Artificial Intelligence in the Life Sciences 2 (2022) 100046

Contents lists available at ScienceDirect

Artificial Intelligence in the Life Sciences

journal homepage: www.elsevier.com/locate/ailsci

Research Article

Symbolic regression for the interpretation of quantitative structure-property relationships

Katsushi Takaki[a], Tomoyuki Miyao[a,b,*]

[a] Graduate School of Science and Technology, Nara Institute of Science and Technology, 8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan
[b] Data Science Center, Nara Institute of Science and Technology, 8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan

ARTICLE INFO

Keywords:
Model interpretability
Quantitative structure-activity relationships
Quantitative structure-property relationships
Symbolic regression
Genetic programming

ABSTRACT

The interpretation of quantitative structure–activity or structure–property relationships is important in the field of chemoinformatics. Although multivariate linear regression models are typically interpretable, they do not generally have high predictive abilities. Symbolic regression (SR) combined with genetic programming (GP) is a well-established technique for generating the mathematical expressions that describe the relationships within a dataset. However, SR sometimes produces complicated expressions that are hard for humans to interpret. This paper proposes a method for generating simpler expressions by incorporating three filters into GP-based SR. The filters are further combined with nonlinear least-squares optimization to give filter-introduced GP (FIGP), which improves the predictive ability of SR models while retaining simple expressions. As a proof-of-concept, the quantitative estimate of drug-likeness and the synthetic accessibility score are predicted based on the chemical structures of compounds. Overall, FIGP generates less-complicated expressions than previous SR methods. In terms of predictive ability, FIGP is better than GP, but is outperformed by a support vector machine with a radial basis function kernel. Furthermore, quantitative structure–activity relationship models are constructed for three matching molecular series with biological targets. In the case of one target, the activity prediction models given by FIGP exhibit better predictive ability than multivariate linear regression and support vector regression with the radial basis function kernel, whereas for the remaining cases, FIGP is slightly less accurate than multivariate linear regression.

1. Introduction

The interpretation of quantitative structure–property or structure–activity relationships (QSPR/QSAR) is an important topic in the field of chemoinformatics [1]. Classical QSAR models are interpretable when multivariate linear regression (MLR) is employed in combination with meaningful molecular descriptors [2,3]. MLR has been widely employed for a range of QSPR/QSAR applications, such as determining the relation between enantio-selectivity and chemical reaction parameters [4–6]. However, as a modeling method, MLR has a poor predictive ability when the relationship between the molecular descriptors and the property (activity) is nonlinear. Thus, in practical applications, nonlinear machine learning (ML) algorithms such as random forests (RF) [7], support vector machines with a nonlinear kernel function (SVM) [8], and neural networks (NNs) are frequently employed. These nonlinear ML models accurately predict the property values, even when the structure–property relationship is linear, by adjusting the model parameters.

In terms of interpreting nonlinear ML models, approaches are generally based on individual compounds. That is, the prediction output of a ML model for a compound can be decomposed into additive molecular descriptor contributions. Because this method is quite effective for understanding the relation between the model output and an input descriptor set, i.e., local interpretation, it is widely employed for various target types [9–12]. However, such a local interpretation does not always provide an understanding of the predictive model itself (i.e., QSAR/QSPR). Thus, modeling approaches that are interpretable to humans and are more flexible than MLR are necessary.

Symbolic regression (SR) searches for the mathematical expressions that explain a training dataset. Roughly speaking, an SR expression consists of a combination of arithmetic operators or mathematical functions and terminals (variables and numerical constants). SR expressions do not rely on a fixed functional form, unlike the engineered features of MLR (multiplication/division). Thus, SR has the potential to represent nonlinear QSARs/QSPRs as explicit expressions without any prior knowledge regarding the functional form of the expression. Because the search space of expressions is generally vast, and expressions can be nat-

* Corresponding author at: Data Science Center and Graduate School of Science and Technology, Nara Institute of Science and Technology, 8916-5 Takayama-cho, Ikoma, Nara 630-0192, Japan.
E-mail address: miyao@dsc.naist.jp (T. Miyao).

https://doi.org/10.1016/j.ailsci.2022.100046
Received 6 September 2022; Received in revised form 1 November 2022; Accepted 2 November 2022
Available online 5 November 2022

urally represented as tree structures, genetic programming (GP) is used to solve SR problems [13].

In materials science, GP-based SR and its variants have recently been employed to derive mathematical expressions of physical phenomena [14]. An SR system named AI Feynman successfully recovered 100 equations of physical laws from datasets [15]. AI Feynman produces sets of expressions through a trade-off between the accuracy and simplicity of the expressions. GP-based SR has also been used to engineer highly correlated features for a target variable [16,17]. In terms of QSAR analysis, only a limited number of SR studies have been reported, and these have mainly focused on the predictive ability of the models [18,19].

To ensure interpretable QSAR/QSPR models, the expressions generated by SR should be as simple as possible, but flexible enough to represent nonlinear structure–property (activity) relationships. For the purpose of improving the fit to a dataset, Kommenda et al. [20] proposed to optimize numerical constants during the evolution of the SR expressions, resulting in a great improvement in the predictive ability of the generated expressions, even for test datasets. This technique is called GP with nonlinear least-squares (NLS) optimization of constant terms. However, the expressions generated by this method sometimes contain complicated functional relations and many constants, most likely from overfitting to a training dataset.

In this paper, we consider the generation of interpretable QSAR/QSPR models. For this purpose, we introduce three filters into GP with NLS optimization to give filter-introduced GP (FIGP). The introduced filters are a function filter (F-filter), variable filter (V-filter), and domain filter (D-filter). These filters constrain the generated expressions to be simple and valid for compounds outside the domain of the training dataset [21]. As a proof-of-concept, two well-defined properties are employed: the quantitative estimate of drug-likeness (QED) [22] and the synthetic accessibility score (SAScore) [23]. The predictive performance of the proposed FIGP model is compared with that of two other SR models, namely GP and AI Feynman. The effects of the filters are analyzed by monitoring the expressions generated during the evolution process. The FIGP model gives simpler expressions and more stable predictive performance than GP without the filters. Furthermore, three QSAR models for substituents of the chemical structures of active compounds are built using FIGP as demonstrative case studies. Our implementation of FIGP is publicly available in the GitHub repository at https://github.com/takakikatsushi/FIGP.

2. Materials and methods

2.1. Compound dataset and molecular representations

From the ZINC15 database [24], a total of 11,670,964 substances from 718 tranches were downloaded as SMILES strings using the following options: representation: 2D, reactivity: clean, purchasability: in-stock. After standardizing the chemical structures in the files, e.g., removing salts and converting to neutralized forms of (de)protonated substructures of chemical structures, 1,000,000 compounds were randomly sampled. This compound pool was used for our virtual experiments. All the molecular descriptors used in this study were implemented in RDKit [25]. These descriptors were manually chosen with the aim of directly connecting to the interpretation of chemical structures. Thus, topological descriptors and descriptors based on the sum of atom-wise surface areas with property contributions were excluded. Furthermore, descriptors counting the functional groups were omitted to prevent the interpretation from being too specific to the functional groups. From this descriptor set, those descriptors having the same value for more than 90% of the molecules were removed. Further variable selection was conducted so that any pair of descriptors had a correlation coefficient less than or equal to 0.9. In this variable selection, variables exhibiting correlation coefficients greater than 0.9 more than once were iteratively removed. The remaining 23 descriptors used for benchmark calculations in this study along with their average values and ranges are listed in Table 1.

2.2. Formulas

The QED [22] and SAScore [23] were employed for the formula estimation in this study. These metrics were chosen because they can be analytically derived from a chemical structure and have been widely used in previous retrospective in-silico studies.

