Structure-Based Drug Screening and Ligand-Based Drug Screening with Machine Learning

Yoshifumi Fukunishi[*,1,2]

[1]*Biomedicinal Information Research Center (BIRC), National Institute of Advanced Industrial Science and Technology (AIST), 2-41-6, Aomi, Koto-ku, Tokyo 135-0064, Japan*

[2]*Pharmaceutical Innovation Value Chain, BioGrid Center Kansai, 1-4-2 Shinsenri-Higashimachi, Toyonaka, Osaka 560-0082, Japan*

Abstract: The initial stage of drug development is the hit (active) compound search from a pool of millions of compounds; for this process, *in silico* (virtual) screening has been successfully applied. One of the problems of *in silico* screening, however, is the low hit ratio in relation to the high computational cost and the long CPU time. This problem becomes serious in structure-based *in silico* screening. The major reason is the low accuracy of the estimation of protein-compound binding free energy. The problem of ligand-based *in silico* screening is that the conventional quantitative structure-activity relationship (QSAR) approach is not effective at predicting new hit compounds with new scaffolds. Recently, machine-learning approaches have been applied to *in silico* drug screening to overcome the above problems. We review here machine-learning approaches for both structure-based and ligand-based drug screening. Machine learning is used to improve database enrichment in two ways, namely by improving the docking score calculated by the protein-compound docking program and by calculating the optimal distance between the feature vectors of active and inactive compounds. Both approaches require compounds that are known to be active with respect to the target protein. In structure-based screening, the former approach is mainly used with a protein-compound affinity matrix. In ligand-based screening, both the former and latter approaches are used, and the latter approach can be applied to various kinds of descriptors, such as 1D/2D descriptors/fingerprints and the affinity fingerprint given by the protein-compound affinity matrix.

Keywords: Virtual screening, affinity fingerprint, machine learning, neural network model, support vector machine, decision tree, Bayesian model, self-organizing map.

1. INTRODUCTION

In silico (virtual) drug screening has played an important role in the initial stage of drug discovery. *In silico* screening and high throughput screening experiments have provided extensive protein-compound interaction data [1-11]. It is said that the probability of finding a hit (active) compound among the compounds in a random library is about 0.01% by a random screening experiment and about 1%-10% by *in silico* screening, if the *in silico* screening is suitably performed. The success of *in silico* screening has been achieved through progress in the theoretical development of protein-compound docking programs [12-23] and compound-similarity search programs, by the extension of a large-scale virtual compound library, and in other ways. We review here recent progress in machine learning applications for *in silico* drug screening.

The two major approaches of *in silico* drug screening are structure-based screening and ligand-based screening. Structure-based screening is based on the prediction of protein-compound binding free energy given by a protein-compound docking simulation for a target protein 3D structure. The ligand-based screening is a sort of chemical compound similarity search based on the known active compound. The compound information is translated into a vector of molecular descriptors. The definition of the distance between two vectors is introduced, and then the compounds which are close to the known active compounds are selected as candidate active compounds. CoMFA generates a pharmacophore by 3-dimensional alignment of several active compounds. Since the pharmacophore represents the active site of the target protein, CoMFA is similar to both structure-base screening and ligand-based screening [24]. The machine-learning approach is easily applied to the vectors of descriptors, and some of these methods are reviewed in this article.

Many docking programs have been developed [12-23] for structure-based screening, although the accuracy of the binding free energy estimation remains about 2-3 kcal/mol [16, 23]. A low accuracy of the binding free energy or docking score causes low database enrichment from *in silico* screening. One approach to improving database enrichment is to improve the docking score itself [25, 26]. But the limitation of this improvement is obvious. The free energy is a concept from statistical physics, and it is calculated on the basis of the partition function, which is based on a structural ensemble of numerous structures at a particular temperature; on the other hand, the docking score is calculated based on a single protein-compound complex structure. Instead of the modification of docking score, a combination of a docking score with 2-dimensional descriptors was developed [27]. In the docking process, the entropy of ligand is decreased, since some free rotations of bonds are fixed. The number of ro-

*Address correspondence to this author at the Biomedicinal Information Research Center (BIRC), National Institute of Advanced Industrial Science and Technology (AIST), 2-41-6, Aomi, Koto-ku, Tokyo 135-0064, Japan; Tel: +81-3-3599-8290; Fax: +81-3-3599-8099;
E-mail: y-fukunishi@aist.go.jp

1386-2073/09 $55.00+.00

tatable bonds of ligand is calculated from the topology of the ligand. Thus, it is reasonable to estimate the binding free energy by a linear combination of a docking score and 2-dimensional descriptors, which are calculated from the ligand's topology. This method is one type of QSAR methods, since the docking score is used as a ligand descriptor.

The other approach to improving database enrichment is the application of the protein-compound affinity matrix. The multiple-active-site correction (MASC) scoring method uses the deviation of the docking score instead of the raw docking score [28]. The multiple-target screening (MTS) method compares the docking scores of many proteins for one compound instead of comparing the docking scores of many compounds for one target protein [29, 30]. If some active compounds are known, the value of each element of the protein-compound affinity matrix is optimized to maximize the database enrichment of the MTS method by a machine-learning approach, denoted as machine-learning score modification MTS (MSM-MTS). We review here the MSM-MTS method as an application of machine learning to *in silico* drug screening.

Molecular descriptors and similarity measurement are key technologies for ligand-based screening. Many types of molecular descriptors are available, and many companies distribute them. The protein-compound affinity matrix (affinity fingerprint) is a new type of descriptor [31-39]. Each compound carries a feature vector calculated by the molecular descriptor. Machine learning is applied to the similarity measurement of the feature vector. The problems and advantages of machine learning were discussed in a previous review [40]. The most widely used methods are the Kohonen neural network (self-organizing map, SOM) [40-50], support vector machine (SVM) [51-58], Bayesian modeling [59-61], multi layer perceptron [62, 63], decision tree [64], and others. Some of these methods are briefly explained in this review.

The combination of structure-based screening and ligand-based screening is possible [65, 66]. The hit ratio achieved by structure-based screening is not high in many cases. A machine-learning approach is applied to extract a common feature of the active compounds predicted by the structure-based screening. Then a second, ligand-based screening is performed based on the common feature of the active compounds predicted by the first screening. This procedure is reviewed in the current paper.

2. APPLICATION OF MACHINE-LEARNING APPROACH TO STRUCTURE-BASED SCREENING

Kauvar *et al*. [67] proposed to approximate the IC_{50} value of a compound for a target protein by a linear combination of the IC_{50} values of the compounds for other proteins,

$$\log(IC_{50}(a,i)) = \sum_{b(b\neq a)} c_b^i \log(IC_{50}(b,i)), \quad (1)$$

where $IC_{50}(a,i)$ is the IC_{50} value of the *a*-th protein and the *i*-th compound, and c_b^i is a constant. This method worked well, but it requires a lot of experimental data. The lesson of this method is that the IC_{50} value could be approximated by the IC_{50} values of the compound *vs* other proteins.

Fukunishi *et al*. assumed that the binding free energy of the compound *vs* the target protein is improved by the linear combination of the binding free energies *vs* the target protein and other proteins [30],

$$s^{new\,i}_{\ a} = \sum_b s_b^i M_a^b, \quad (2)$$

where $s^{new\,i}_{\ a}$, $s_b^{\ i}$ and $M_a^{\ b}$ are the modified docking score of the *a*-th protein and the *i*-th compound, the raw docking score of the *b*-th protein and the *i*-th compound, and the constant coefficient, respectively. Since eq. 2 is an extension of eq. 1, we could expect eq. 2 to work well too. The problem is how to determine the coefficient $M_a^{\ b}$ without any experimental observation of the binding free energy.

The calculated score carries noise, and if the distribution of the scores satisfies the Gaussian distribution, Fukunishi *et al*. approximated eq. 2 as follows based on an analogy of the error theory [30].

$$s^{new\,i}_{\ a} = \frac{\sum_b s_b^i R_a^b}{\sum_b R_a^b}, \quad (3)$$

where $R_a^{\ b}$ is the correlation coefficient between the *a*-th and the *b*-th proteins

$$R_a^b = \frac{\sum_i (s_b^i - \frac{\sum_i s_b^i}{Nc})(s_a^i - \frac{\sum_i s_a^i}{Nc}) + \varepsilon}{\sqrt{\sum_i (s_b^i - \frac{\sum_i s_b^i}{Nc})^2 \cdot \sum_i (s_a^i - \frac{\sum_i s_a^i}{Nc})^2} + \varepsilon}. \quad (4)$$

Here, ε is a small number used to avoid the problem of division by zero when the correlation coefficient is zero, and *Nc* is the number of compounds. $R_a^{\ b}$ is a similarity measure between the *a*-th and the *b*-th proteins, and the similarity among the proteins could actually be calculated based on the protein-compound docking affinity matrix. This method is called direct score modification (DSM).

