	OC1=CC=C(\C	3.901	-0.129	0.065	7.752	5	-0.716	High
20	22	OC1=CC(\C=C	4.116	-0.127	0.064	7.874	6	-0.663	High
21									

Fig. 3 Screenshot of compiled data using Microsoft Excel

Tables 2 and 3 with quantitative and qualitative **Y** labels. Line 1 represents the name of the data set, lines 3–8 represent the descriptor names, and lines 11–29 represent the block of descriptors and their **Y** values or labels as correspondingly specified by the following terms with the @ symbol preceding it: RELATION, ATTRIBUTE, and DATA. The NUMERIC terms that follow the descriptor names in lines 3–8 are syntax that are recognized by the Weka program as descriptors having quantitative values, whereas the braces {} encapsulating the Low and High terms are also syntax that are recognized by the Weka program as descriptors having qualitative values. It should be noted that the **Y** descriptor is typically located as the last variable, whereas **X** descriptors precede it. Generally, QSAR modeling of data sets having quantitative or qualitative **Y** variables is subjected to either regression or classification analysis, respectively.

3.5 Machine Learning Algorithms

Before we move on to the multivariate analysis phase and actually construct the QSAR model, it is pertinent to first provide a glimpse of the algorithmic details of machine learning algorithms that are commonly used in QSAR/QSPR on biological [47–56] and chemical systems [57–61] and which are implemented in AutoWeka.

3.5.1 Artificial Neural Network

The artificial neural network (ANN) is a well-known multivariate method that is commonly used to develop QSAR/QSPR models. ANN takes its inspiration from the biological brain with its organization of neurons [62]. The single-layer perceptron (SLP), which

Table 2
Example of ARFF input file of curcumin analogs with quantitative Y variable

@RELATION RadicalScavengingRegression
@ATTRIBUTE dipole NUMERIC
@ATTRIBUTE gap NUMERIC
@ATTRIBUTE hardness NUMERIC
@ATTRIBUTE softness NUMERIC
@ATTRIBUTE OH NUMERIC
@ATTRIBUTE pIC50 NUMERIC
@DATA
4.82,−0.134,0.067,7.463,2,−1.322
3.53,−0.141,0.071,7.092,2,−1.531
3.793,−0.143,0.072,6.993,2,−1.519
5.91,−0.124,0.062,8.065,2,−1.38
5.819,−0.127,0.064,7.874,0,−1.716
7.447,−0.133,0.067,7.519,2,−1.415
6.721,−0.129,0.065,7.752,2,−1.407
4.584,−0.142,0.071,7.042,2,−1.681
5.27,−0.139,0.07,7.194,2,−1.633
5.122,−0.142,0.071,7.042,2,−2
3.643,−0.146,0.073,6.849,2,−2
3.277,−0.134,0.067,7.463,4,−0.778
3.949,−0.134,0.067,7.463,3,−0.845
6.099,−0.125,0.063,8,4,−0.903
2.758,−0.138,0.069,7.246,3,−0.881
1.995,−0.129,0.065,7.752,4,−0.732
3.507,−0.127,0.064,7.874,4,−0.799
3.901,−0.129,0.065,7.752,5,−0.716
4.116,−0.127,0.064,7.874,6,−0.663

Table 3
Example of ARFF input file of curcumin analogs with qualitative Y variable

```
@RELATION
  RadicalScavengingClassification

@ATTRIBUTE dipole NUMERIC
@ATTRIBUTE gap NUMERIC
@ATTRIBUTE hardness NUMERIC
@ATTRIBUTE softness NUMERIC
@ATTRIBUTE OH NUMERIC
@ATTRIBUTE class {Low, High}

@DATA
4.82,-0.134,0.067,7.463,2,Low
3.53,-0.141,0.071,7.092,2,Low
3.793,-0.143,0.072,6.993,2,Low
5.91,-0.124,0.062,8.065,2,Low
5.819,-0.127,0.064,7.874,0,Low
7.447,-0.133,0.067,7.519,2,Low
6.721,-0.129,0.065,7.752,2,Low
4.584,-0.142,0.071,7.042,2,Low
5.27,-0.139,0.07,7.194,2,Low
5.122,-0.142,0.071,7.042,2,Low
3.643,-0.146,0.073,6.849,2,Low
3.277,-0.134,0.067,7.463,4,High
3.949,-0.134,0.067,7.463,3,High
6.099,-0.125,0.063,8,4,High
2.758,-0.138,0.069,7.246,3,High
1.995,-0.129,0.065,7.752,4,High
3.507,-0.127,0.064,7.874,4,High
3.901,-0.129,0.065,7.752,5,High
4.116,-0.127,0.064,7.874,6,High
```

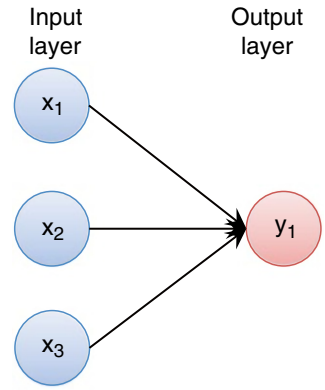


Fig. 4 Schematic architecture of the single-layer perceptron

is the classical model of ANN, is the simplest type of ANN; it is comprised of several input nodes and a single output node as shown in Fig. 4.

For a given training sample $D=(x_1,y_1),(x_2,y_2),\ldots,(x_N,y_N)$, we can estimate the variable y_i by combining the weighted sum of its N inputs as follows:

$$y_i = \theta\left(\sum_{i=1}^{N} w_i x_i\right) \tag{1}$$

where y_i belongs to class 1 if the weighted sum is greater than the selected threshold; otherwise, y_i belongs to class 0. $\theta(\bullet)$ is the activation function that maps the weighted sum of inputs to the output. The most popularly currently used activation functions are the logistic sigmoid ($1/1+e^{-x_i}$) and hyperbolic tangent ($\tanh(x_i)$). In practical application, SLP is unable to solve a nonlinearly separable data problem, owing to the fact that during the learning process, this approach cannot properly predict all data points on D. Thus, the multilayer perceptron (MLP) was proposed for handling nonlinearly separable data by adding one or more hidden layers (*see* Fig. 5) [63, 64].

The MLP approach is a type of supervised learning method in which the back-propagation algorithm is applied to estimate the optimized parameter w_i by changing the weighted connection, which is dependent on the magnitude of the error (the difference between the actual and predicted value). The back-propagation algorithm is comprised of two major phases: (1) forward phase and (2) backward phase, as briefly described below.