2.2.1. QED

The QED is the geometric mean of eight desirability functions. Each function represents a desirable property of a compound in terms of a single molecular descriptor, and the combination of these functions quantifies the drug-likeness of the compound. These descriptors are the molecular mass (M_r), octanol–water partition coefficient (ALOGP), numbers of hydrogen bond donors (HBDs) and acceptors (HBAs), molecular polar surface area (PSA), number of rotatable bonds (ROTB), number of aromatic rings (AROM), and number of structural alerts (ALERTS). Each desirability function d is an asymmetric double-sigmoidal function with six parameters. These parameters were determined by fitting the function to the density (histograms) of the descriptor values from a collection of 771 orally dosed approved drugs. For each descriptor, a high desirability score is assigned to molecules with descriptor values around the mode of the distribution. Because each desirability function is scaled by the maximum desirability function score, the QED scores range from 0 (undesirable) to 1 (desirable). The outputs of the eight desirability functions are weighted as follows to derive the weighted QED score.

$$\mathrm{QED_w} = D_{M_r} \cdot D_{\mathrm{ALOGP}} \cdot D_{\mathrm{HBD}} \cdot D_{\mathrm{HBA}} \cdot D_{\mathrm{PSA}} \cdot D_{\mathrm{ROTB}} \cdot D_{\mathrm{AROM}} \cdot D_{\mathrm{ALERTS}}, \tag{1}$$

where

$$D_i = \exp\left(\frac{w_i \ln d_i}{\mathrm{W}}\right),$$

$$\mathrm{W} = w_{M_r} + w_{\mathrm{ALOGP}} + w_{\mathrm{HBD}} + w_{\mathrm{HBA}} + w_{\mathrm{PSA}} + w_{\mathrm{ROTB}} + w_{\mathrm{AROM}} + w_{\mathrm{ALERTS}}$$

and w_i is the weight for the i-th desirability function (one of the eight descriptors). A high weighted QED score can only be achieved when a molecule gives high scores for all desirability functions. Note that $\mathrm{QED_w}$ takes a value of 0 if any one of the d_i is equal to 0. Three weighting schemes have been proposed based on the information content of $\mathrm{QED_w}$. In this study, $\mathrm{QED_{w,mo}}$ is used, which takes the average of the top 1000 weight combinations that give the highest information content. These weights are $w_{\mathrm{Mr}} = 0.66$, $w_{\mathrm{ALOGP}} = 0.46$, $w_{\mathrm{HBD}} = 0.61$, $w_{\mathrm{HBA}} = 0.05$, $w_{\mathrm{PSA}} = 0.06$, $w_{\mathrm{ROTB}} = 0.65$, $w_{\mathrm{AROM}} = 0.48$, and $w_{\mathrm{ALERTS}} = 0.95$. The QED scores were calculated by the *QED.default* function implemented in the RDKit library [25].

2.2.2. SAScore

The SAScore represents the difficulty of synthesis based on the chemical structure of a compound, from 1 (easy to synthesize) to 10 (difficult to synthesize) [23]. The SAScore consists of two factors: the appearance of rare substructures (FragmentScore) and the complexity of molecular structures (ComplexityPenalty).

$$SAScore_{raw} = -FragmentScore + ComplexityPenalty \tag{2}$$

In FragmentScore, frequently appearing molecular fragments contribute to positive values, while rare fragments produce negative scores. These fragment frequencies were determined from 1,000,000 molecules in the PubChem database [26]. The ComplexityPenalty term is further decomposed into four equally weighted penalty terms:

$$RingComplexityScore = \log_{10}(nRingBridgeAtoms + 1) + \log_{10}(nSpiroAtoms + 1), \tag{3}$$

Table 1
Statistics of descriptor values for 1,000,000 ZINC compounds.

Descriptor	Definition	Mean (std)	Range [min, max]
EMWt	Exact molecular weight	381.1 (92.7)	[58, 998.3]
FCSP3	Fraction of C atoms that are SP3 hybridized	0.4 (0.2)	[0, 1]
MaxAbsESI	Maximum absolute value of the E-state indicator	12.2 (2)	[1.5, 18.7]
MaxAbsPC	Maximum absolute value of partial charge	0.4 (0.1)	[0, 0.8]
MaxPC	Maximum value of partial charge	0.3 (0.1)	[-0.3, 0.8]
MinAbsESI	Minimum absolute value of E-state indicator	0.1 (0.1)	[0, 6]
MinESI	Minimum value of E-state indicator	-1.1 (1.5)	[-8.5, 2]
NHOHCount	Number of NH and OH	1.1 (1)	[0, 26]
NOCount	Number of N and O	5.8 (1.9)	[0, 35]
NAliCc	Number of aliphatic carbon rings	0.2 (0.6)	[0, 13]
NAliHc	Number of aliphatic heterocycles	0.7 (0.8)	[0, 22]
NAliR	Number of aliphatic rings	1 (0.9)	[0, 22]
NAroCc	Number of aromatic carbon rings	1.4 (0.9)	[0, 29]
NAroHc	Number of aromatic heterocyclic rings	0.9 (0.9)	[0, 7]
NAroR	Number of aromatic rings	2.3 (1.1)	[0, 29]
NHA	Number of hydrogen bond acceptors	4.7 (1.8)	[0, 30]
NHetAtm	Number of heteroatoms	6.9 (2.2)	[0, 43]
NRB	Number of rotatable bonds	5.3 (2.4)	[0, 53]
NSCc	Number of saturated carbon rings	0.2 (0.5)	[0, 13]
NSHc	Number of saturated heterocycles	0.5 (0.7)	[0, 22]
NSR	Number of saturated rings	0.7 (0.8)	[0, 22]
RCount	Number of rings	3.3 (1.1)	[0, 34]
logp	Octanol–water partition coefficient	3.3 (1.6)	[-13.1, 19]

$$StereoComplexityScore = \log_{10}(nStereoCenters + 1), \quad (4)$$

$$MacrocyclePenalty = \log_{10}(nMacrocycles + 1), \quad (5)$$

$$SizePenalty = natoms^{1.005} - natoms, \quad (6)$$

where a macrocycle is defined as having more than eight atoms in a ring. Therefore, large compounds consisting of many complicated fragments represented by the structural features above produce high values of $SAScore_{raw}$ in Eq. (2). This raw score is scaled from 1–10 to produce the final SAScore. In the RDKit implementation for SAScore, *sascorer.calculateScore*, the original $SAScore_{raw}$ definition was slightly modified to treat macrocyclic structures and the symmetry of molecules.

2.3. Symbolic regression with genetic programming

The goal of SR is to learn the mathematical expressions underlying a regression model between an objective variable y and independent variables **x** from a training dataset. Unlike black-box ML models, a mathematical expression describes the regression model and the structure of the model is not fixed before training. Because the solution space of expressions is vast, efficient search algorithms are necessary. GP [13] mimics the natural evolution process to search for the optimal solution, and has been successfully applied to searches across the solution space of expressions. In GP, an expression is represented as a tree, where leaves are numerical constants or variables and (non-leaf) nodes are mathematical operators applied to their child node(s) (Fig. 1**A**). This structure makes it possible to apply evolution-mimicking operations in GP, namely mutation and crossover. In the mutation operation, a selected subtree is replaced by another randomly generated subtree with a certain probability. In the crossover operation, two individuals (expressions) are randomly selected, and one (randomly selected) subtree from each expression is swapped with one from the other expression (Fig. 1**B**).

To guide the evolution of the expressions in a desirable direction, a fitness function is used. The fitness function determines whether newly created individuals (expressions) survive into the next generation. In general, individuals with higher fitness values survive. As a fitness metric, the root mean square error (RMSE), the coefficient of determination (R^2), or the mean absolute error (MAE) between the observed and predicted y values is usually employed [13, 20].

2.3.1. Constant optimization in GP

One of the issues in GP with SR is the treatment of numerical parameters or constants in expression trees. For example, in Fig. 1**A**, the constants a and b appear in the expression $y=ax+b$. In a naïve approach, these parameters can be absorbed in the structure of GP by giving them only a limited number of choices, such as 0, 1, and π. In more refined ways, these parameters can be either sampled from a probability distribution, e.g., a uniform distribution (Fig. 1**C** left), or numerically optimized as a parameter set during the GP operation (Fig. 1**C** right). A previous methodological comparison showed that the NLS estimation of the constant terms outperforms other GP variants in addition to several ML models [20]. Thus, our proposed method is based on GP with NLS optimization in addition to introducing three filters, as explained below. In this manuscript, GP with NLS optimization of the constant terms is referred to as GP for simplicity.