2.1. Machine-Learning Score Modification (MSM) Method

Eq. 3 is an analytical form of eq. 2. In contrast, an alternative interpretation of eq. 2 is possible; $s_b^{\ i}$ is an input vector for a two-layer perceptron, $M_a^{\ b}$ is a coefficient representing the network, and $s^{new\,i}_{\ a}$ is an output signal. From this point of view, $M_a^{\ b}$ could be determined by a machine-learning approach [30, 39]. The true value of $s^{new\,i}_{\ a}$ is unknown; another true signal is necessary for machine learning. One useful measure is the database enrichment achieved by $s^{new\,i}_{\ a}$. If known active compounds are available, we can determine $s^{new\,i}_{\ a}$ to maximize the database enrichment. To do so, a quantitative measure for database enrichment must be introduced. Let *x* and *f*(*x*) be the numbers of compounds (%) selected from the total compound library and from the database enrichment curve, respectively. The surface area under the database enrichment curve (*q*) is a measure of the database enrichment.

$$q = \int_0^{100} f(x)dx \quad (5)$$

Higher q values correspond to better database enrichment, and $0 < q < 100$. For the random screening, q=50.

The optimization procedure for $M_a{}^b$ is as follows.

Step 1. The initial matrix $\boldsymbol{M}$ in eq. 2 is estimated by eq. 3. The all-new docking scores calculated by eq. 8 are equal to the original docking scores. Then, screening by the combined MTS and MASC scoring method, which is the *in silico* screening method described in the next section, gives the q value by eq. 5.

Step 2. Many new matrices $\boldsymbol{M}$ are generated from a seed matrix $\boldsymbol{M}$ using random numbers. In the first step, the seed matrix $\boldsymbol{M}$ is the initial matrix $\boldsymbol{M}$, which is a unit matrix. The a-b element of the new matrix $\boldsymbol{M}$ ($M^{new}{}_a{}^b$) is given by $M^{new}{}_a{}^b = M_a{}^b + \eta_a{}^b$; here, $\eta_a{}^b$ is a random number and $-1 < \eta_a{}^b < 1$. Even if the number is digitized ($\eta_a{}^b$ ={-1, 0, 1}), this procedure works well.

Step 3. Using the newly generated matrix, the new docking score is calculated by eq. 2. Then screening by the combined MTS and MASC scoring method gives the q value by eq. 5. The best matrix $\boldsymbol{M}$, which gives the highest q value, is selected as the seed matrix for step 2.

Steps 2 and 3 are repeated until the q value shows convergence. This method is called the machine learning score modification (MSM) method.

2.2. *In Silico* Screening Method with the Combined MTS and MASC Scoring Method

We combined the multiple target screening (MTS) method [29, 30] and the multiple active site correction (MASC) scoring method [28] as an *in silico* screening method. The MTS and the MASC scoring methods can select different compounds; thus, the combination of the results obtained by these two methods is taken as the set of candidate hit compounds [29].

First, let us briefly explain the MTS method. We prepared a set of protein pockets $\boldsymbol{P}=\{p_1, p_2, p_3, \ldots p_M\}$, where p_a represents the a-th pocket. The total number of pockets is M. We also prepared a set of compounds $\boldsymbol{X}=\{x^1, x^2, \ldots x^N\}$, where x^i represents the i-th compound. The total number of compounds is N. For each pocket p_a, all compounds of set $\boldsymbol{X}$ are docked to pocket p_a with the score $s_a{}^i$ between the a-th pocket and the i-th compound. Here, $s_a{}^i$ corresponds to the binding free energy; a lower $s_a{}^i$ means a higher affinity between the a-th pocket and the i-th compound.

For the i-th compound, $\{s_a{}^i ; a=1,\ldots M\}$ were sorted in descending order, and the order $n_a{}^i$ was assigned to each a-th pocket depending on its value $s_a{}^i$. For example, when $n_a{}^i = 1$, the a-th pocket binds the i-th compound with the strongest affinity. When $n_a{}^i$ = M, the a-th pocket binds with the weakest affinity. This procedure was repeated until the order $\{n_a{}^i; a=1,...,M \mid i=1,...,N\}$ was determined for all compounds.

Next, we focused on the target a-th pocket. The compounds having the order $n_a{}^i = 1$ were assigned to be members of compound group 1, the compounds having $n_a{}^i = 2$ were assigned to be members of compound group 2, and so on. Among the group 1 members, the compound with the lowest $s_a{}^i$ should be the most probable hit compound. If there are no compounds in group 1, the compound with the lowest $s_a{}^i$ in group 2 should be the most probable hit compound. This procedure is repeated until the most probable hit compound is found.

Second, the MASC score can be explained as follows. The MASC score $s'_a{}^i$ for the a-th pocket and the i-th compound has been reported by Vigers and Rizzi as follows [28]:

$$s''_a{}^i = (s_a^i - \mu_i) / \sigma_i , \quad (6)$$

where $s_a{}^i$ is the raw docking score for the a-th pocket and the i-th compound, and μ_i and σ_i are the average and standard deviation of the raw docking scores across all pockets for the i-th compound, respectively. In this method, $s'_a{}^i$ is used for screening instead of $s_a{}^i$.

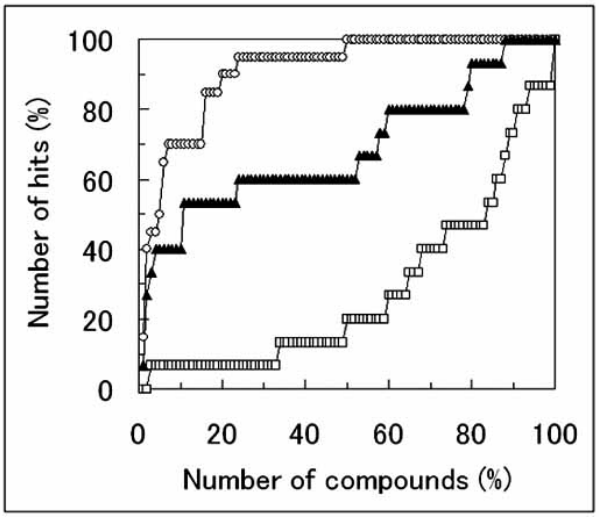
Both the MTS and the MASC scoring methods were applied in this study, and the combination of the results obtained by these two methods was taken as the set of candidate hit compounds.

2.3. MSM-MTS Method

The *in silico* screening method with a combined MTS and MASC scoring method based on the score given by the MSM method is called the MSM-MTS method [30]. A test calculation showed that about 40% of the active compounds were found within the first 1% of the database by the MSM-MTS method; this value was several times higher than the value obtained by the *in silico* screening method with a combined MTS and MASC scoring method without using the machine-learning approach. Considering that the MTS, MASC, and the combined MTS and MASC methods achieve a hit ratio that is several times higher than that of the conventional single target screening method, the advantage of the MSM-MTS method over the conventional method is obvious. The database enrichment depends on the number of proteins used in the protein-compound affinity matrix; the larger the number of proteins used, the greater the database enrichment. If a known active compound is unavailable, the *in silico* screening method with the combined MTS and MASC scoring method based on the score given by the DSM method (DSM-MTS) can be used to find about 22% of the active compounds within the first 1% of the database.

Results by structure-based *in silico* screening strongly depend on not only the target structure but also the particular protein-compound docking program that is used [68]. The results by the MTS method also depend on the target structure. Fig. (**1a**, **b**) show the database enrichment results of inhibitors of cyclo-oxygenase-2 (COX-2) by the MTS method with the raw docking score and that by the MSM-MTS method, respectively. The dataset and computational procedure are exactly the same as those reported in a previous paper [30]. McGovern *et al*. [69] reported that *holo* (protein-ligand complex) crystal structures would give better enrichments than *apo* (protein without ligand) crystal structures. This is not true in the present case; one *holo* structure gave better enrichment than the corresponding *apo* structure while another *holo* structure gave worse enrichment than the corresponding *apo* structure as shown in Fig. (**1a**). In general, the MSM-MTS method improved the results by the MTS method and is robust against structural changes of target protein as shown in Fig. (**1b**).