Step 1. Initialize a connection weight w_i with a small random value.

Step 2. Randomly select a training sample $D_i \subset D$, $|D_i| = p$ where $p < N$.

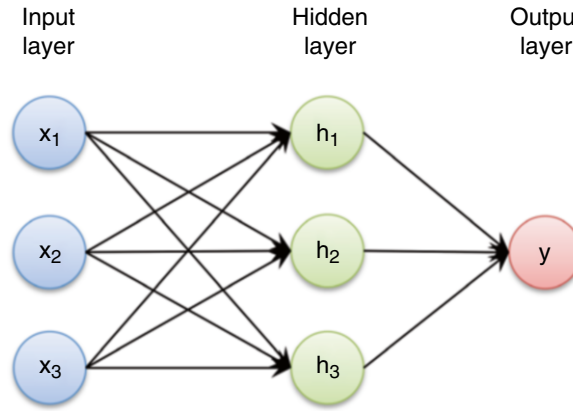


Fig. 5 Schematic architecture of the multilayer perceptron

Step 3. Calculate the partial derivative of weight w_i.

Step 4. Update weight w_i according to the following equations:

$$w_{ij}(n+1) = w_{ij}(n) + \eta \frac{\partial E(n)}{\partial w_{ij}} \tag{2}$$

$$w_{ij}(n+1) = w_{ij}(n) + \eta \frac{\partial E}{\partial w_{ij}} \tag{3}$$

where $w_{ij}(n)$ is the value of w_{ij} prior to updating by n times while $w_{ij}(n+1)$ is the value of w_{ij} after such updates and η is the learning rate that determines both the convergence rate and stability of the training process while E is the cost or error function.

Step 5. Repeat **steps 2–4** for every training sample D_i, and repeat for these sets until the cost of output errors is minimized.

3.5.2 Support Vector Machine

The support vector machine (SVM) is a popular learning method that had been adopted for solving a plethora of problems via classification and regression analysis. A notable property of SVM is its estimation of model parameters by means of the convex optimization approach that guarantees that a local solution is also the global optimum [65]. Vapnik first introduced this technique based on the principles of structure risk minimization of the statistical learning theory [66, 67]. In practice, the SVM method constructs a maximum-margin hyperplane to separate two classes. To solve nonlinearly separable data, the SVM method acts in conjunction with a mapping function $\Phi(x): x \in R^M \rightarrow R^P$ that is used to transform the original data set of M-dimension onto a higher dimensional space or feature space of P-dimension where $M \ll P$. Subsequently, a simple linear classifier $f(x_i)$ can then be used for classifying P-dimensional samples [68, 69] (shown in Fig. 6).

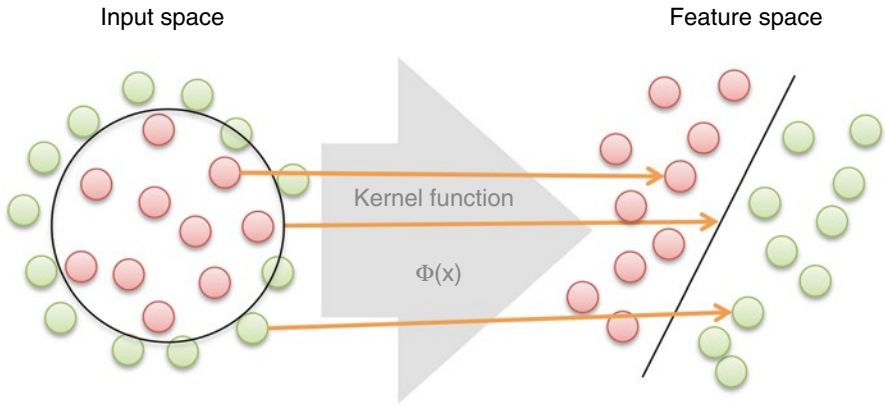


Fig. 6 Schematic representation of relationship between input and feature spaces using a mapping function

The kernel function $K(x_i, x_j)$ is used to represent the mapping function by taking the inner product between two samples x_i and x_j in D, which is defined as:

$$K\left(x_i, x_j\right) = \Phi\left(x_i\right)^T \Phi\left(x_j\right) = \sum_{i,j=1}^{N} \Phi\left(x_i\right)^T \Phi\left(x_j\right) \tag{4}$$

In practice, given D, SVM is used to construct a linear function $f(x_i)$ representing the correlation between the structure and biological activities/chemical properties of data x_i:

$$f\left(x_i\right) = \sum_{i=1}^{N} w_i \Phi\left(x_i\right) + b \tag{5}$$

where $w_i \in R^M$ are the coefficients, $b \in R$ is the bias, and N is the number of samples. The most popularly used kernel comprises the following:

- Linear kernel:

$$\Phi\left(x_i\right)^T \Phi\left(x_j\right) \tag{6}$$

- Polynomial kernel:

$$\left(1 + \Phi\left(x_i\right)^T \Phi\left(x_j\right)\right)^d \tag{7}$$

where $d = 2$, 3, and 4 (it should be noted that $d = 1$ for a linear kernel).

- Radial basis function (RBF) kernel:

$$\exp\left(-\gamma\left(x_i - x_j\right)\right) \tag{8}$$

where γ is greater than 0.

Table 4
Summary of computational methods used in QSAR/QSPR modeling

Method	Advantage	Disadvantage	Built-in feature selector	Single/ ensemble
ANN	Performs well on complex data	Low interpretability	No	Single
SVM	Solves nonlinearly separable data	Low interpretability	No	Single
PCA	Summarizes a data set without losing too much variation	Does not consider relationship between X and Y	Yes	Single
PLS	Simple and interpretable model	Requires cross-term	Yes	Single
DT	Simple and interpretable model	Requires a number of training data sets	Yes	Single
RF	High interpretability and low risk of over-fitting	Complexity method	Yes	Ensemble

ANN, SVM, PCA, PLS, DT, and RF are acronyms for artificial neural network, support vector machine, principal component analysis, partial least squares regression, decision tree, and random forest, respectively

In regression problems, when y is a numerical value, estimation of the parameter w_i can be achieved by utilizing the ε-insensitive loss function ($L_\varepsilon(y, f(x, w))$) [70, 71] described as follows:

$$L_\varepsilon\left(y, f\left(x,w\right)\right) = \begin{cases} \left| y - f\left(x,w\right)\right| - \varepsilon, & y - f\left(x,w\right) \geq \varepsilon \\ 0, & y - f\left(x,w\right) < \varepsilon \end{cases} \quad (9)$$

where y is the actual value, $f(x, w)$ is the predicted function for estimating the output value, and ε is the insensitivity parameter. Although SVM is widely popular, SVM and ANN methods both suffer from the fact that they are a black-box approach and therefore lack interpretability; they do not readily indicate which feature(s) is of greatest importance in the structure of the prediction model. Table 4 highlights and compares the advantages and disadvantages of SVM and ANN in the context of other commonly used learning algorithms in QSAR/QSPR modeling.