2.4. Filter-introduced GP

2.4.1. Three filters

One motivation for using SR as the modeling method is to understand natural phenomena or experimental results in the form of mathematical expressions. Therefore, simple and understandable mathematical expressions must be constructed. The expressions generated by GP without any constraints sometimes contain undesirable features, such as one variable appearing in several terms (1), nested operations (2), and invalid operations (3). These three situations are illustrated in Fig. 2. Fig. 2**A** depicts situation (1) with an expression tree containing three variables. Variable x_2 appears twice in different terms of the expression, resulting in a model that is difficult to interpret. Fig. 2**B** represents situation (2) with an expression tree containing two consecutive "exp" operations. Situation (3) only occurs when applying the expression to another dataset (Fig. 2**C**). Expressions that are well-fitted to the training dataset sometimes produce infinite values for unseen test compounds because of an invalid operation. This occurs when test data points reside outside the domain of applicability of the expression [27]. For example, when HBD is selected as the descriptor, and all molecules in the training dataset have at least one HBD, this descriptor may become the denominator of an expression. Subsequently, if molecules without HBDs are encountered, this expression will result in division by zero and be invalid.

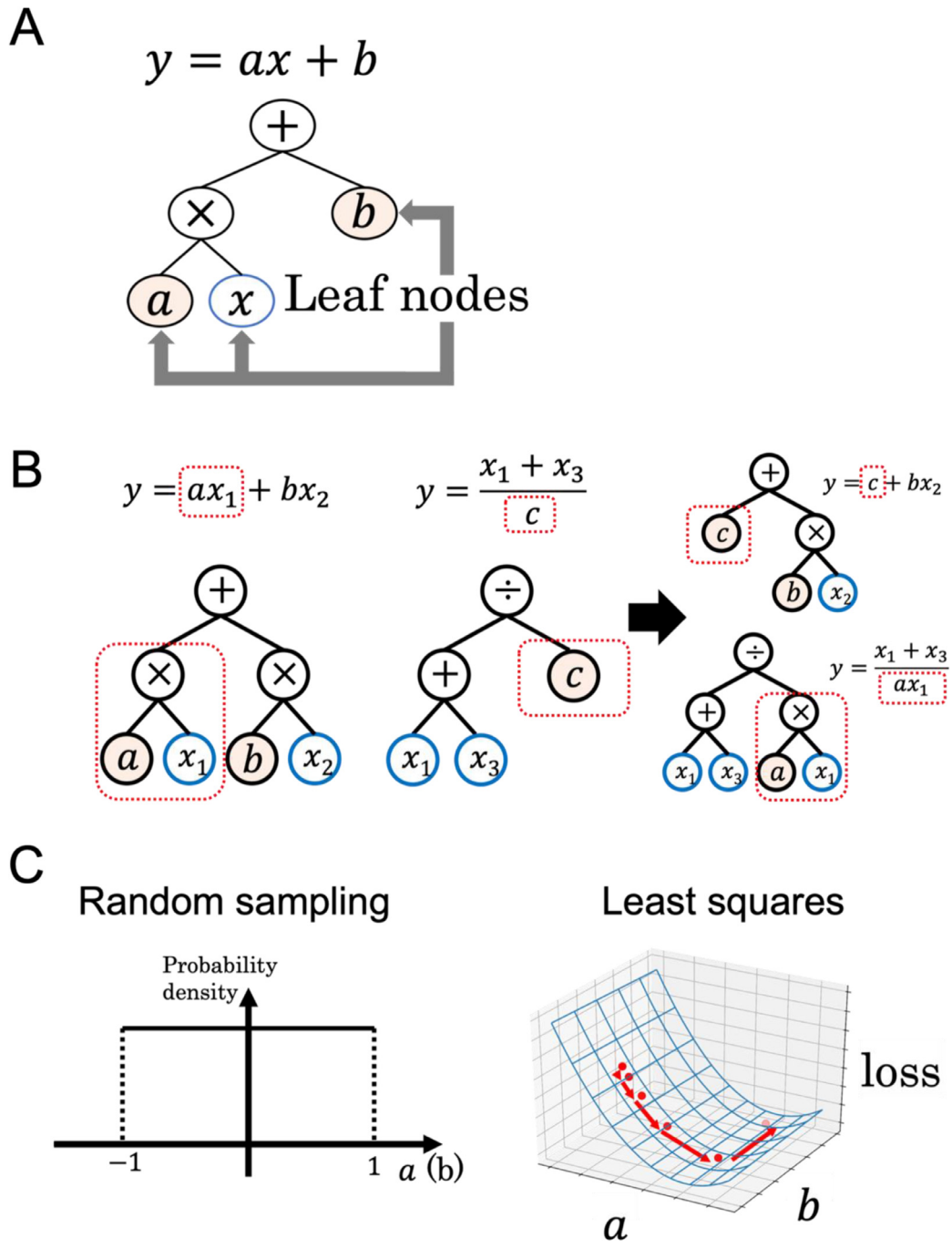

Fig. 1. Concept of symbolic regression (SR) with genetic programming (GP). A: Linear regression model with a graph representation in GP, where a and b are constants to be optimized. **B:** Crossover operation of two expressions, where red dotted-squares are selected as target subtrees. **C:** Optimization of constants a and b. Initial values for these two constants are randomly sampled (left), followed by nonlinear least-squares optimization (right).

To solve situation (1), expressions containing the same variables in different leaves are simply discarded. This is the V-filter. For situation (2), expressions with specific operators such as “exp” or “log” that appear more than once in a subtree are removed by the F-filter. In this study, the F-filter detects the hierarchical usage of {exp, ln} and {sqrt, square, cube}. For situation (3), the detection of potentially harmful expressions during the training phase of GP is introduced through the D-filter. In the D-filter, several test data points outside the domain of the training data are used to determine whether the output values exceed the range of the objective variable or not. In this study, all data points in the training and test datasets passed through this filter. By applying these three filters during expression evolution, surviving expressions are expected to be easy for humans to understand.

2.4.2. *Limitations of FIGP*

One of the biggest limitations of the proposed FIGP is the possibility of not being able to find the optimal expression for a phenomenon, such as when the ground-truth formula for the phenomenon violates one or more of the to-be-avoided situations explained in Fig. 2. The precise expression of a mathematical formula usually requires the same variable to appear in different terms (violation of situation (1)). However, by ensuring that a variable appears at most once, the relation between the objective and independent variables becomes clearer. Note that simple