(**a**)



(**b**)

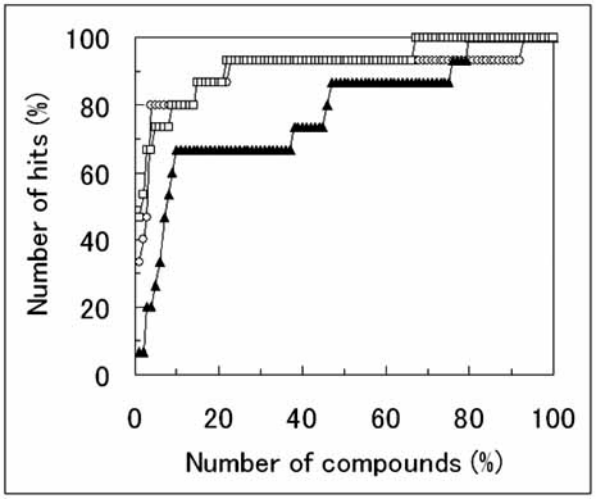


Fig. (1). Database enrichment curves of COX-2 inhibitors. Filled triangles, open circles, and open squares represent the results of 5cox (*apo* form), 4cox (*holo* form), and 6cox (*holo* form), respectively. Here, 5cox, 4cox and 6cox are the PDB ID codes. (**a**) Results by the MTS method with the raw docking scores. (**b**) Results by the MSM-MTS method.

2.4. Maximum Entropy Method and MTS Method

The docking score carries error. The maximum entropy method suggests that the most probable model can be created by changing the original data to maximize the informational entropy, given the level of experimental error [70]. We assumed that the calculated docking score of the *a*-th protein and the *i*-th compound $s^{calc}{}_a^i$ is the sum of the true docking score $s^{true}{}_a^i$ and the noise η_a^i,

$$s^{calci}{}_a = s^{truei}{}_a + \eta_a^i . \quad (7)$$

In other words, the new docking score can be defined as

$$s^{newi}{}_a = s_a^i + \eta_a^i , \quad (8)$$

where s_a^i is the original docking score. The informational entropy of the score is defined by

$$E = -\sum_{i,j} p_i^j \log_2 p_i^j \quad (9)$$

where $p_i^j = s_i^j$ or $p_i^j = (s_i^j - \overline{s^j})^2$. Here the score must be normalized to satisfy the relation $\sum_{i,j} p_i^j = 1$.

We applied this procedure to the data (the protein compound affinity matrix) used in our previous work [23]. The informational entropy *E* was maximized, allowing 1% error ($|\eta_a^i / s_a^i| < 0.01$), then the database enrichment was calculated by the MTS method. The database enrichment in the first 5% of the database was increased by 5%. The improvement is small, but this result shows that data mining based on the protein-compound affinity matrix is possible and that database enrichment can be improved without the improvement of the protein-compound docking software.

3. APPLICATION OF MACHINE-LEARNING APPROACH TO LIGAND-BASED SCREENING

For the ligand-based drug screening, usually a 1D and/or 2D descriptor of the compounds, such as the mass, or the number of rotatable bonds, or the number of hydrogen donors/acceptors, is used to evaluate the similarities between the compounds in the library and the known active compounds. Many methods have been proposed for the similarity searching of chemical compounds [71], such as the overlapping of chemical structure, the CATS descriptor method developed by Schneider *et al.* [72], the BCUT descriptor method [73], and others. One of the most popular methods is to use substructures of the compounds. A number of substructures or fragments (100-200 types) are provided by some companies, and each descriptor corresponds to the number of these substructures found in the compound.

In the CATS descriptor method, for each pair of pharmacophoric features (donor, acceptor, acid, base, *etc.*) in the molecule, the frequency of occurrence as a function of the number of bonds separating the features is accumulated in a pharmacophore pair vector. The bond distances from 1 to 10 are considered over all 15 feature combinations to give a vector size of 150. The Euclidian distance between two pharmacophore pair vectors is used as the similarity.

BCUT is one of the most widely used descriptor methods to evaluate the similarity of chemical compounds and the diversity of a given library. BCUT is one of topological index methods [74] and a set of several descriptors, which are eigenvalues of matrices. The diagonal parts of the matrices represent the atomic charge, polarizability, hydrogen donors and acceptors, and the off-diagonal parts of the matrices represent the structure of the compound. The protein-compound docking score (affinity fingerprint) has come to be used as the descriptor of the compound instead of a usual 1D or 2D descriptor [31-39].

The docking score index is a principal component of the affinity fingerprint [36-39]. Principal component analysis (PCA) was applied to the protein-compound interaction matrix to distinguish the active compounds of a target protein from the negative compounds. The active compounds were

localized in the PCA space of the compounds, which is a finite-dimension Hilbert space, while the negative compounds showed a wide distribution. In the PCA space, the compounds in a multi-dimensional sphere whose center was set to the average coordinates of known compounds were selected as a focused library, the database enrichment of which was equivalent to or better than that obtained by *in silico* screening.

All of these methods assign a vector of descriptors to each compound and evaluate the similarity between the known active compound and the test compound. The vector of descriptors can be easily manipulated in neural network theory, Bayes models, *etc*. Thus, there have been many reports on ligand-based screening supported by machine learning. In this section, some reports on this topic are reviewed.

3.1. Kohonen Neural Network

The self-organizing map (SOM) or Kohonen neural network is a sort of non-linear version of PCA in which high-dimensional feature space is projected into lower dimensional space; usually 2D space is adopted to visualize its output result [40-42]. Let $\boldsymbol{x_a}$ be an *n*-dimensional input feature vector such as $\boldsymbol{x_a}=\{x_1, x_2, \ldots x_n\}$. 2D SOM is composed of an *m* x *m* lattice with a neuron (u_{ij}) assigned to each lattice point (*i*,*j*). Each neuron u_{ij} carries a coefficient vector $\boldsymbol{w_{ij}} = \{w_{ij1}, w_{ij2}, \ldots. w_{ijn}\}$. When the feature $\boldsymbol{x_a}$ is input, the $\boldsymbol{w_{ij}}$ that is the most similar to $\boldsymbol{x_a}$ is selected, and it is updated to decrease the distance between $\boldsymbol{x_a}$ and the $\boldsymbol{w_{ij}}$. Also, $\boldsymbol{w_{kl}}$s, which are close to the $\boldsymbol{w_{ij}}$, are also updated to decrease the distance between $\boldsymbol{x_a}$ and $\boldsymbol{w_{kl}}$. These steps are iteratively performed until the value of $\boldsymbol{w}$ converges.

By using SOM, similar feature vectors are projected to the neurons that are close to each other on the map. Thus, we can find the candidate active compounds that are close to the known active compounds on the Kohonen map. The SOM overcomes the problems of over-fitting and over-training, which are serious problems in primitive neural network models. The problem in SOM is that the result depends on the initial value of $\boldsymbol{w}$. Since the initial value of $\boldsymbol{w}$ is given by a random number, the Kohonen map does not converge always to the same map when starting with the same test data set. The SOM has been used for the ligand-based screening and profiling/classification of chemical compounds.

3.2. Support Vector Machine (SVM)

The support vector machine (SVM) is one of the most widely used machine learning methods [50, 51]. Linear SVM generates a hyperplane separating two different classes (in this case, active and inactive compounds) of feature vectors ($\boldsymbol{x}$) with a maximum margin. Fig. (**2**) shows a schematic representation of SVM. The output of SVM is given by $f(\boldsymbol{x})=sign(g(\boldsymbol{x}))$ where $g(\boldsymbol{x})=\boldsymbol{w}^t\boldsymbol{x}+b$, $\boldsymbol{w}$ is a vector and *b* is a scalar. This hyperplane is constructed by finding values of $\boldsymbol{w}$ and *b* that maximize the $\boldsymbol{w}^t\boldsymbol{w}$ which satisfies the following conditions: $g(\boldsymbol{x}) > 1$ for active compounds and $g(\boldsymbol{x}) < -1$ for inactive compounds. In practical use, linear SVM is not so useful, since linear SVM can be applied only when the active and inactive compounds can be divided by a straight line (hyperplane) in the feature (compound descriptor) space, which is the input space, which is why nonlinear SVM is more useful.