3.6 Multivariate Analysis

We will now proceed with the step-by-step case study of constructing the QSAR model using AutoWeka. Although we try to provide as much detail as possible on the practical aspects of using AutoWeka, some adjustments may need to be made as the reader adapts this protocol to their own projects. As previously mentioned in Subheading 3.1, the case study used in this chapter is based on the set of 22 curcumin analogs reported by Venkateswarlu et al. [46].

Let us start by going to the website of AutoWeka that is accessible at http://www.mt.mahidol.ac.th/autoweka. Readers can click on either the Download link on top or the orange Download button down below (Fig. 7).

Fig. 7 Screenshot of AutoWeka's website

On the following page, readers have access to the AutoWeka User Manual, the Python scripts for creating graphs and plots, as well as the AutoWeka software. After agreeing to the terms of the license agreement, click on the "*I Agree, Please Proceed to Download AutoWeka*" button (Fig. 8) to proceed. The next page will then bring out a registration form that users can fill out, and after its submission, the Download link will appear. The zip file of the software is approximately 19 MB in size. After successfully downloading the file, unzip it to a desired location, and the following contents as shown in Fig. 9 (left panel) will appear. After double-clicking on the *AutoWeka.exe* file (left panel) the program window will appear (right panel) as shown in Fig. 9. The menu bar contains three possible options, *Run*, *Tools*, and *About*, which correspondingly allow users to run the machine learning algorithms for constructing the QSAR/QSPR model, adjust the memory value to use (Fig. 10), and provide access to the *About* window.

Now that the software is up and running, let us get started with setting up an ANN calculation (Fig. 11) by first clicking on *Run* → *Artificial Neural Network*. This brings up a new window that allows us to set up the various ANN parameters or choose the easiest option, which is to use the default parameters. Here we click on the *Browse* button and select the *ARFF* input file of interest (in our case, the *Curcumin_regression.arff*) file. Next, click on the *Default* button and finally the *Start* button to proceed with building the ANN model.

Fig. 8 Screenshot of the Download page on AutoWeka's website

Subsequently, a new window appears that summarizes the parameter settings that will be used for the calculation. Upon clicking on the *OK* button, a pop-up window asks for confirmation of the name of an automatically generated folder (Fig. 12). Here, we can click on the OK button again to use the default name. Next, the progress window (Fig. 13) appears; the time required for completion will depend on the complexity of the input data. Upon completion, a pop-up notification box appears, and we can then click on the OK button to finalize the calculation.

All calculation files are located in the *Results* folder (Fig. 14) that also goes by the same folder name in the root folder of AutoWeka, meaning that if we unpacked AutoWeka directly to the C drive, the relative path of the root folder would be *C:\AutoWeka* and thereby the Results folder could be found at *C:\AutoWeka\Results*. Inside the Results folder, double-click on the *Curcumin_analogs*

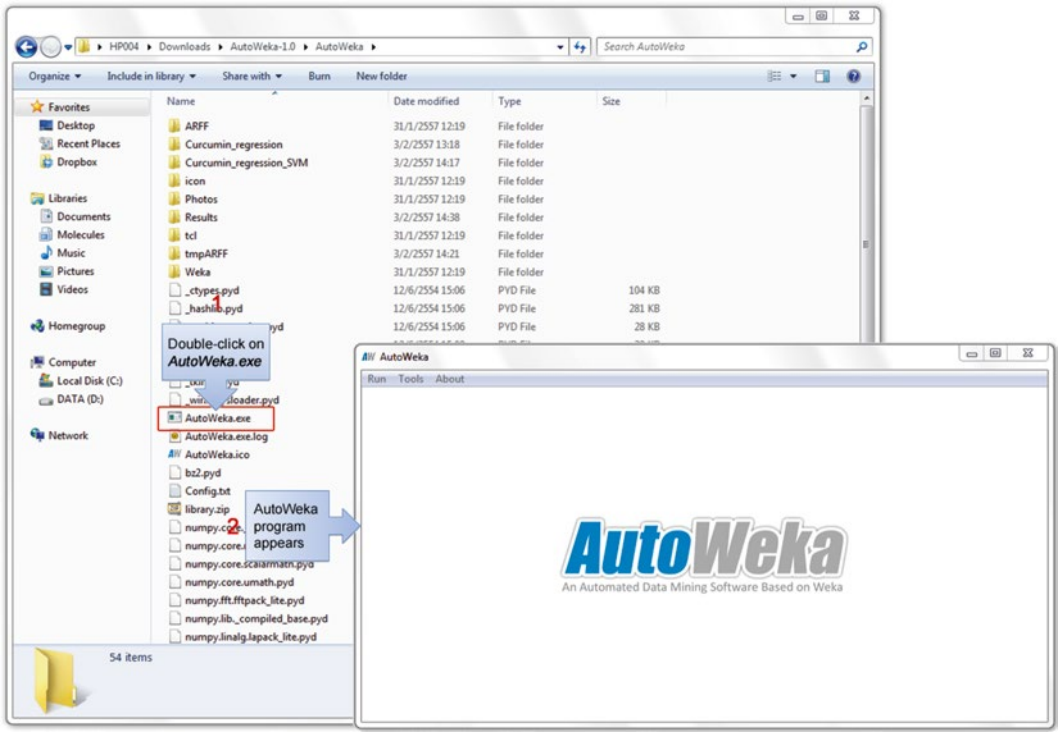


Fig. 9 Screenshot of the contents of the zip file of AutoWeka software (*left panel*) and upon opening the AutoWeka program (*right panel*)

folder to show a sub-folder that is automatically named according to the learning method used and the selected parameters. For example, if ANN was selected as the learning method and hidden nodes of 1–25 was chosen then the folder name would start with *ANN_Hidden_1_to_25*. The same convention also applies to the other parameters that will be appended at the end of the abovementioned folder name.