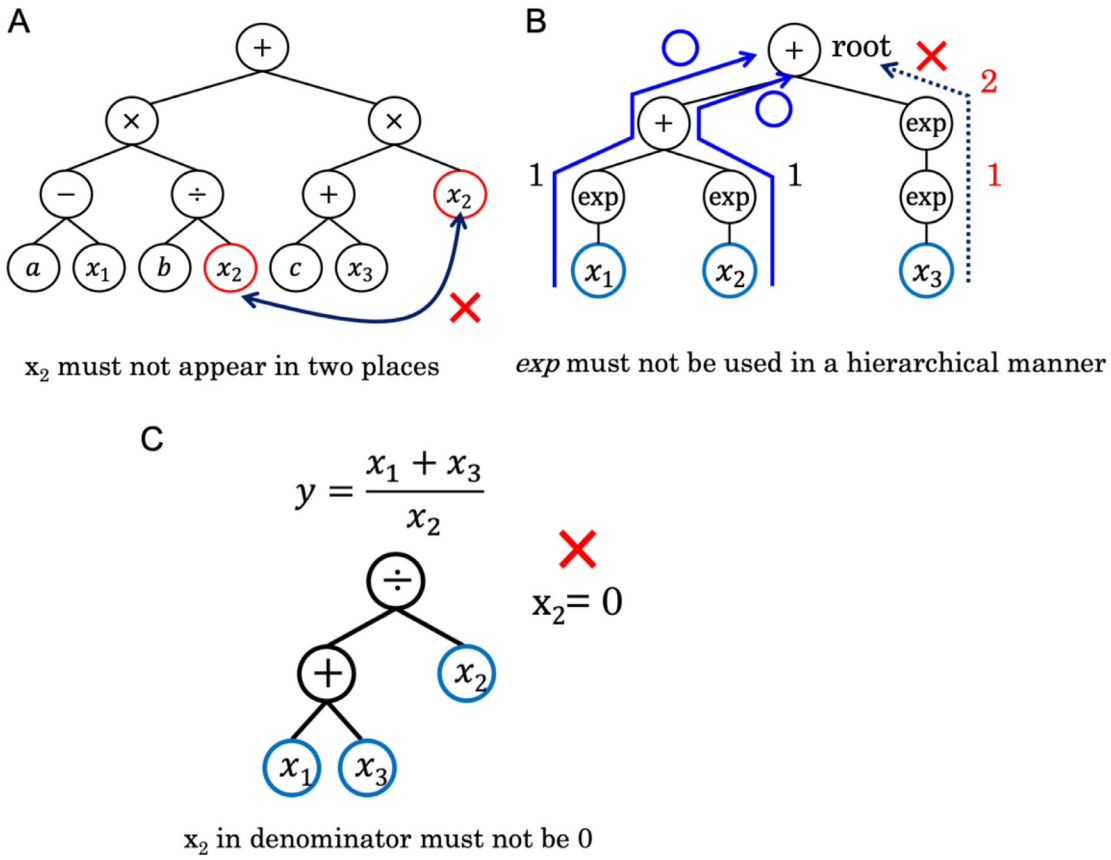

Fig. 2. Three cases where interpretation of the expression becomes difficult. A: The same variable appears more than once in a GP tree. B: Nested exponential operations (right), while unnested usage of the operation is allowed (left). C: Possibility of zero division.

expressions can also be generated by introducing penalty terms in the fitness function of GP, instead of using hard constraints. However, in our opinion, the three constraints must be satisfied to make the expression understandable, and are thus introduced as filters instead of numerical penalty terms.

2.4.3. FIGP procedures

The procedures involved in FIGP are summarized in Fig. 3. There are four components: initial expression generation, selection, evolutionary operation, and fitness calculation. In the initial expression generation, one half of the individuals are generated with the Full method and the other half are generated with the Grow method [13] to ensure a diverse set of individuals. Constant nodes in these individuals are optimized by the NLS method. These individuals (expressions) are filtered out by the V-, F-, and D-filters. This process is repeated until the number of individuals reaches a predetermined population size. Expressions consisting of only numerical parameters are disqualified in this phase. In the selection phase, the expression with the best fitness value passes to the next generation, alongside the expressions selected through tournament selection from five randomly selected individuals. In the evolutionary process, crossover and mutation operations are applied to individuals or pairs of individuals with predefined probabilities. When the same individual is produced as a result of an evolutional operation, another operation is applied until a unique individual is created (up to a predefined number of iterations). In the fitness calculation module, numerical parameters (constants) are optimized by NLS, followed by score calculation. In this study, the negative RMSE is applied to the training dataset as the fitness function to be maximized. This process is repeated for a predefined number of iterations.

Table 2
Experimental conditions for FIGP.

Parameter name	Value(s)
Population size	1000
Number of generations	200
Tree depth for initial population	1–2
Crossover probability	0.7
Mutation probability	0.2
Tree depth for mutation	0–2
Maximum depth	4
Function node types	+, −, ×, ÷, sqrt, square, cube, exp, ln
Function filter	{sqrt}, {square, cube}, {ln, exp}
Tournament size	5

2.5. Experimental conditions for GP

The parameter names and values of the FIGP procedure are listed in Table 2. These values were determined based on trial runs using GP. For the conventional GP modeling, the same parameter values as for FIGP were used.

2.6. Comparison methods

As comparison methods, we considered MLR and support vector regression (SVR) [28] with radial basis function (RBF) and linear kernels. The objective loss function of the SVR is the sum of the norms of the coefficient vectors and the soft margin loss, which leads to robust models even in the presence of outliers. Nonlinear SVR with the RBF kernel has been extensively used for QSAR models [29,30]. Linear SVR and MLR are directly interpretable based on the regression coefficients of the de-

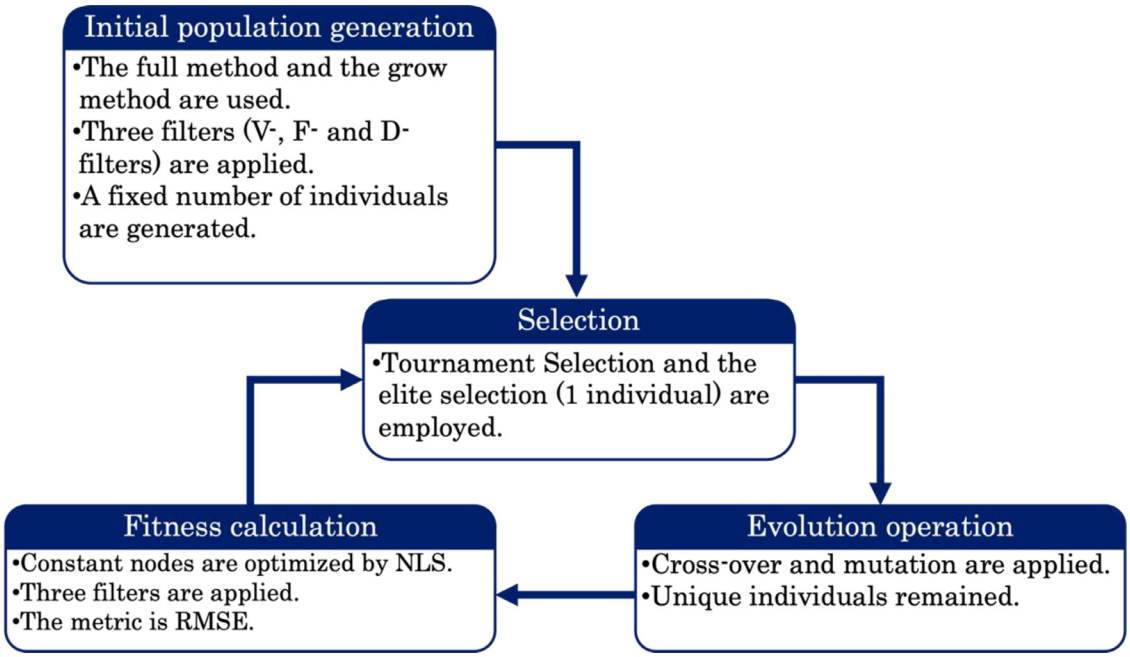

Fig. 3. Procedure of FIGP. Four steps of finding the best SR model are described with methodological points in each step. Initial population generation creates the individuals in the initial population. Selection selects the individuals that survive, followed by evolutional operations and fitness calculations.

scriptors. In this study, the SVR hyperparameters C, ϵ, and γ (only for RBF kernel) were optimized by five-fold cross-validation of the training dataset with the help of Optuna using the TPESampler function [31].

AI Feynman [15], a physics-inspired SR modeling system, was also used for comparison. AI Feynman contains a cascade of filtering processes to generate feasible equations satisfying physics-oriented constraints such as symmetry. AI Feynman can propose expressions for SR models on the Pareto frontier between fitness and complexity.

2.7. Evaluation metrics

The prediction performance was evaluated in terms of the coefficient of determination (R^2), RMSE, and MAE on test datasets.

2.8. Software and implementation of FIGP

FIGP was implemented on top of the DEAP GP library [32]. The code for FIGP, containing the V-, F-, and D- filters, is publicly available in a GitHub repository at https://github.com/takakikatsushi/FIGP, along with example notebooks.