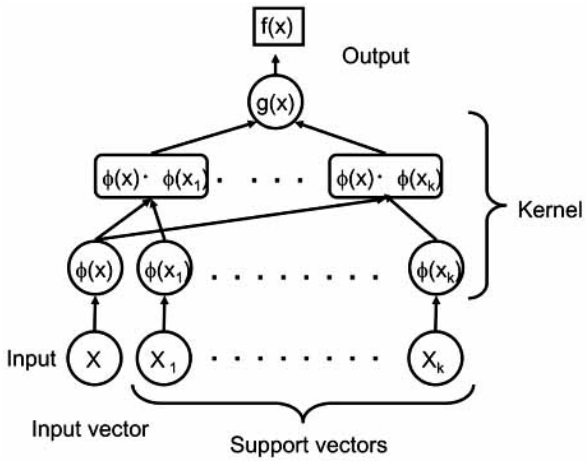


Fig. (2). Schematic representation of non-linear SVM. The input vector $\boldsymbol{x}$ and the support vectors ($\boldsymbol{x}_1,\ldots\boldsymbol{x}_k$) are transformed by the feature function $\boldsymbol{\phi}$ and mapped into the high-dimensional feature space. The dot product for feature functions $\boldsymbol{\phi}(\boldsymbol{x})\cdot\boldsymbol{\phi}(\boldsymbol{x}_i)$ is replaced with a kernel function $K(\boldsymbol{x}_i,\boldsymbol{x})$. Thus, the feature function $\boldsymbol{\phi}$ does not explicitly appear in eq. 10.

Nonlinear SVM projects feature vectors in the input space into another high-dimensional feature space by a nonlinear projection defined by a kernel function. The output of SVM is given by $f(\boldsymbol{x})=sign(g(\boldsymbol{x}))$ and

$$g(x) = \sum_{k=1}^{m} w_k K(x_k, x) + b \qquad (10)$$

where *K* is the so-called kernel function, the suffix *k* represents the support vector, and *m* stands for the number of support vectors.

A Gaussian kernel function such as $K(x_i,x_j)=exp(-(x_i-x_j)^2)/2\sigma^2)$ is popular, and in some cases, a polynomial kernel such as $K(x_i,x_j)=(1+x_ix_j)^p$ is adopted. The parameters σ and *p* must be given by the user *a priori*. Then, the linear SVM is applied to the projected feature vectors. After the determination of $\boldsymbol{w}$ and *b*, a given vector $\boldsymbol{x}$ can be classified by $f(\boldsymbol{x})=sign(g(\boldsymbol{x}))$, with positive or negative values indicating that the vector $\boldsymbol{x}$ belongs to the active or inactive compound, respectively.

An active learning method is used to enhance the effectiveness of SVM [52]. This method iteratively selects examples (active and inactive compounds) from a pool that improve $g(\boldsymbol{x})$. In the drug discovery cycle, candidate active compounds are selected by the SVM with an initial $g(\boldsymbol{x})$ determined by the known active compounds. Then, the true active and true negative compounds are determined by a wet experiment. This information is used to improve $g(\boldsymbol{x})$, and the next candidate compounds are selected by the improved $g(\boldsymbol{x})$. These steps are iteratively performed. Selecting the examples is problematic. The obvious selection strategy is to select the compounds with the largest positive scores ($g(\boldsymbol{x})$), since they are the most likely to be active. Another strategy is to select the compounds that are close to the hyperplane

(*i.e*. where $g(\boldsymbol{x})$ is nearly zero). Among the several strategies proposed, the former strategy has given the best performance.

3.3. Decision Tree (DT) Method

ID3 and C4.5 are the most popular DT algorithms. C4.5 DT is a sort of binary DT [64]. The decision rules are determined to maximize the gain of information. If the compound can be classified into 2 classes (*P* and *N*) based on its features, and the test data set consists of *p* compounds of class *P* and *n* compounds of class *N*, the information entropy of this classification is

$$I(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n} \quad (11)$$

If the test data set is divided into two subsets (S_1 and S_2) based on a feature *A*, and the subset S_i consists of p_i compounds belonging to *P* and n_i compounds belonging to *N*, the information entropy of this system is

$$E(A) = \frac{p_1+n_1}{p+n} I(p_1,n_1) + \frac{p_2+n_2}{p+n} I(p_2,n_2) \quad (12)$$

Thus, the information gain by this classification based on *A* is

$$Gain(A) = I(p,n) - E(A) \quad (13)$$

In DT, the feature *A*, which maximizes *Gain*(*A*), is selected. DT is a combination of the classifications. A greedy algorithm is adopted to select the features for the classification; the first classification gives the highest gain, the second classification gives the second highest gain, *etc*. In the latest version C5.0 DT, the test data set is divided into more than two subsets, but the basic concept is the same.

3.4. Machine-Learning Docking Score Index (ML-DSI) Method

In the framework of the DSI method, a measure to represent the distance between two compounds is determined based on the protein-compound affinity matrix. From the covariance matrix of the compounds, principal component analysis (PCA) is performed to find similar clusters of compounds. This DSI method was described in detail in previous papers [36, 39], and is briefly introduced below.

We prepare a set of pockets $\boldsymbol{P}=\{p_1, p_2, p_3, \ldots p_{Nr}\}$, where p_i represents the *i*-th pocket and *Nr* the total number of pockets, and a set of compounds $\boldsymbol{X}=\{x^1, x^2, \ldots x^{Nc}\}$, where x^k represents the *k*-th compound and *Nc* the total number of compounds. For each pocket p_i, all compounds of the set *X* are docked to the pocket p_i with a score of s_i^k between the *i*-th pocket and the *k*-th compound. Here, s_i^k corresponds to the binding free energy.

The covariance matrix $\boldsymbol{M^P}$ of the proteins is defined as

$$M^P{}_{ij} = \frac{1}{N_c}\sum_{k=1}^{N_c}(s_i^k - \overline{s_i})(s_j^k - \overline{s_j}), \text{ and} \quad (14)$$

$$\overline{s_i} = \frac{1}{N_c}\sum_{k}^{N_c} s_i^k , \quad (15)$$

where the upper bar represents an average. Let $\boldsymbol{\phi_j}$ be the *j*-th eigenvector of $\boldsymbol{M^P}$ with an eigenvalue ε_j, and let the order of ε_j be descendant. The vector of docking scores for the *k*-th compound $\boldsymbol{X_k}=(s_1^k, s_2^k, \ldots s_{Nr}^k)$ is represented by the linear combination of $\boldsymbol{\phi_j}$

$$X_k = \sum_{j=1}^{Nr} c_j^k \phi_j . \quad (16)$$

The coefficient $\{c_j^k\}$ represents the *j*-th coordinate of the PCA space of the *k*-th compound. In this study, we call this coefficient $\{c_j^k\}$ the "docking score index (DSI)".

Candidate hit compounds are selected using the following method. In the PCA space, compounds that are close to the known active compounds are selected as the candidate hit compounds. In the original version of the DSI method, the distance from the *k*-th compound to the average position of the active compounds (D_k) is defined as

$$D_k = \sqrt{\sum_{j=1}^{Nselect}(c_j^k - \overline{c_j})^2} , \quad (17)$$

and

$$\overline{c_j} = \sum c_j^{active} / N_a \quad (18)$$

where $c^{active}{}_j$ and N_a are the DSI values of the active compounds and the total number of active compounds.

The standard deviations (σ) of the DSI values were calculated for each axis, and DSI values more than 5 σdistant from the origin were removed from the analysis. We adopt a "standard Euclidian distance"; namely, the DSI values were scaled to set the standard deviation of the distribution of compounds of each axis to 1.

Since the selection of principal components is effective at distinguishing particular data from others, in this study, the suffix *j* runs over the selected axes $\{\alpha_1, \alpha_2, \ldots \alpha_{Nselect}\}$ in eq. 17 [37, 39], and the next modified distance D'_k is introduced.

$$D'_k = \sqrt{\sum_{j=\{\alpha 1,\alpha 2,\cdots \alpha Nselect\}}(c_j^k - \overline{c_j})^2} \quad (19)$$

The principal component axes are selected in the following manner. The contribution of each principal component is estimated using a database enrichment curve. The surface area under the database enrichment curve q_α is evaluated for the α-th principal component axis; namely, the suffix *j* in eq. 19 is set as α and N_{select} is set as 1, and the database enrichment curve f_α is calculated for the α-th axis. The q_α values are calculated by

$$q_\alpha = \int_0^{100} f_\alpha(x)dx , \quad (20)$$

where *x* and $f_\alpha(x)$ are the percentages of compounds that are selected from the total compound library and the database enrichment curve, respectively. A higher q_α value corresponds to better database enrichment, and the q_α value is always more than zero and less than 100. For the random screening, q_α=50.