Inside the ANN results folder (Fig. 15), there are three sub-folders corresponding to the three ANN parameters, accordingly named as *HiddenNode*, *LearningAndMomentum*, and *TrainingTime*. Double-clicking on one of the sub-folders reveals a collection of sequentially numbered files where one parameter setting will generate one calculation output file. Thus, for an investigation of 25 hidden nodes (also bearing in mind that for each parameter investigated, ten separate runs are performed owing to the inherently random nature of the weight initialization of the back-propagation algorithm of ANN), a total of $25 \times 10 = 250$ output files will be generated. Subsequently, an average value for each investigated parameter will be derived from these ten calculations and saved into a new file called *AvgHidden.txt*. As data values within the AvgHidden.txt results file are in a tab-delimited format,

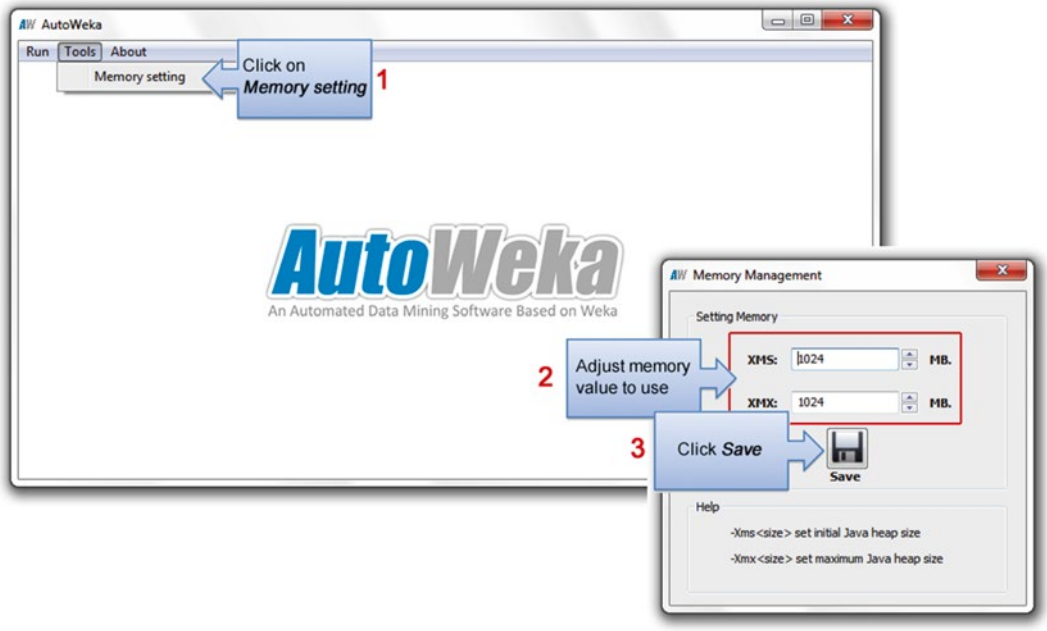


Fig. 10 Screenshot of adjusting the memory settings to be used by AutoWeka

Fig. 11 Screenshot of setting up an ANN calculation

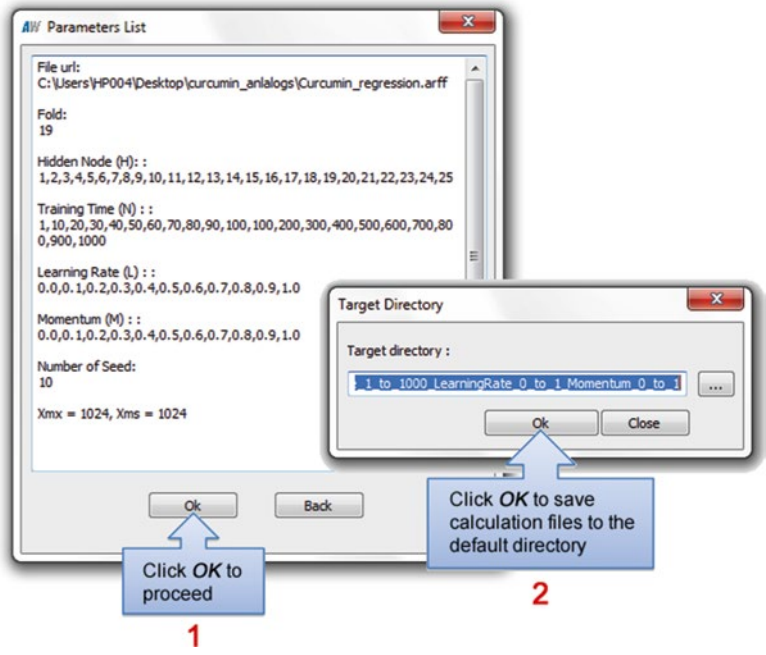


Fig. 12 Screenshot of pop-up windows that appear prior to ANN calculations

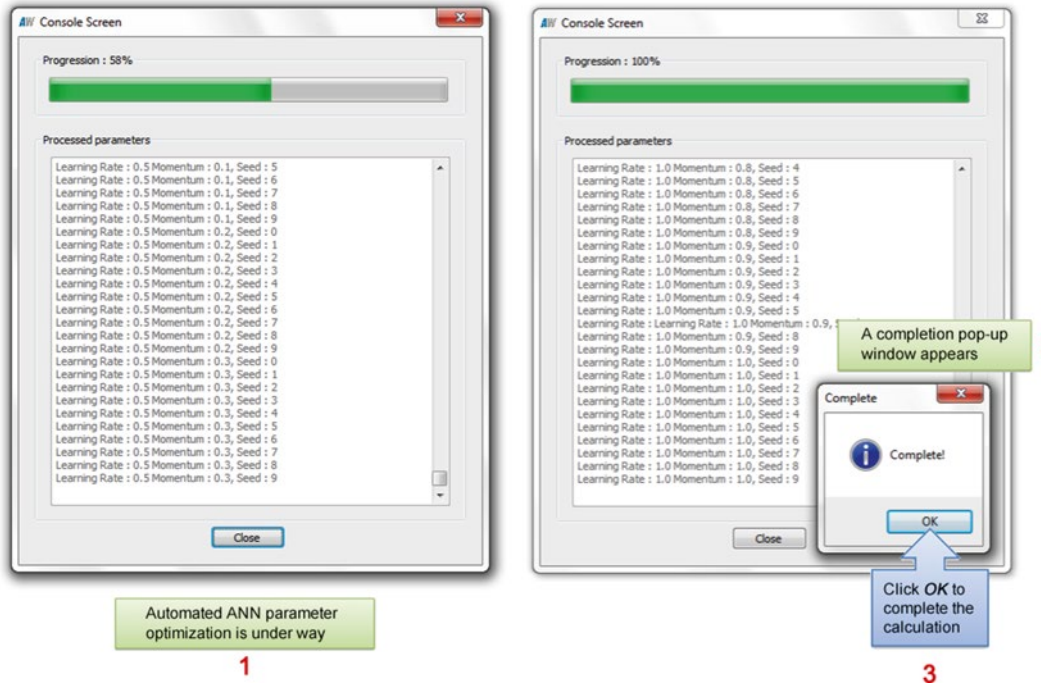


Fig. 13 Screenshot of an ongoing (*left panel*) and completed (*right panel*) ANN calculation