3. Results and discussion

3.1. Study design

The predictive ability of various ML models was compared in terms of the two objective variables of the QED and SAScore. The modeling methods employed in this study were FIGP with the F- and D-filters (FIGP_FD), FIGP with the F-, V-, and D-filters (FIGP_FVD), GP, MLR, and SVR with the RBF kernel (SVR (rbf)) and linear SVR (SVR (linear)). Without the D-filter, the FIGP expressions sometimes output infinite values for test compounds. Thus, FIGP was constrained to include the D-filter.

In the GP algorithm, the division, sqrt, and ln operators were protected from undefined operations, such as zero division. In the DEAP implementation [32], a value of 1 is returned when zero division is attempted. For GP, R^2 values of less than zero for the test datasets were treated as zero for ease of comparison in terms of property prediction.

To understand the effect of the size of the training dataset on the predictive ability and complexity of expressions, the number of training compounds was varied from 50 to 800: {50, 100, 200, 400, 800}. The rest of the 1,000,000 ZINC compounds constituted the test data. For each number of training compounds, five training datasets were randomly compiled, and five prediction trials were conducted. For each training dataset, five GP and FIGP models were built by changing the seed values of the random number generator in GP. The representative model was chosen as that which gave the highest R^2 for the training dataset. Note that cross-validation was not conducted during the training phase because the FIGP and GP models have no hyperparameters to be optimized. Thus, the expression that best explains the training dataset was selected.

The AI Feynman system was only applied to the QED with default parameters. The training dataset size was fixed to 100. For this calculation, a further limited descriptor set was employed as a means of reducing the computational cost and to ensure errorless outputs. The top 13 of 26 descriptors were selected based on the mutual information against y (**Table S1**) for 1,000,000 ZINC compounds. Thus, only meaningful descriptors were employed in this method.

3.2. SR for QED and SAScore

3.2.1. Predictive performance

As a metric of the predictive ability of ML models, the R^2 values produced for the test datasets were measured against the number of training compounds (Fig. 4). Overall, SVR (rbf) shows the best predictive ability. While GP models without any filters give a better fit to the training datasets than those with filters, the R^2 values for the test datasets exhibit large variances, implying that the GP models tend to be overfitted to the training data, especially when the training datasets are small. In contrast, FIGP_FD and FIGP_FVD exhibit stable predictive ability with the various training dataset sizes. For the QED, these two modeling methods consistently outperform SVR (linear) and MLR. In terms of the SAScore, the FIGP models perform as well as MLR, but are inferior to SVR (linear) and SVR (rbf) in terms of R^2 scores. This may be explained by the nature of SAScore: a simple summation of complexity scores, although

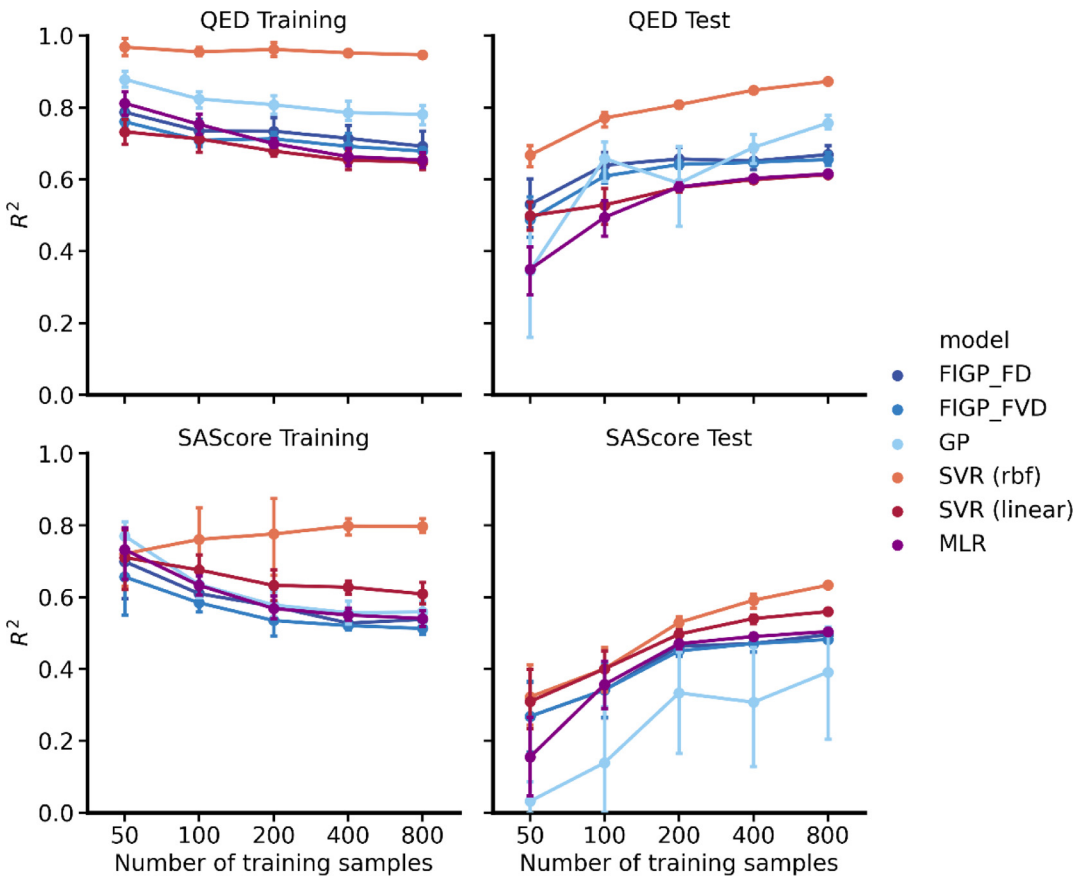

Fig. 4. Predictive capability of SR models. The average R^2 values for the test datasets of QED and SAScore are plotted against the training dataset sizes. Five modeling methods were tested: FIGP_FD, FIGP_FVD, GP, SVR (rbf), and SVR (linear). Error bars represent the 95% confidence intervals based on the results of five trials.

each penalty term is in the logarithmic scale. This assertion is supported by the fact that FIGP_FD and FIGP_FVD exhibit similar performance for this target. By introducing the V- and D-filters into GP, the fitness scores with the training data become slightly worse, whereas the R^2 values for the test datasets remain unchanged.

3.2.2. Convergence of GP-NLS

For both target properties, the fitness values achieved on the training data (RMSE) and the expression diversity were monitored as the computations progressed. The expression diversity was measured in terms of operators and descriptors. For the operator diversity, the ratio of expressions containing specific operators to the total number of expressions, i.e., population size, was monitored. In the same way, for the descriptor diversity, the ratio of expressions containing specific descriptors to the population size was monitored. Fig. 5 reports the expression diversity for the QED when the training seed ID was 0 and the training dataset had a size of 100. Transition plots with other seeds (IDs of 1, 2, 3, and 4) are consistent with that obtained from a seed of 0, as shown in **Figs. S1–S4** in the Supporting Information. Overall, the RMSE values decrease monotonically and converge within 200 generations. The minimum RMSE value is achieved by GP, followed by FIGP_FD and FIGP_FVD. FIGP_FVD exhibits slower convergence than FIGP_FD in terms of RMSE and the descriptor and operator ratios.

Furthermore, as an indirect metric of the degree of overfitting to the training data, the ratio of constant nodes per expression was monitored during the GP progress. Over the five trials for the QED with a training dataset of size 100, the average constant node ratio after convergence is 4.36 (sd: 1.59) for GP, 2.63 (0.74) for FIGP_FD, and 2.16 (0.36) for FIGP_FVD. For the QED, the FIGP models used fewer constant nodes than GP. For the SAScore with a training dataset of size 200, the average constant node ratios are 4.56 (sd: 2.06) for GP, 4.40 (1.54) for FIGP_FD, and 4.15 (1.35) for FIGP_FVD, showing no significant difference.

For the QED, where a nonlinear relation was expected between the chemical structure and the objective variable, GP tends to employ more constant nodes than the other methods.