The axes are sorted in descending order with respect to the q_α value. The *q* value given by eq. 5 is a measure of the database enrichment in addition to q_α in eq 20. The *q* value is calculated by changing the number of axes (N_{select}) used in

eq. 19 to find the optimal N_{select} value, which gives the maximum q value.

To apply the DSI method, the known active compounds are supposed to be available; hence, the docking score can be modified to increase the database enrichment. If the new docking score is given by the linear combination of the docking scores with many proteins as given by eq. 2, we can optimize the coefficients $M_a{}^b$ to maximize the q value as in the MSM-MTS method.

The optimization procedure for $M_a{}^b$ is as follows.

Step 1. The initial matrix $\boldsymbol{M}$ in eq. 2 is set as a unit matrix ($M_a{}^b = \delta_a{}^b$). The new docking scores are equal to the original docking scores. Then the DSI method gives the q value by eq. 5.

Step 2. Many new matrices $\boldsymbol{M}$ are generated from the seed matrix $\boldsymbol{M}$ using random numbers. In the first step, the seed matrix $\boldsymbol{M}$ is the initial matrix $\boldsymbol{M}$, which is a unit matrix. The *a*-*b* element of the new matrix $\boldsymbol{M}$ ($M^{new}{}_a{}^b$) is given by $M^{new}{}_a{}^b = M_a{}^b + \eta_a{}^b$; here, $\eta_a{}^b$ is a random number and $-1 < \eta_a{}^b < 1$.

Step 3. Using each newly generated matrix, the new docking score is calculated by eq. 6. Then the DSI method gives the q value by eq. 5. The best matrix $\boldsymbol{M}$ that gives the highest q value is selected as the seed matrix for step 2.

Steps 2 and 3 are repeated until the q value shows convergence; in this study, the number of cycles is set at 40. This method is called the machine-learning docking score index (ML-DSI) method. When $M_a{}^b$ in eq. 2 is set as a unit matrix, the method is called the factor-selection docking score index (FS-DSI) method. Importantly, in the FS-DSI method, the important principal component axes are selected without machine learning.

A test calculation showed that about 70% of the active compounds were found within the first 1% of the database by the ML-DSI method and this value was several times higher than that found using the *in silico* screening method with the original DSI method without machine learning. The database enrichment depends on the number of proteins used in the protein-compound affinity matrix, just as in the MSM-MTS method; the larger the number of proteins is, the higher the database enrichment.

Fig. (**3**) shows the screening result of inhibitors of macrophage migration inhibitory factor (MIF) by the ML-DSI method. The data sets and computational procedure were the same as those in a previous paper [39]. Compounds are depicted in the ranking order by the ML-DSI method. The known active compounds are compounds (**2**), (**3**), (**4**), (**5**), (**6**), (**7**), and (**8**), which are reported in reference [1] and are registered in the protein data bank. The newly predicted active compounds are compounds (**1**), (**9**), (**10**), (**11**), and (**12**). Screening by the ML-DSI method was successfully performed. Namely, all the known active compounds were found in the top 113 compounds selected out of a total of 11,212 compounds (1.0 %), and all the new active compounds were found in the top 430 compounds selected out of the total 11212 compounds (3.8 %). MIF has three binding pockets, but MIF binds to more than three molecules of compound (**1**). Thus, compound (**1**) is not a selective compound and must be omitted from the following discussion.

Many of the known active compounds are similar to each other. Namely, compounds (**2**), (**3**), (**7**), and (**8**) have a coumarin-like scaffold, and compound (**6**) is similar to coumarin. Compound (**4**) is similar to these known compounds. On the contrary, the newly predicted compounds do not have a coumarin-like scaffold. Finding new active compounds with a new scaffold is more important than finding new active compounds with the same scaffold. This is an example of successful scaffold hopping (lead hopping or chemical hopping), and such scaffold hopping is a key advantage of using the protein compound affinity matrix (affinity fingerprint).

3.5. SVM-DSI Method

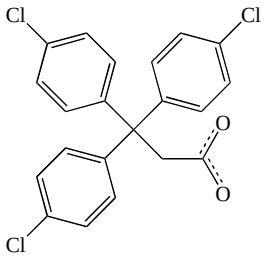
Li *et al.* compared the performances of SVM, k-nearest neighbor (k-NN), probabilistic neural network (PNN) and C4.5 decision tree (DT), and the authors found that SVM showed the best prediction performance among them [54].

In *k*-NN, the distance between an unclassified vector $\boldsymbol{x}$ and each individual vector $\boldsymbol{x_i}$ in the training set is measured [75]. The class of the majority of the *k* nearest neighbors is chosen as the predicted class of $\boldsymbol{x}$. Thus, the original DSI and ML-DSI methods represent some of the simplest versions of *k*-NN.

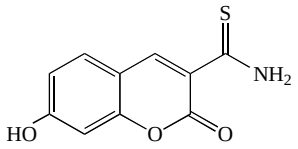
We applied SVM to the DSI method. At first, the ML-DSI method was applied. Each compound carries the vector of modified DSI. The averaged position (***Pa***) of the known active compounds was calculated and the averaged distance (*Ra*) from ***Pa*** to the active compounds was also calculated. The distance (*D*) is the distance between each compound and ***Pa***. The compounds that satisfied $1.5\ Ra < D < 3\ Ra$ were selected as inactive compounds. SVM with a Gaussian kernel was applied by using these active and inactive compounds. The q value of eq. 5 was calculated by changing the σ value of the Gaussian kernel from 40 to 70 to find the optimal σ_{opt}, which gave the highest q value. For each target, the σ_{opt} value was calculated automatically. Finally, SVM with the σ_{opt} was applied after the ML-DSI, where the compounds were sorted in descending order of their $g(\boldsymbol{x})$ values instead of the *D'* values determined using eq. 19. Thus, the *b* value of eq. 10 is not necessarily in this method. We call this procedure the SVM-DSI method.

We applied the SVM-DSI method to the same data used in our previous work [39]. Fig. (**4**) shows the database enrichment results obtained by the ML-DSI and SVM-DSI methods. The protein sets used for the protein-compound affinity matrix were the protein sets A, B, C, D and E reported in the previous work [39], and these sets consist of 180, 123, 93, 63, and 24 proteins, respectively. The results obtained by the ML-DSI method were slightly better than those obtained by the SVM-DSI method in every case however Fig. (**4**) shows only the results for the protein sets A and E. Even if the selection rule was changed, and the compounds that satisfied $2\ Ra < D < 4\ Ra$ were selected as inactive compounds, this trend did not change.

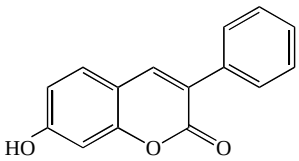
This result showed that SVM is not the best method in some cases. In some cases, another method can give a better result than that obtained by SVM. We must choose the most suitable method for the given feature vector.



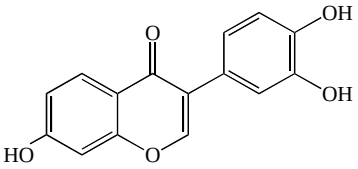
1



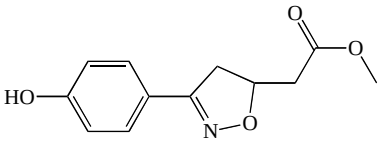
2



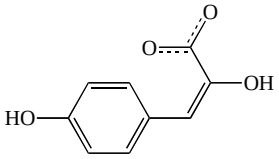
3



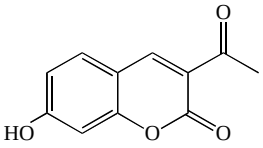
4



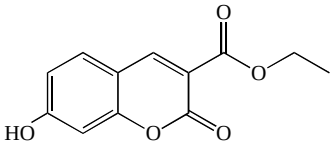
5



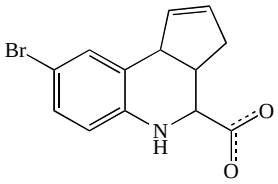
6



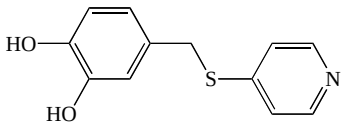
7



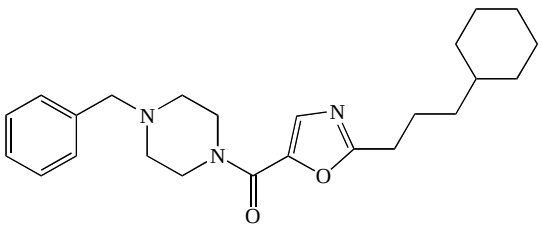
8



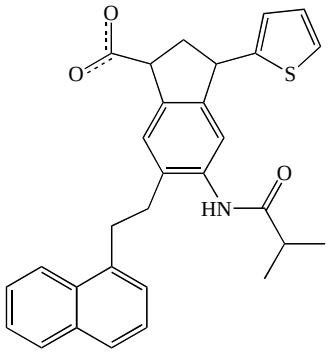
9



10



11



12

Fig. (3). MIF inhibitors.