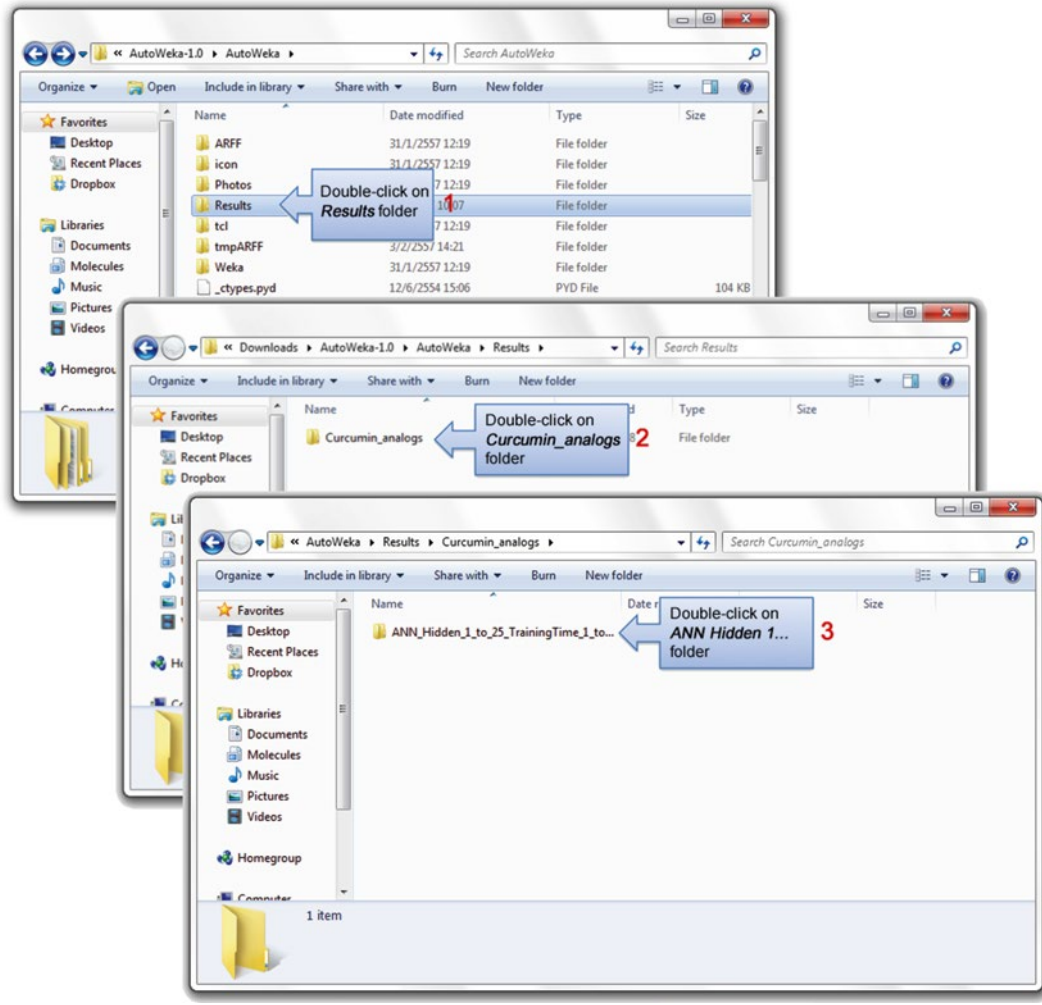


Fig. 14 Screenshot of the method for accessing the calculation results folder

it can be readily visualized by importing the file (or copying and pasting the data) into Microsoft Excel (Fig. 16). For convenience, the best parameter settings along with a summary of the statistical performance are provided in the *SummaryHidden.txt* file.

An alternative way of assessing the raw numerical data of the calculation results is to make a visual representation of it by creating graphical plots. This can be carried out by using the prewritten Python scripts or the graphical plot software mentioned in Subheading 2.6. Graphical plots created by the former are shown in Fig. 17.