3.2.3. Expressions returned by AI Feynman

AI Feynman was also used to evaluate the QED with 100 training compounds and a seed of 0. Recall that AI Feynman produces a set of expressions on the Pareto frontier, with a tradeoff between fitness and simplicity of expression (**Table S2**). For most of the Pareto solutions, the R^2 values for the test data were infinite because of undefined functional operations. The simplest expression showed an R^2 value of −0.09 for the training set and negative infinity for the test set. The generated expression was

$$\begin{aligned}\mathrm{QED} = {} & \arccos\left(-0.03\times\mathrm{NAliCc}^2 + 0.03\times\mathrm{NAliCc} + 1.0\right) \\ & + \arctan\left(-0.08\times\mathrm{NSR}^3 + 0.3\times\mathrm{NSR}^2 + 0.11\times\mathrm{NSR} + 0.06\right) \\ & -1.52\times\arccos\left(1.0 - 0.01\times\mathrm{NHOHCount}^2\right) \\ & \times\arctan\left(-0.55\times\mathrm{NAliHc}^3 + 1.84\times\mathrm{NAliHc}^2 - 1.08\times\mathrm{NAliHc} - 0.39\right).\end{aligned} \tag{7}$$

This expression contains two *arccos* functions and two *arctan* functions. The number of aliphatic heterocycles (NAliHc) appears in several terms. Even the simplest expression is hard to interpret through a visual inspection, notwithstanding that it completely fails to explain the training data. For the most complex expression, the R^2 value for the training data set was 0.98, suggesting a good fit to the training data.

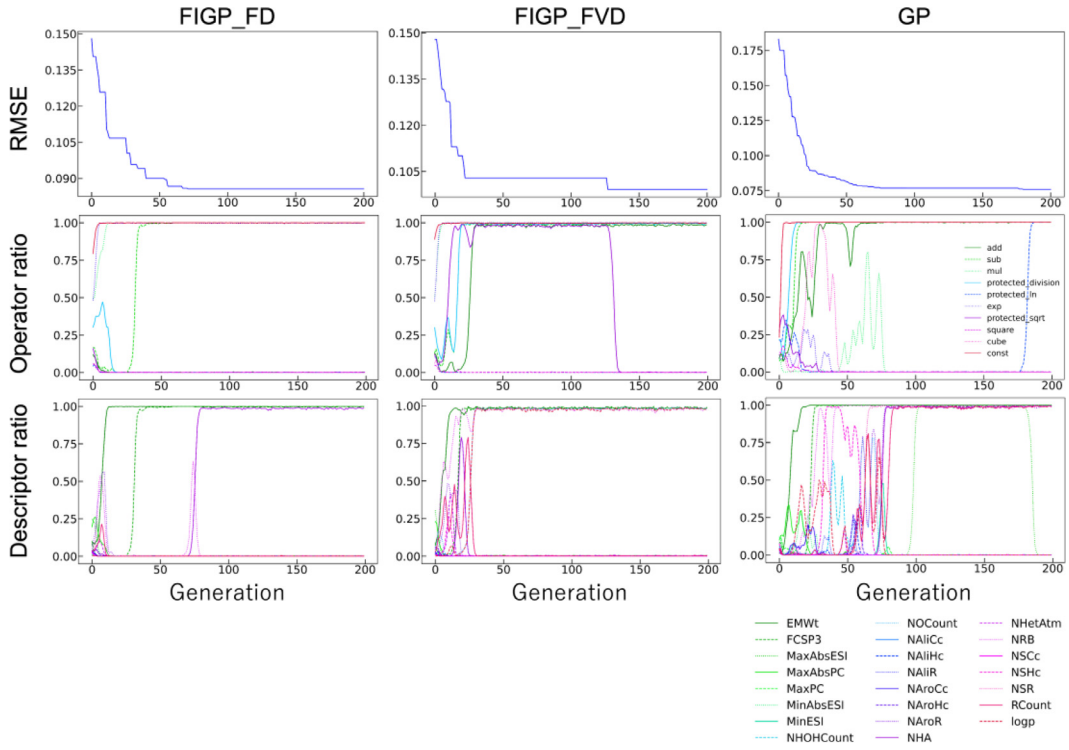

Fig. 5. Convergence of FIGP training process. For the first of five GP trials (seed 0), the converge of the minimum RMSE values are plotted against the generation (top row), the probabilities of using operators in an expression (middle row), and the probabilities of using descriptors in an expression (bottom row). For GP, protected versions of the division, logarithm, and sqrt operation were used.

However, the R^2 value for the test data was −18.22 and the generated expression contained too many terms (274 plus signs and 286 minus signs).

AI Feynman is intended to derive physics formulas from a dataset, with symmetries and separability considered inside the system. This might not be appropriate for QSPR/QSAR analysis, because models for QSPR/QSAR simply approximate the relations between chemical structures and properties/activities. Furthermore, in AI Feynman, a neural network is constructed using a training dataset to detect symmetries and smoothness. To form an accurate response surface, many data points are needed. In our calculation setting, we selected 13 out of 26 variables based on the mutual information and 100 training compounds. This calculation setting might impede AI Feynman from generating "true" expressions.

3.2.4. *Expression analysis*

The three expressions for QED generated by FIGP_FD, FIGP_FVD, and GP from a training seed ID of 0 and a training dataset of size 100 are reported in Table 3. All SR expressions from five trials generated by FIGP_FD, FIGP_FVD, and GP with a training dataset of size 100 are reported in **Tables S3–S5**, respectively. Among the expressions in Table 3, GP gives the best predictive ability, although this is not always true according to Fig. 4. The expression generated by GP is more complicated than those given by FIGP_FD and FIGP_FVD. For example, the effect of molecular weight (EMWt) on the QED prediction values is hard to understand. The expressions from FIGP are similar to each other (**Table S4**). Neither expression contains a variable appearing in more than one term. This is not always true for FIGP_FD, as can be seen from **Table S3** (seed ID1 and ID4). The logarithm of the FIGP_FVD expression in Table 3 becomes the product of $(\mathrm{NAroR}^3 + 7.13 \cdot \mathrm{NRB})$ and $(-0.000694 \cdot \mathrm{NHetAtm} - 0.000694 \cdot \mathrm{logp})$. The effects of the contributing features on the logarithm of QED values differ in scale and combination. For example, the third power of the number of aromatic rings (NAroR) might have an equivalent effect on the QED values as 7.13*NRB (number of rotatable bonds). Likewise, the number of heteroatoms (NHetAtm) and logP values have equivalent effects on the QED values based on this equation. Note that the correlation among descriptors is not considered, although it is likely that these descriptor values cannot be altered independently. Compared with the ground-truth QED definition in Eq. (1), several descriptors appear frequently in the FIGP expressions: NAroR, NRB, and logP. The exponential term in Eq. (1) was correctly identified by FIGP_FVD in all five trials (**Table S4**) and by FIGP_FD in four of the five trials (**Table S3**). However, the generated expressions are not identical to Eq. (1).

Table 3
GP and FIGP expressions for QED. For the first of five GP trials (seed 0) with a training dataset size of 100 for QED, SR expressions along with R^2 for the training and test datasets are listed. The descriptors in the expressions are defined in Table 1.

Method	Expression	Train R^2	Test R^2
FIGP_FD	$\exp(-1.40 \cdot 10^{-6} \cdot \mathrm{EMWt} \cdot (\mathrm{NRB} + 8.02) \cdot (\mathrm{NAroR}^3 + \mathrm{NHetAtm} + 32.9))$	0.72	0.64
FIGP_FVD	$\exp((\mathrm{NAroR}^3 + 7.13 \cdot \mathrm{NRB}) \cdot (-0.000694 \cdot \mathrm{NHetAtm} - 0.000694 \cdot \mathrm{logp}))$	0.70	0.59
GP	$\frac{1.40 \cdot 10^7 \cdot \mathrm{NAliHc} - 2.60 \cdot 10^7 \cdot \mathrm{NAroR} + \frac{\mathrm{EMWt}+9.30 \cdot 10^9}{\mathrm{NHA}+46.3}}{\mathrm{EMWt}^3 - \mathrm{EMWt} + 6.03 \cdot 10^7 + \frac{\mathrm{MinESI}+1.96 \cdot 10^8}{\mathrm{FCSP3}+\mathrm{NAroR}}}$	0.84	0.72

Table 4
GP and FIGP expressions for SAScore. For the first of five GP trials (seed 0) with a training dataset size of 200 for SAScore, SR expressions along with R^2 for the training and test datasets are listed. The descriptors in the expressions are defined in Table 1.