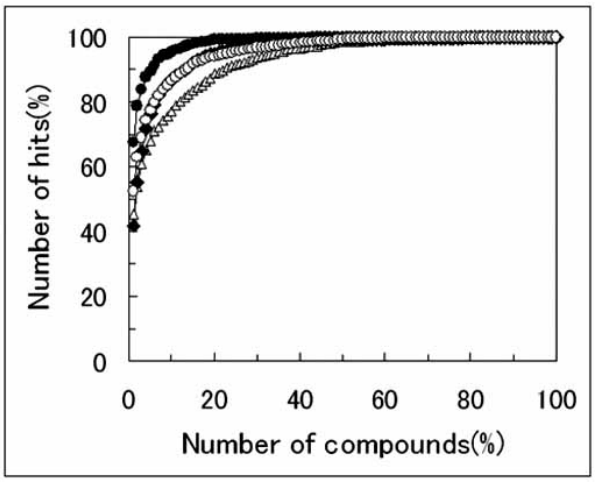


Fig. (4). Averaged database enrichment curves of 14 target proteins using an affinity matrix of 180 proteins (protein set A) and 24 proteins (protein set E) [38]. Filled circles, open circles, filled diamonds and open triangles represent the averaged database enrichments obtained by the ML-DSI method with protein set A, the SVM-DSI method with protein set A, the ML-DSI method with protein set E and the SVM-DSI method with protein set E, respectively.

4. APPLICATION OF MACHINE-LEARNING APPROACH TO COMBINATION OF STRUCTURE-BASED SCREENING AND LIGAND-BASED SCREENING

Klon *et al.* proposed a combination of structure-based screening and ligand-based screening [65, 66]. The target protein was HIV-1 protease, the active compounds were 424 small chemical compounds and 175 peptide inhibitors, and 179805 compounds were used as the candidate inactive compounds. At first, conventional structure-based screening was performed by a protein-compound docking program, such as Glide, FlexX [13] or GOLD [14]. These three programs succeeded in finding both the small chemical active compounds and the peptide inhibitors. The top-ranked compounds found by the structure-based screening were designated as candidate active compounds, while all other compounds were designated as candidate inactive compounds. Then fingerprints were calculated for all compounds in the database. A Bayesian model was trained using the fingerprints from the candidate active and inactive compounds. Finally, all of the compounds were re-ranked according to the Bayesian model.

A Bayesian model is a statistical model based on the Bayes rule of conditional probability [59-61]

$$P(active \mid feature) = P(feature \mid active)\frac{P(active)}{P(feature)}. \qquad (21)$$

$P(active|feature)$ is the probability that the compound will be active when the compound has the feature. $P(feature|active)$ is the probability that the compound will have the feature when the compound is active. $P(active)$ is the probability that a given compound in the database will be active, $P(feature)$ is the probability that a given feature will occur in the database. Usually, the feature is a vector of many descriptors. If the descriptors are independent (naïve Bayesian model), $P(active|feature)$ is given by

$$P(active \mid feature) = P(feature_1 \mid active)P(feature_2 \mid active) \cdots P(feature_n \mid active)\frac{P(active)}{P(feature)} \qquad (22)$$

where $feature_1$, $feature_2$, … $feature_n$ are the descriptors of the feature vector.

If a test set of active and inactive compounds is given and the features of these compounds are calculated, $P(active|feature)$ can be calculated. Once $P(active|feature)$ is given, an unclassified compound can be easily classified by calculating its feature.

With several hundred features incorporated into the Bayesian model, this procedure drastically improved the database enrichment. When Glide was used, the percentages of active compounds found in the top 2% of the database were 82-93% by the conventional structure-based screening and 99% by this procedure. When FlexX was used, these percentages were determined to be 29-38% by the conventional structure-based screening and 70-77% by this procedure [13]. When GOLD was used, these percentages were found to be 23-87% by the conventional structure-based screening and 96-100% by this procedure [14]. While the enrichment depends on the type of docking program used and the type of active compounds (small chemical compounds or peptide inhibitors) this combination procedure worked well in all cases.

This procedure could be applied to other screening methods, such as the MSM-MTS, SVM, and ML-DSI methods.

Another naïve combination of structure-based screening and ligand-based screening is the simple union of the compounds selected by the structure-based and ligand-based screenings. The candidate active compounds found by structure-based screening are rich in variety and the hit ratio is low, while those found by ligand-based screening are poor in variety with a high hit ratio. Thus, we took the union of those predicted compounds. We applied the MSM-MTS and ML-DSI methods to TNF-alpha converting enzyme (TACE) (Tsujishita, H. personal communication 2007). For TACE, six protein-compound complex structures are reported in PDB the ligands of these complex structures were adopted as the active compounds. Previously, a random screening and conventional structure-based screenings by DOCK [12] and Glide were performed. Seven active compounds were found by the random screening of 76,000 compounds. In a preliminary screening test with the known active compounds, no known active compound was predicted by DOCK. Six hundred sixty candidate active compounds were predicted by Glide among a library of 400,000 compounds, but there was no true active compound. About 900 candidate active compounds were selected by the combination method of MSM-MTS and ML-DSI among 1,000,000 compounds compiled by myPresto (http://presto.protein.osaka-u.ac.jp/myPresto/index_e.html) [23, 30, 39, 76], and 38 true active compounds were found for a hit ratio of 4.2%. Compared to the random screening, the enrichment factor obtained by the combination method was about 500. The naïve combination method is simple, but it can be useful.

5. APPLICATION OF DSI/ML-DSI METHODS TO COMBINATORIAL CHEMISTRY

Fujita *et al*. applied the FS-DSI method to drug screening for vasopressin 1b (V1b) receptor (Orita M.; Kanai C.; Fujita S. personal communication 2007). For V1b, only one active compound (Sanofi: SSR-149415) has been reported recently. V1b is a G-protein coupled receptor whose 3D structure is unknown thus the 3D model was constructed based on the 3D structure of bovine rhodopsin (PDB code: 1F88) by homology modeling. Conventional structure-based screening was performed using DOCK [12] and 33 active compounds were found among the 15,000 compounds selected by DOCK. Six active compounds (compounds (**1b**), (**2b**), (**3b**), (**4b**), (**5b**) and (**6b**) out of these 33 compounds (IC_{50} < 20 μM) were selected as scaffolds for the following combinatorial synthesis. The numbers of synthesized compounds for these six scaffolds and the number of active compounds are summarized in Table **1**. These synthesized compound libraries were evaluated by the DSI method. The distributions of the synthesized libraries based on compounds (**5b**) and (**6b**) were close to SSR-149415 in the PCA compound space, and 2 and 13 active compounds (IC_{50} < 20 μM) were newly found based on compounds (**5b**) and (**6b**), respectively. In contrast, those based on compounds (**1**b), (**2b**), (**3b**) and (**4b**) were far from the known active compounds in the PCA compound space, and zero active compounds were found among these libraries. Three hundred twenty-six compounds were generated from compound (**5b**) by further combinatorial synthesis, but there was no active compound. Furthermore, a virtual library of 756 compounds was generated from compound (**6b**) by combinatorial synthesis, and then 84 compounds that were close to the known active compounds were selected by the FS-DSI method. Among these 84 compounds, three new active compounds were found.

Fig. (**5**) shows the PCA compound space calculated by a kind of FS-DSI method. The active compounds are localized in the PCA space. The compounds generated by the combinatorial chemistry based on compound (**6b**) are also localized and are close to the known active compounds. This study shows that the sub-library generated by combinatorial chemistry could be evaluated by the DSI/FS-DSI methods before the actual experiment. The virtual combinatorial library can be generated by the VCOL program of myPresto suite [23, 76].