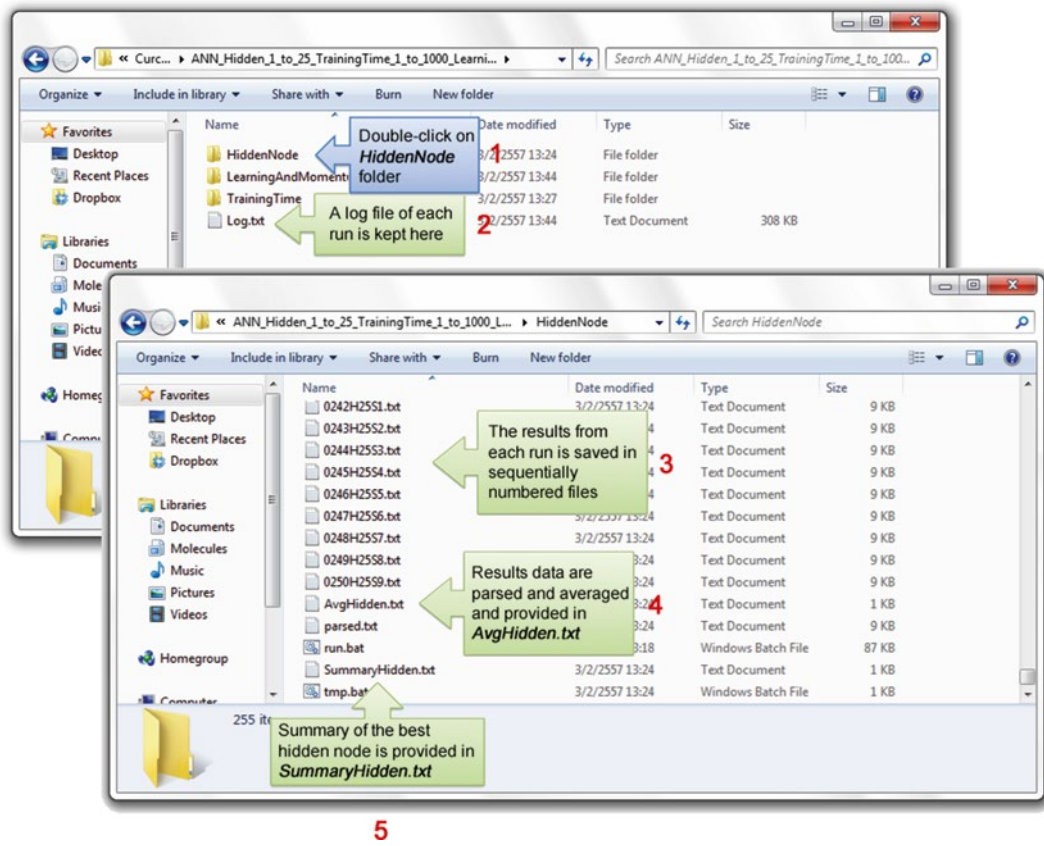
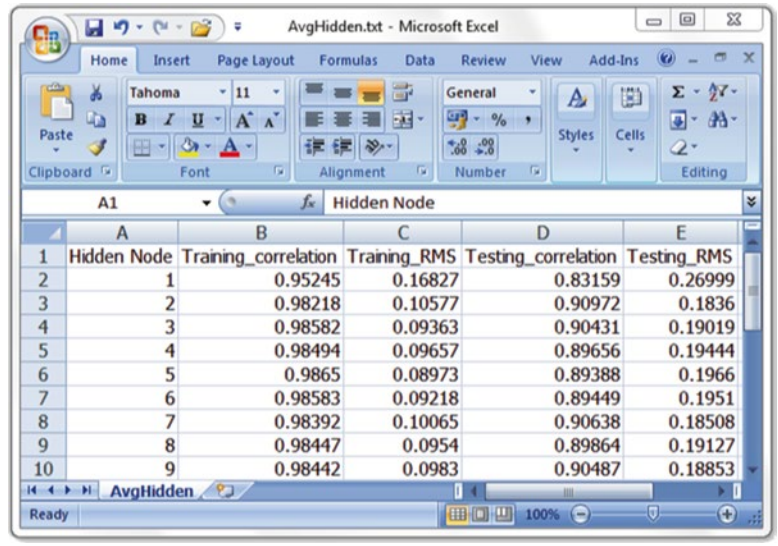


Fig. 15 Screenshot of calculation results as divided by the three folders of the optimized parameters: HiddenNode, LearningRateAndMomentum, as well as TrainingTime. Shown are contents from the AvgHidden.txt file found in the HiddenNode folder. It should be noted that the same structure and organization of calculation results apply to the other two folders



	A	B	C	D	E
1	Hidden Node	Training_correlation	Training_RMS	Testing_correlation	Testing_RMS
2	1	0.95245	0.16827	0.83159	0.26999
3	2	0.98218	0.10577	0.90972	0.1836
4	3	0.98582	0.09363	0.90431	0.19019
5	4	0.98494	0.09657	0.89656	0.19444
6	5	0.9865	0.08973	0.89388	0.1966
7	6	0.98583	0.09218	0.89449	0.1951
8	7	0.98392	0.10065	0.90638	0.18508
9	8	0.98447	0.0954	0.89864	0.19127
10	9	0.98442	0.0983	0.90487	0.18853

Fig. 16 Screenshot of AvgHidden.txt file as depicted in Microsoft Excel

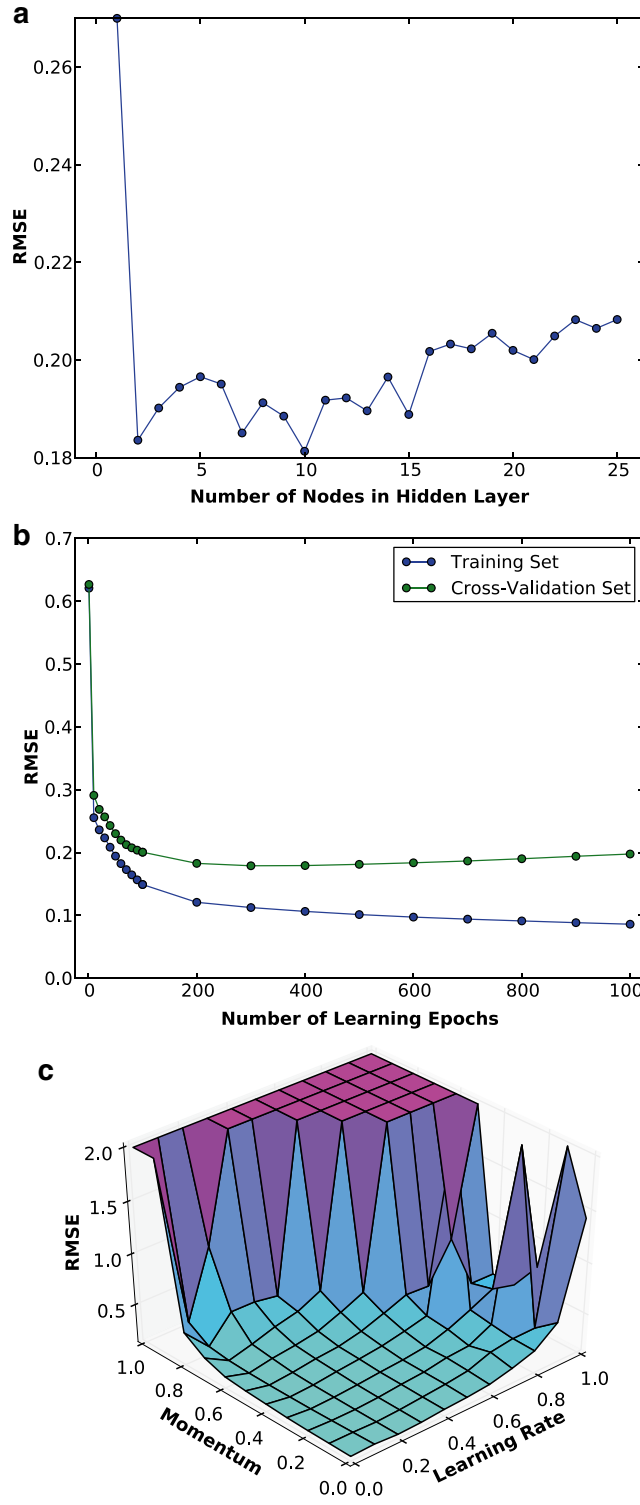


Fig. 17 Graphical plots from the parameter optimization process for (**a**) the number of hidden nodes, (**b**) the number of learning epochs, and (**c**) the learning rate and momentum

4 Notes

1. AutoWeka can be used in conjunction with Weka if users would like to benchmark their models with other learning algorithms.
2. In a typical QSAR modeling project, the rate-limiting step, that is, the lengthiest step, is generally that of data compilation and curation. It is here that critical errors (e.g., correctness of data entry) must be identified to ensure data integrity. The next most time-consuming step is the parameter optimization phase, in which several iterations of parameter fine-tuning are carried out to produce the best performance. AutoWeka handles the latter point automatically as it seeks the best set of parameters.