Method	Expression	Train R^2	Test R^2
FIGP_FD	$\mathrm{FCSP3} + 0.0179 \cdot \mathrm{NRB} + \frac{\mathrm{EMWt}-4.99\cdot 10^3}{\mathrm{NHOHCount}+227} + 21.2 + (\mathrm{NAliR} + 131) \cdot \frac{\mathrm{NAliR}+\sqrt{\mathrm{NAroHc}}+2.02}{\mathrm{EMWt}+\mathrm{MaxAbsESI}}$	0.66	0.47
FIGP_FVD	$\mathrm{FCSP3} - \mathrm{MaxPC} + 0.262 \cdot \mathrm{NAroHc} + 0.262 \cdot \mathrm{RCount} + (0.000490 \cdot \mathrm{EMWt} - 0.554) \cdot (\mathrm{NAroR} + 2.44) + 3.25$	0.62	0.44
GP	$0.777 \cdot \mathrm{FCSP3} + 0.0815 \cdot \mathrm{NHOHCount} + 0.102 \cdot \mathrm{NRB} + (0.608 - 0.0395 \cdot \mathrm{NRB}) \cdot (\mathrm{NAliR} + 0.596 \cdot \mathrm{NAroHc}) + 1.19$	0.64	0.46

Table 5
MMS profiles.

ID	Target name	#CPDs	Potency range [pK_i]	Core SMILES
1	Tyrosine-protein kinase ABL	76	[6.4, 10.7]	O=C(Nc1cc2ccc(*)cc2cn1)C1CC1
2	Kappa opioid receptor	83	[5.1, 9.2]	COC(=O)[C@@H]1C[C@H](*)C(=O)[C@H]2[C@@]1(C)CC[C@H]1C(=O)O[C@H](c3ccoc3)C[C@]21C
3	Histamine H3 receptor	53	[6.8, 10.2]	c1cc(*)ccc1OCCCN1CCCCC1

Similar analysis was conducted for SAScore with a dataset size of 200. The expressions generated by the three algorithms are reported in Table 4 for a training seed ID of 0. All SR expressions for the five trials with a training dataset size of 200 are reported in **Tables S6–S8** for FIGP_FD, FIGP_FVD, and GP, respectively. The three expressions in Table 4 indicate comparable predictive performance for the test dataset. All expressions use the fraction of SP3 carbon atoms (FCSP3) as a descriptor with a positive effect on the SAScore values (difficult to synthesize). This descriptor is related to the number of stereo centers, and is thus an important descriptor for SAScore prediction. The FIGP_FVD expression is the simplest, as expected. For FIGP_FD and GP, the effect of NRB on the SAScore is not clear because this descriptor appears in multiple terms in the expressions. Although FIGP_FD produces the most complicated expression in Table 4, GP without any filters generates more complicated expressions based on the number of terms in **Tables S6–S8**.

Table 6
Predictive ability of ML models for training and test datasets.

Data ID	Method	Training		Test	
		R^2	RMSE	R^2	RMSE
1	SVR (rbf)	0.60	0.59	0.19	0.79
	MLR	0.39	0.72	**0.28**	**0.74**
	FIGP_FVD	0.49	0.66	0.26	0.75
2	SVR (rbf)	0.70	0.54	0.34	0.71
	MLR	0.36	0.79	0.24	0.76
	FIGP_FVD	0.51	0.69	**0.46**	**0.64**
3	SVR (rbf)	0.64	0.41	**0.22**	**0.47**
	MLR	0.58	0.43	0.03	0.53
	FIGP_FVD	0.66	0.39	-0.04	0.54

For each dataset, the best predictive performance for the test data is highlighted in bold.

3.3. Demonstration of QSAR modeling

3.3.1. Datasets

For a demonstrative application of FIGP, three sets of substituents with specific cores (analogous compounds) against specific targets were compiled from the ChEMBL database version 29 [33]. Only bioactive compounds annotated with K_i values against specific human target macromolecules were considered. These compound and K_i data were extracted from assays with a confidence score of 9 (highest) and direct binding. Targets with more than 300 bioactive compounds after discarding the upper and lower 10th percentiles in the number of heavy atoms and showing a minimum potency range of 5 were extracted. From these compound datasets, target-wise matching molecular series (MMS) [34] were created with the help of the RDKit community contribution module "mmpa" [25], with a substituent ratio against the core of 0.35. MMS with a single-cut core and containing at least 40 substituents with a minimum potency range of 3.0 were selected. Twelve MMS against eight targets were identified. From the eight targets, three MMS were selected based on their target diversity and potency range. The profiles of the selected MMS are provided in Table 5.

3.3.2. Substituent descriptors

The following seven descriptors were used: number of aromatic rings (NaroR), number of hydrogen bond acceptor/donor atoms (NHA/NHD), logarithm of the octanol/water partition coefficient (logp), rotatable bond counts (NRB), topological polar surface area (TPSA), and molecular weight (MWt). These descriptor values were only calculated for the substituents after replacing the attachment points with carbon atoms. The descriptor calculations were conducted using the Molecular Operating Environment Software ver. 2022.02 [35]. The MMS datasets with these descriptors and potency values, as well as substituent SMILES strings, are provided as tab-separated text in the Supporting Information.

3.3.3. FIGP models

Each MMS dataset was randomly split into training (80%) and test (20%) sets. MLR, SVR (rbf), and FIGP_FVD were employed with the same settings as for property prediction. For FIGP, all the filters were included, and all the data points for each target were used in the D-filter. The goodness-of-fit to the training and test datasets is reported in Table 6. The best modeling methods are different for each dataset. For datasets ID1 and ID3, MLR performs almost as well as FIGP_FVD. For dataset ID3, SVR (rbf) is the best method, whereas for dataset ID2, FIGP_FVD is the best. Table 7 reports the expressions generated by FIGP_FVD. The expression for dataset ID2 is nonlinear. In the numerator, NRB multiplied by MWt has a negative effect on the pK_i prediction. The effect of NHA is less important because it is much smaller than the other constant in the logarithm function. This is consistent with the regression coefficient value of 0.001 for NHA in the MLR model. NAroR is divided by logp in the denominator. The domain of logp contains zero, so this function is not defined for compounds for which logp = 0. Thus, data points for the D-filter should be carefully selected for avoiding invalid operations. The FIGP_FVD expression for dataset ID3 can be expressed by linear combinations of descriptors and polynomial terms. That is why FIGP_FVD and MLR exhibit a similar predictive ability for the test dataset. Overall, FIGP_FVD provides a better fit to the training data than MLR, but the predictive ability of FIGP_FVD is not always better than that of MLR.

Table 7
Mathematical expressions generated by FIGP_FVD.

Data ID	Expression
1	$\text{logp}^2 \cdot (\text{NRB} - 0.476) \cdot (-0.0201 \cdot \text{TPSA} - 0.105) + (1.01 - \frac{23.7}{\text{MWt}}) \cdot (\text{NAroR} + 0.286 \cdot \text{NHD} + 9.74)$
2	$\frac{-0.0541 \cdot \text{NRB} \cdot (\text{MWt} - 137) + 52.3}{\frac{\text{NAroR}}{\text{logp}} + 0.576 \cdot \text{NHD} + \log(\text{NHA} + 1.17 \cdot 10^5)}$
3	$0.0164 \cdot \text{MWt} + 0.0984 \cdot \text{NRB} - 0.152 \cdot \text{logp} - (0.0551 - 0.0510 \cdot \text{NAroR}) \cdot (-4.88 \cdot \text{NHD} + \text{TPSA} - 7.87) + 7.27$

4. Conclusions

This paper has described an interpretable QSAR/QSPR method based on the use of three filters in GP for SR. The V-filter forces every variable to appear no more than once, the F-filter suppresses the recursive usage of functionals, and the D-filter ensures the expression does not output infinite or undefined values when compounds outside the domain of the training dataset are given.