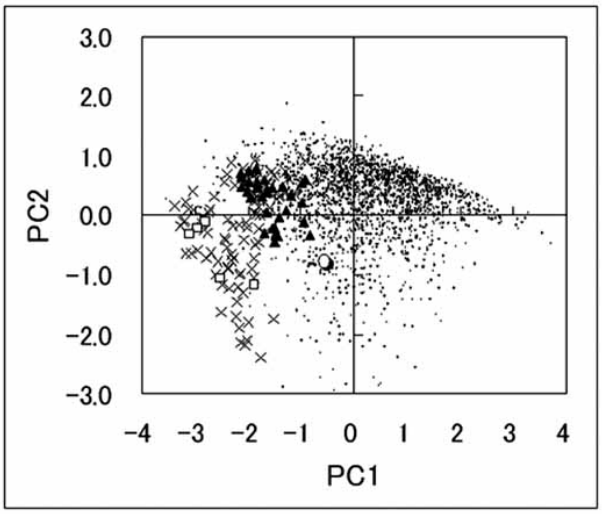


Fig. (5). PCA results of active and inactive compounds for V1b receptor. The open circle, filled triangles, open squares, crosses and dots represent SSR-149415, active compounds found by DOCK, active compounds found by a combinatorial synthesis, inactive compounds found by a combinatorial synthesis and inactive compounds found by a random screening, respectively. The compounds generated by the combinatorial synthesis are based on compound **6** and only the active compounds with IC_{50} < 10 μM are depicted.

6. CONCLUSION

The machine-learning approach has been used in both structure-based drug screening and ligand-based drug screening.

In structure-based drug screening, the MSM-MTS method was reviewed. The MTS method is based on the protein-compound affinity matrix, whose element is a docking score calculated by protein-compound docking software. The MTS method is a sort of target profiling of compounds. When the target of a compound is equal to the target protein in question, the compound is selected as a candidate hit compound of the target protein. In the MTS method, the new docking score is given by a linear combination of other docking scores, and the machine-learning approach determines the optimal coefficients of the linear combination, which gives the maximum database enrichment. The

Table 1. The Results Obtained by Combinatorial Synthesis

Compound (Scaffold)	First Combinatorial Synthesis			Second Combinatorial Synthesis		
	No of Synthesized Compounds	DSI	No of Hits	No of Synthesized Compounds	DSI	No of Hits
1b	117	far[a]	0	-	-	-
2b	175	far[a]	0	-	-	-
3b	168	far[a]	0	-	-	-
4b	175	far[a]	0	-	-	-
5b	360	close[b]	2	326	-	0
6b	99	close[b]	13	84	close[b]	3

a: The distribution of the synthesized compounds is far from the distribution of the known active compounds in the PCA compound space when determined by the FS-DSI method.
b: The distribution of the synthesized compounds is close to the distribution of the known active compounds in the PCA compound space when determined by the FS-DSI method.

meaning of the linear combination can be explained by error theory.

Machine learning is mainly applied to ligand-based drug screening and it is applied to the calculation of the optimal distance between the feature vectors of active and inactive compounds. One of the applications is score modification just as in the MSM-MTS method. The other popular methods are SVM and Bayesian modeling. The advantages of these methods depend on the type of feature vector, and their effectiveness is evaluated in the test calculation for practical use. Thus we cannot say which method is the best among the available methods, *a priori*.

The combination of structure-based screening and ligand-based screening should be useful. The training set for machine-learning ligand-based screening is composed of the candidate active compounds predicted by the structure-based screening. Then, the machine-learning ligand-based screening selects the final candidate active compounds out of the compound library. Usually, the machine-learning approach requires known active compounds. On the contrary, in this procedure, known active compounds are not necessary.

The machine-learning approach can be applied to various stages of structure-based screening and ligand-based screening, and many theoretical techniques should be combined to increase the database enrichment.

ACKNOWLEDGEMENTS

This work was supported by grants from the New Energy and Industrial Technology Development Organization of Japan (NEDO) and the Ministry of Economy, Trade, and Industry (METI) of Japan.

REFERENCES

[1] Orita, M.; Yamamoto, S.; Katayama, N.; Aoki, M.; Takayama, K.; Yamagiwa, Y.; Seki, N.; Suzuki, H.; Kurihara, H.; Sakashita, H.; Takeuchi, M.; Fujita, S.; Yamada, T.; Tanaka, A. *J. Med. Chem.*, **2001**, *44*, 540.

[2] Cotesta, S.; Giordanetto, F.; Trosset, J.-Y.; Crivori, P.; Kroemer, R.T.; Stouten, P.F.W.; Vulpetti, A. *Proteins*, **2005**, *60*, 629.

[3] Schellhammer, I.; Rarey, M. *Proteins*, **2004**, *57*, 504.

[4] Evers, A.; Hessler, G.; Matter, H.; Klabunde, T. *J. Med. Chem.*, **2005**, *48*, 5448.

[5] Howard, M.H.; Cenizal, T.; Gutteridge, S.; Hanna, W.S.; Tao, Y.; Totrov, M.; Wittenbach, V.A.; Zheng, Y.-J. *J. Med. Chem.*, **2004**, *47*, 6669.

[6] Godden, J.W.; Stahura, F.L.; Bajorath, J. *J. Med. Chem.*, **2004**, *47*, 5608.

[7] Zhao, L.; Brinton, R.D. *J. Med. Chem.*, **2005**, *48*, 3463.

[8] Mestres, J.; Veeneman, G.H. *J. Med. Chem.*, **2003**, *46*, 3441.

[9] Shacham, S.; Marantz, Y.; Bar-Haim, S.; Kalid, O.; Warshaviak, D.; Avisar, N.; Inbal, B.; Heifetz, A.; Fichman, M.; Topf, M.; Naor, Z.; Noiman, S.; Becker, O.M. *Proteins*, **2004**, *57*, 51.

[10] Cavasotto, C.N.; Orry, A.J.W.; Abagyan, R.A. *Proteins*, **2003**, *51*, 423.

[11] Katada, S.; Hirokawa, T.; Oka, Y.; Suwa, M.; Touhara, K. *J. Neurosci.*, **2005**, *25*, 1806.

[12] Kuntz, I.D.; Blaney, J.M.; Oatley, S.J.; Langridge, R.; Ferrin, T.E. *J. Mol. Biol.*, **1982**, *161*, 269.

[13] Rarey, M.; Kramer, B.; Lengauer, T.; Klebe, G. *J. Mol. Biol.*, **1996**, *261*, 470.

[14] Jones, G.; Willet, P.; Glen, R.C.; Leach, A.R.; Taylor, R. *J. Mol. Biol.*, **1997**, *267*, 727.

[15] Paul, N.; Rognan, D. *Proteins*, **2002**, *47*, 521.

[16] Baxter, C.A.; Murray, C.W.; Clark, D.E.; Westhead, D.R.; Eldridge, M.D. *Proteins*, **1998**, *33*, 367.

[17] McGann, M.R.; Almond, H.R.; Nicholls, A.; Grant, J.A.; Brown, F.K. *Biopolymers*, **2003**, *68*, 76.

[18] Goodsell, D.S.; Olson, A.J. *Proteins*, **1990**, *8*, 195.

[19] Taylor, J.S.; Burnett, R.M. *Proteins*, **2000**, *41*, 173.

[20] Abagyan, R.; Totrov, M.; Kuznetsov, D. *J. Comput. Chem.*, **1994**, *15*, 488.

[21] Colman, P.M. *Curr. Opin. Struct. Biol.*, **1994**, *4*, 868.

[22] Kramer, A.; Kirchhoff, P.D.; Jiang, X.; Venkatachalam, C.M.; Waldman, M. *J. Mol. Graph. Model.*, **2005**, *23*, 395.

[23] Fukunishi, Y.; Mikami, Y.; Nakamura, H. *J. Mol. Graph. Model.*, **2005**, *24*, 34.

[24] Cramer, R.R.III.; Patterson, D.E.; Bunce, J.D. *J. Am. Chem. Soc.*, **1988**, *110*, 5959.

[25] Zhang, C.; Liu, S.; Zhu, Q.; Zhou, Y. *J. Med. Chem.*, **2005**, *48*, 2325.

[26] Muegge, I.; Martin, Y.C. *J. Med. Chem.*, **1999**, *42*, 791.

[27] Hetenyi, C.; Paragi, G.; Maran, U.; Timar, Z.; Karelson, M.; Penke, B. *J. Am. Chem. Soc.*, **2006**,*128*, 1233.

[28] Vigers, G.P.A.; Rizzi, J.P. *J. Med. Chem.*, **2004**, *47*, 80.

[29] Fukunishi, Y.; Mikami, Y.; Kubota, S.; Nakamura, H. *J. Mol. Graph. Model.*, **2005**, *25*, 61.