Acknowledgments

This work was supported by Mahidol University via the Goal-Oriented Research Grant to C.N.; postdoctoral fellowship to W.S.; research assistantships to P.M., S.J., and L.P.; and partial financial support to S.S.

References

1. Brodin A (1858) On the analogy of arsenic and phosphoric acid with respect to chemical and toxicology. Medico-Surgical Academy, St. Petersburg, Russia
2. Cros A (1863) Action de l'alcool amylique sur l'organisme. University of Strasbourg, Strasbourg
3. Kekulé A (1865) Sur la constitution des substances aromatiques. Bull Soc Chim Fr 3:98
4. Richardson B (1869) Physiological research on alcohols. Med Times Gaz 2:703–706
5. Richet C (1893) On the relationship between the toxicity and the physical properties of substances. Compt Rendus Seances Soc Biol 9:775–776
6. Overton E (1897) Osmotic properties of cells in the bearing on toxicology and pharmacology. Z Phys Chem 22:189–209
7. Meyer H (1899) On the theory of alcohol narcosis. Arch Exp Pathol Pharmacol 42:109–118
8. Moore W (1917) Volatility of organic compounds as an index of the toxicity of their vapors to insects. J Agric Res 10(7):365
9. Hammett LP (1937) The effect of structure upon the reactions of organic compounds. Benzene derivatives. J Am Chem Soc 59(1): 96–103
10. Taft RW (1952) Polar and steric substituent constants for aliphatic and o-benzoate groups from rates of esterification and hydrolysis of esters1. J Am Chem Soc 74(12):3120–3128
11. Hansch C, Maloney PP, Fujita T et al (1962) Correlation of biological activity of phenoxyacetic acids with Hammett substituent constants and partition coefficients. Nature 194:178–180
12. Hansch C, Muir RM, Fujita T et al (1963) The correlation of biological activity of plant growth regulators and chloromycetin derivatives with Hammett constants and partition coefficients. J Am Chem Soc 85(18):2817–2824
13. Hansch C, Muir RM (1950) The ortho effect in plant growth-regulators. Plant Physiol 25(3):389
14. Hansch C, Fujita T (1964) p-σ-π analysis. A method for the correlation of biological activity and chemical structure. J Am Chem Soc 86(8): 1616–1626
15. Free SM Jr, Wilson JW (1964) A mathematical contribution to structure-activity studies. J Med Chem 7:395–399
16. Hansch C (1969) Quantitative approach to biochemical structure-activity relationships. Acc Chem Res 2(8):232–239

17. Nantasenamat C, Isarankura-Na-Ayudhya C, Naenna T et al (2009) A practical overview of quantitative structure-activity relationship. Excli J 8:74–88
18. Nantasenamat C, Isarankura-Na-Ayudhya C, Prachayasittikul V (2010) Advances in computational methods to predict the biological activity of compounds. Expert Opin Drug Discov 5(7):633–654
19. Medina-Franco JL, Martinez-Mayorga K, Bender A et al (2009) Characterization of activity landscapes using 2D and 3D similarity methods: consensus activity cliffs. J Chem Inf Model 49(2):477–491
20. Bajorath J (2012) Modeling of activity landscapes for drug discovery. Expert Opin Drug Discov 7(6):463–473
21. Doweyko AM (2008) QSAR: dead or alive? J Comput Aided Mol Des 22(2):81–89
22. Doweyko AM (2008) Is QSAR relevant to drug discovery? IDrugs 11(12):894–899
23. Tropsha A, Golbraikh A (2007) Predictive QSAR modeling workflow, model applicability domains, and virtual screening. Curr Pharm Des 13(34):3494–3504
24. Golbraikh A, Tropsha A (2002) Beware of q2! J Mol Graph Model 20(4):269–276
25. Huang J, Fan X (2011) Why QSAR fails: an empirical evaluation using conventional computational approach. Mol Pharm 8(2):600–608
26. Tropsha A, Gramatica P, Gombar VK (2003) The importance of being earnest: validation is the absolute essential for successful application and interpretation of QSPR models. QSAR Comb Sci 22(1):69–77
27. Tropsha A (2010) Best practices for QSAR model development, validation, and exploitation. Mol Inf 29(6–7):476–488
28. Fourches D, Muratov E, Tropsha A (2010) Trust, but verify: on the importance of chemical structure curation in cheminformatics and QSAR modeling research. J Chem Inf Model 50(7):1189–1204
29. Scior T, Bender A, Tresadern G et al (2012) Recognizing pitfalls in virtual screening: a critical review. J Chem Inf Model 52(4):867–881
30. Dearden JC, Cronin MT, Kaiser KL (2009) How not to develop a quantitative structure-activity or structure-property relationship (QSAR/QSPR). SAR QSAR Environ Res 20(3–4):241–266
31. Jewell NE, Turner DB, Willett P et al (2001) Automatic generation of alignments for 3D QSAR analyses. J Mol Graph Model 20(2): 111–121
32. Tervo AJ, Nyronen TH, Ronkko T et al (2004) Comparing the quality and predictiveness between 3D QSAR models obtained from manual and automated alignment. J Chem Inf Comput Sci 44(3):807–816
33. Olah M, Bologa C, Oprea TI (2004) An automated PLS search for biologically relevant QSAR descriptors. J Comput Aided Mol Des 18(7–9):437–449
34. Bhonsle JB, Wang Z-X, Tamamura H et al (2005) A simple, automated quasi-4D-QSAR, quasi-multi way PLS approach to develop highly predictive QSAR models for highly flexible CXCR4 inhibitor cyclic pentapeptide ligands using scripted common molecular modeling tools. QSAR Comb Sci 24(5): 620–630
35. Cartmell J, Enoch S, Krstajic D et al (2005) Automated QSPR through competitive workflow. J Comput Aided Mol Des 19(11): 821–833
36. Zhang S, Golbraikh A, Oloff S et al (2006) A novel automated lazy learning QSAR (ALL-QSAR) approach: method development, applications, and virtual screening of chemical databases using validated ALL-QSAR models. J Chem Inf Model 46(5):1984–1995
37. Bhonsle JB, Bhattacharjee AK, Gupta RK (2007) Novel semi-automated methodology for developing highly predictive QSAR models: application for development of QSAR models for insect repellent amides. J Mol Model 13(1):179–208
38. Obrezanova O, Csanyi G, Gola JM et al (2007) Gaussian processes: a method for automatic QSAR modeling of ADME properties. J Chem Inf Model 47(5):1847–1857
39. Rodgers SL, Davis AM, Tomkinson NP et al (2007) QSAR modeling using automatically updating correction libraries: application to a human plasma protein binding model. J Chem Inf Model 47(6):2401–2407
40. Ma CY, Buontempo FV, Wang XZ (2008) Inductive data mining: automatic generation of decision trees from data for QSAR modelling and process historical data analysis. Comput Aid Chem Eng 25:581–586
41. Wood DJ, Buttar D, Cumming JG et al (2011) Automated QSAR with a hierarchy of global and local models. Mol Inf 30(11–12):960–972
42. Perez-Castillo Y, Lazar C, Taminau J et al (2012) GA(M)E-QSAR: a novel, fully automatic genetic-algorithm-(meta)-ensembles approach for binary classification in ligand-based drug design. J Chem Inf Model 52(9):2366–2386
43. Cox R, Green DV, Luscombe CN et al (2013) QSAR workbench: automating QSAR modeling to drive compound design. J Comput Aided Mol Des 27(4):321–336