In our proof-of-concept study, the proposed FIGP generated simpler QSPR models than two existing SR methods (AI Feynman and GP). The QSPR expressions given by FIGP provide insights into the original functional forms of the objective variable for the QED and SAScore, while maintaining a distance from the ground-truth expressions. For the QED, FIGP showed better predictive ability than linear regression modeling methods, while for the SAScore, the predictive ability was slightly inferior to that of SVR (linear). The black-box machine learning method of SVM (rbf) exhibited the highest predictive ability.

The mathematical expressions generated by GP can be used to derive gradients. This makes it possible to constrain the generation of expressions to those with smooth response surfaces during evolution. Designing molecules based on the gradient of a compound may be a useful optimization approach. Furthermore, GP contains stochastic operations in nature, so we must determine which expression should be used in practical applications. This selection process may be heuristic, but FIGP has been designed to help humans interpret QSPR/QSAR. It is also possible to derive common features by analyzing multiple generated expressions, which might lead to interpretation of QSPRs/QSARs.

Inside the FIGP architecture, the only criterion tested in this study was the goodness-of-fit of expressions produced using a training dataset (RMSE). Other criteria could be used, such as the Akaike information criterion and Bayesian information criterion. Thus, further research is needed to identify methods for generating simple predictive expressions.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

We thank Swarit Jasial for carefully proofreading a draft of this manuscript. We also thank Ryosuke Asahara for helping us set up computational analyses. This work was supported by a Grant-in-Aid for Transformative Research Areas (A) 21A204 Digitalization-driven Transformative Organic Synthesis (Digi-TOS) from the Ministry of Education, Culture, Sports, Science & Technology, Japan, and was supported by JSPS KAKENHI Grant Number JP20K19922. We thank Stuart Jenkinson, PhD, from Edanz (https://jp.edanz.com/ac) for editing a draft of this manuscript.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ailsci.2022.100046.

References

[1] Polishchuk P. Interpretation of quantitative structure-activity relationship models: past, present, and future. J Chem Inf Model 2017;57(11):2618–39.
[2] Hansch C, Maloney PP, Fujita T, Muir RM. Correlation of biological activity of phenoxyacetic acids with Hammett substituent constants and partition coefficients. Nature 1962;194(4824):178–80.
[3] Hansch C. The advent and evolution of QSAR at Pomona College. J Comput Mol Des 2011;25(6):495–507.
[4] Zahrt AF, Athavale SV, Denmark SE. Quantitative structure-selectivity relationships in enantioselective catalysis: past, present, and future. Chem Rev 2020;120(3):1620–89.
[5] Santiago CB, Guo JY, Sigman MS. Predictive and mechanistic multivariate linear regression models for reaction development. Chem Sci 2018;9(9):2398–412.
[6] Reid JP, Proctor RSJ, Sigman MS, Phipps RJ. Predictive multivariate linear regression analysis guides successful catalytic enantioselective minisci reactions of diazines. J Am Chem Soc 2019;141(48):19178–85.
[7] Ho TK. The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 1998;20(8):832–44.
[8] Cortes C, Vapnik V, Saitta L. Support-vector networks. Mach Learn 1995;20(3):273–97.
[9] Rodríguez-Pérez R, Bajorath J. Interpretation of machine learning models using shapley values: application to compound potency and multi-target activity predictions. J Comput Aided Mol Des 2020;34(10):1013–26.
[10] Balfer J, Bajorath J. Visualization and interpretation of support vector machine activity predictions. J Chem Inf Model 2015;55(6):1136–47.
[11] Tamura S, Jasial S, Miyao T, Funatsu K. Interpretation of ligand-based activity cliff prediction models using the matched molecular pair kernel. Molecules 2021;26(16):4916.
[12] Asahara R, Miyao T. Extended connectivity fingerprints as a chemical reaction representation for enantioselective organophosphorus-catalyzed asymmetric reaction prediction. ACS Omega 2022;7(30):26952–64.
[13] Koza JR. Genetic programming as a means for programming computers by natural selection. Stat Comput 1994;4(2):87–112.
[14] Schmidt M, Lipson H. Distilling free-form natural laws from experimental data. Science 2009;324(5923):81–5.
[15] Udrescu SM, Tegmark M, Feynman AI. A physics-inspired method for symbolic regression. Sci Adv 2020;6(16):eaay2631.
[16] Xie J, Zhang L. Machine learning and symbolic regression for adsorption of atmospheric molecules on low-dimensional TiO_2,. Appl Surf Sci 2022;597:153728.
[17] Weng B, Song Z, Zhu R, Yan Q, Sun Q, Grice CG, Yan Y, Yin WJ. Simple descriptor derived from symbolic regression accelerating the discovery of new perovskite catalysts. Nat Commun 2020;11(1):1–8.
[18] Archetti F, Lanzeni S, Messina E, Vanneschi L. Genetic programming for computational pharmacokinetics in drug discovery and development. Genet Program Evolvable Mach 2007;8(4):413–32.
[19] Archetti F, Giordani I, Vanneschi L. Genetic programming for QSAR investigation of docking energy. Appl Soft Comput 2010;10(1):170–82.
[20] Kommenda M, Burlacu B, Kronberger G, Affenzeller M. Parameter identification for symbolic regression using nonlinear least squares. Genet Program Evolvable Mach 2020;21(3):471–501.
[21] Miyao T, Funatsu K. Finding chemical structures corresponding to a set of coordinates in chemical descriptor space. Mol Inform 2017;36(8):1700030.
[22] Bickerton GR, Paolini GV, Besnard J, Muresan S, Hopkins AL. Quantifying the chemical beauty of drugs. Nat Chem 2012;4(2):90–8.
[23] Ertl P, Schuffenhauer A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. J Cheminform 2009;1(1):1–11.
[24] Sterling T, Irwin JJ. ZINC 15 - ligand discovery for everyone. J Chem Inf Model 2015;55(11):2324–37.
[25] RDKit Open-source cheminformatics. https://www.rdkit.org
[26] Kim S, Chen J, Cheng T, Gindulyte A, He J, He S, Li Q, Shoemaker BA, Thiessen PA, Yu B, et al. PubChem in 2021: new data content and improved web interfaces. Nucleic Acids Res 2021;49(D1):D1388–95.
[27] Dragos H, Gilles M, Alexandre V. Predicting the predictability: a unified approach to the applicability domain problem of Qsar Models. J Chem Inf Model 2009;49(7):1762–76.

[28] Smola AJ, Schölkopf B. A tutorial on support vector regression. Stat Comput 2004;14(3):199–222.
[29] Li L, Wang B, Meroueh SO. Support vector regression scoring of receptor-ligand complexes for rank-ordering and virtual screening of chemical libraries. J Chem Inf Model 2011;51(9):2132–8.
[30] Rodríguez-Pérez R, Bajorath J. Evolution of support vector machine and regression modeling in chemoinformatics and drug discovery. J Comput Aided Mol Des 2022;2022:1–8.
[31] Akiba T, Sano S, Yanase T, Ohta T, Koyama M. Optuna: a next-generation hyperparameter optimization framework. In: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.; 2019. p. 2623–31.
[32] Fortin F-A, Marc-André Gardner U, Parizeau M, Gagné C. DEAP: evolutionary algorithms made easy. J Mach Learn Res 2012;13:2171–5.
[33] Gaulton A, Bellis LJ, Bento AP, Chambers J, Davies M, Hersey A, Light Y, McGlinchey S, Michalovich D, Al-Lazikani B, Overington JP. ChEMBL: a large-scale bioactivity database for drug discovery. Nucleic Acids Res 2012;40:D1100–7.
[34] Wawer M, Bajorath J. Local structural changes, global data views: graphical substructure–activity relationship trailing. J Med Chem 2011;54:2944–51.
[35] MOE (Molecular Operating Environment). Montreal, Canada: Chemical Computing Group Inc; 2022.