[30] Fukunishi, Y.; Kubota, S.; Nakamura, H. *J. Chem. Inf. Comput. Sci.*, **2006**, *46*, 2071.

[31] Briem, H.; Kuntz, I.D. *J. Med. Chem.*, **1996**, *39*, 3401.

[32] Lessel, U.F.; Briem, H. *J. Chem. Inf. Comput. Sci.*, **2000**, *40*, 246.

[33] Briem, H.; Lessel, U.F. *Persp. Drug Discov. Des.*, **2000**, *20*, 231.

[34] Weber, A.; Teckentrup, A.; Briem, H. *J. Comput. Aided Mol. Des.*, **2002**, *16*, 903.

[35] Hsu, N.; Cai, D.; Damodaran, K.; Gomez, R.F.; Keck, J.G.; Laborde, E.; Lum, R.T.; Macke, T.J.; Martin, G.; Schow, S.R.; Simon, R.J.; Villar, H.O.; Wick, M.M.; Beroza, P. *J. Med. Chem.*, **2004**, *47*, 4875.

[36] Fukunishi, Y.; Mikami, Y.; Takedomi, K.; Yamanouchi, M.; Shima, H.; Nakamura, H. *J. Med. Chem.*, **2006**, *49*, 523.

[37] Fukunishi, Y.; Kubota, S.; Kanai, C.; Nakamura, H. *J. Comput. Aided Mol. Des.*, **2006**, *20*, 237.

[38] Fukunishi, Y.; Kubota, S.; Nakamura, H. *J. Mol. Graph. Model.*, **2007**, *25*, 633.

[39] Fukunishi, Y.; Hojo, S., Nakamura, H. *J. Chem. Inf. Comput. Sci.*, **2006**, *46*, 2610.

[40] Manallack, D.T.; Livingstone, D.J. *Eur. J. Med. Chem.*, **1999**, *34*, 195.

[41] Kohonen, T. *Self-Organizing Maps*; Springer: New York, **1995**.

[42] Burkard, U. In *Chemoinfomatics-A Textbook-*; Gasteiger, J.; Engel, T. Eds.; Wiley-VCH: Weinheim, **2005**, pp. 435-481.

[43] Livingstone, D. In *Neural networks in QSAR and drug design*; Devillers, J. Ed.; Academic press: London, **1996**, pp. 157-176.

[44] Simon, V.; Gasteiger, J.; Zupan, J. *J. Am. Chem. Soc.*, **1993**, *115*, 9148.

[45] Anzali, S.; Barnickel, G.; Krug, M.; Sadowski, J.; Wagener, M.; Gasteiger, J.; Polanski, J. *J. Comput. Aided Mol. Des.*, **1996**, *10*, 521.

[46] Anzali, S.; Barnickel, G.; Krug, M.; Sadowski, J.; Wagener, M.; Gasteiger, J. In *Neural networks in QSAR and drug design*; Devillers, J. Ed.; Academic Press: London, **1996**, pp. 209-222.

[47] Polanski, J.; Gasteiger, J.; Wagener, M.; Sadowski, J. *J. Quant. Struct. Act. Relat.*, **1998**, *17*, 27.

[48] Bauknecht, H.; Zell, A.; Bayer, H.; Levi, P.; Wagener, M.; Sadowski, J.; Gasteiger, J. *J. Chem. Inf. Comput. Sci.*, **1996**, *36*, 1205.

[49] Anzali, S.; Mederski, W.K.R.; Osswald, M.; Dorsch, D. *Bioorg. Med. Chem. Lett.*, **1998**, *8*, 11.

[50] Bienfait, B. *J. Chem. Inf. Comput. Sci.*, **1994**, *34*, 890.

[51] Vapnik, V.N. *The Nature of Statistical Learning Theory*; Springer: New York, **1995**.

[52] Ivanciuc, O. In *Reviews in Computational Chemistry*, Lipkowitz, K.B.; Cundari., T.R. Eds.; Wiley-VCH: Weinheim, **2007**, Vol. *23*, pp. 291-400.

[53] Warmuth, M.K.; Liao, J.; Ratsch, G.; Mathieson, M.; Putta, S.; Lemmen, C. *J. Chem. Inf. Comput. Sci.*, **2003**, *43*, 667.

[54] Jorissen, R.N.; Gilson, M.K. *J. Chem. Inf. Comput. Sci.*, **2005**, *45*, 549.

[55] Li, H.; Ung, C.Y.; Yap, C.W.; Xue, Y.; Li, Z.R.; Chen, Y.Z. *J. Mol. Graph. Model.*, **2006**, *25*, 313.

[56] Byvatov, E.; Fechner, U.; Sadowski, J.; Schneider, G. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1882.

[57] Xue, Y.; Yap, C.W.; Sun, L.Z.; Cao, Z.W.; Wang, J.F.; Chen, Y.Z. *J. Chem. Inf. Comput. Sci*., **2004**, *44*, 1497.
[58] Doniger, S.; Hofman, T.; Yeh, J. *J. Comput. Biol*., **2002**, *9*, 849.
[59] Klon, A.E.; Glick, M.; Davies, J.W. *J. Med. Chem*., **2004**, *47*, 4356.
[60] Jacobsson, M.; Liden, P.; Stjernschantz, E.; Bostrom, H.; Norinder, U. *J. Med. Chem*., **2003**, *46*, 5781.
[61] Xia, X.Y.; Maliski, E.G.; Gallant, P.; Rogers, D. *J. Med. Chem*., **2004**, *47*, 4463.
[62] Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. *Nature*, **1986**, *333*, 533.
[63] Zupan, J.; Gasteiger, J. *Neural Networks in Chemistry and Drug Design*; Wiley-VCH: Weinheim, **1999**.
[64] Quinlan, J.R. *C4.5: Programs for Machine Learning*; Morgan Kaufmann: San Mateo, **1993**.
[65] Klon, A.E.; Glick, M.; Davies, J.W. *J. Chem. Inf. Comput. Sci*., **2004**, *44*, 2216.
[66] Klon, A.E.; Glick, M.; Thoma, M.; Acklin, P.; Davies, J.W. *J. Med. Chem*., **2004**, *47*, 2743.
[67] Kauvar, L. M.; Higgins, D. L.; Villar, H. O.; Sportsman, J. R.; Engqvist-Goldstein, A.; Bukar, R.; Bauer, K. E.; Dilley, H.; Rocke, D.M. *Chem. Biol*., **1995**, *2*, 107.
[68] Warren, G.L.; Andrews, C.W.; Capelli, A.M.; Clarke, B.; LaLonde, J.; Lambert, M.H.; Lindvall, M.; Nevins, N.; Semus, S.F.; Senger, S.; Tedesco, G.; Wall, I.D.; Woolven, J.M.; Peishoff, C.E.; Head, M.S. *J. Med. Chem*., **2006**, *49*, 5912.
[69] McGovern, S.L.; Shoichet, B.K. *J. Med. Chem*., **2003**, *46*, 2895
[70] Fukunishi, Y.; Kasai, T.; Kuwata, K. *Chem. Phys*., **1993**, *177*, 85.
[71] Ghose, A. K.; Viswanadhan, V. N. *Combinatorial Library Design and Evaluation – Principle, Software Tools, and Applications in Drug Discovery*; Marcel Dekker: New York, **2001**, pp. 337-362.
[72] Pickett, S. In *Protein-Ligand Interactions from Molecular Recognition to Drug Design – Methods and Principles in Medicinal Chemistry*; Boehm, H. J.; Schneider, G.; Mannhold, R.; Kubinyi, H.; Folkers, G. Eds.; Wiley-VCH: Weinheim. **2003**, pp. 88-91.
[73] Pearlman, R.S.; Smith, K.M. *J. Chem. Inf. Compt. Sci*., **1999**, *39*, 28.
[74] Terfloth, L. In *Chemoinfomatics-A Textbook-*; Gasteiger, J.; Engel, T. Eds.; Wiley-VCH: Weinheim, **2005**, pp. 397-433.
[75] Hert, J.; Willett, P.; Wilton, D.J.; Acklin, P.; Azzaoui, K.; Jacoby, E.; Schuffenhauer, A. *J. Chem. Inf. Comput. Sci*., **2006**, *46*, 462.
[76] Fukunishi, Y.; Mikami, Y.; Nakamura, H. *J. Phys. Chem. B*, **2003**, *107*, 13201.

Received: May 9, 2008 **Revised: July 2, 2008** **Accepted: August 29, 2008**