44. Martins JPA, Ferreira MMC (2013) QSAR modeling: a new open source computational package to generate and validate QSAR models. Quim Nova 26:554–560
45. Hall M, Frank E, Holmes G et al (2009) The WEKA data mining software: an update. SIGKDD Explorations 11 (1)
46. Venkateswarlu S, Ramachandra MS, Subbaraju GV (2005) Synthesis and biological evaluation of polyhydroxycurcuminoids. Bioorg Med Chem 13(23):6374–6380
47. Worachartcheewan A, Nantasenamat C, Isarankura-Na-Ayudhya C et al (2011) Predicting the free radical scavenging activity of curcumin derivatives. Chemometr Intell Lab Syst 109(2):207–216
48. Mandi P, Nantasenamat C, Srungboonmee K et al (2012) QSAR study of anti-prion activity of 2-aminothiazoles. Excli J 11:453–467
49. Nantasenamat C, Isarankura-Na-Ayudhya C, Naenna T et al (2008) Prediction of bond dissociation enthalpy of antioxidant phenols by support vector machine. J Mol Graph Model 27(2):188–196
50. Nantasenamat C, Li H, Mandi P et al (2013) Exploring the chemical space of aromatase inhibitors. Mol Div. doi:10.1007/s11030-11013-19462-x
51. Nantasenamat C, Piacham T, Tantimongcolwat T et al (2008) QSAR model of the quorum-quenching *N*-acyl-homoserine lactone lactonase activity. J Biol Syst 16(2):279–293
52. Pingaew R, Tongraung P, Worachartcheewan A et al (2012) Cytotoxicity and QSAR study of (thio) ureas derived from phenylalkylamines and pyridylalkylamines. Med Chem Res 22:4016-4029
53. Prachayasittikul S, Wongsawatkul O, Worachartcheewan A et al (2010) Elucidating the structure-activity relationships of the vasorelaxation and antioxidation properties of thionicotinic acid derivatives. Molecules 15(1):198–214
54. Thippakorn C, Suksrichavalit T, Nantasenamat C et al (2009) Modeling the LPS neutralization activity of anti-endotoxins. Molecules 14(5): 1869–1888
55. Worachartcheewan A, Nantasenamat C, Isarankura-Na-Ayudhya C et al (2013) Predicting antimicrobial activities of benzimidazole derivatives. Med Chem Res 22:5418–5430
56. Worachartcheewan A, Nantasenamat C, Naenna T et al (2009) Modeling the activity of furin inhibitors using artificial neural network. Eur J Med Chem 44(4):1664–1673
57. Nantasenamat C, Li H, Isarankura-Na-Ayudhya C et al (2012) Exploring the physicochemical properties of templates from molecular imprinting literature using interactive text mining approach. Chemometr Intell Lab Syst 116:128–136
58. Nantasenamat C, Isarankura-Na-Ayudhya C, Tansila N et al (2007) Prediction of GFP spectral properties using artificial neural network. J Comput Chem 28(7):1275–1289
59. Nantasenamat C, Naenna T, Isarankura N-AC et al (2005) Quantitative prediction of imprinting factor of molecularly imprinted polymers by artificial neural network. J Comput Aid Mol Des 19(7):509–524
60. Nantasenamat C, Isarankura-Na-Ayudhya C, Naenna T et al (2007) Quantitative structure-imprinting factor relationship of molecularly imprinted polymers. Biosens Bioelectron 22(12):3309–3317
61. Nantasenamat C, Srungboonmee K, Jamsak S et al (2013) Quantitative structure-property relationship study of spectral properties of green fluorescent protein with support vector machine. Chemometr Intell Lab Syst 120: 42–52
62. McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. Bull Math Biophys 5(4):115–133
63. Lawrence J (1993) Introduction to neural networks: design, theory, and applications, 6th edn. California Scientific Software, California
64. Smith M (1993) Neural networks for statistical modeling. Van Nostrand Reinhold, New York
65. Bishop CM, Nasrabadi NM (2006) Pattern recognition and machine learning, vol 1. Springer, New York
66. Vapnik V (2000) The nature of statistical learning theory. Springer, New York
67. Vapnik V (1998) Statistical learning theory. Wiley, New York
68. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
69. Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machines and other kernel-based learning methods. Cambridge University Press, Cambridge
70. Platt JC (1999) Fast training of support vector machines using sequential minimal optimization. In: Schoelkopf B, Burges C, Smola A (eds) Advances in kernel methods: support vector learning. MIT Press, Cambridge, USA, pp 185–208
71. Smola AJ, Schölkopf B (2004) A tutorial on support vector regression. Stat Comput 14(3): 199–222