Interpreting Machine Learning Pipelines Produced by Evolutionary AutoML for Biochemical Property Prediction

Alex G. C. de Sá
Baker Heart and Diabetes Institute
Melbourne, Victoria, Australia
School of Chemistry & Molecular Biosciences
The University of Queensland
Brisbane City, Queensland, Australia
Alex.deSa@baker.edu.au

Gisele L. Pappa
Computer Science Department
Universidade Federal de Minas Gerais
Belo Horizonte, Brazil
glpappa@dcc.ufmg.br

Alex A. Freitas
School of Computing
The University of Kent
Canterbury, Kent, United Kingdom
a.a.freitas@kent.ac.uk

David B. Ascher
School of Chemistry & Molecular Biosciences
The University of Queensland
Brisbane City, Queensland, Australia
Baker Heart and Diabetes Institute
Melbourne, Victoria, Australia
d.ascher@uq.edu.au

ABSTRACT

Machine learning (ML) has been playing a crucial role in drug discovery, mainly through quantitative structure-activity relationship models that relate molecular structures to properties, such as absorption, distribution, metabolism, excretion, and toxicity (ADMET) properties. However, traditional ML approaches often lack customisation to a particular biochemical task and fail to generalise to new biochemical spaces, resulting in reduced predictive performance. Automated machine learning (AutoML) has emerged to address these limitations by automatically selecting the suitable ML pipelines for a given input dataset. Despite its potential, AutoML is underutilised in cheminformatics, and its decisions often lack interpretability, reducing user trust – especially among non-experts. Accordingly, this paper proposes an evolutionary AutoML method for biochemical property prediction that outputs an interpretable model for understanding the evolved ML pipelines. It combines grammar-based genetic programming with Bayesian networks to guide search and enhance the searched pipelines' interpretability. The evaluation on 12 benchmark ADMET datasets showed that the proposed AutoML method obtained similar or better results than three existing methods. Additionally, the interpretable Bayesian network identified, among the ML pipelines' components generated by the AutoML method (i.e. components like biochemical feature extraction methods, preprocessing techniques and ML algorithms), which components affect the ML pipelines' predictive performance.

CCS CONCEPTS

• **Computing methodologies → Machine learning**; • **Applied computing → Bioinformatics**; • **Theory of computation → Evolutionary algorithms**.

KEYWORDS

Automated Machine Learning (AutoML), Cheminformatics, Drug Discovery, AutoML-generated Pipeline Interpretability, Bayesian Networks

ACM Reference Format:
Alex G. C. de Sá, Gisele L. Pappa, Alex A. Freitas, and David B. Ascher. 2025. Interpreting Machine Learning Pipelines Produced by Evolutionary AutoML for Biochemical Property Prediction. In *Genetic and Evolutionary Computation Conference (GECCO '25 Companion), July 14–18, 2025, Malaga, Spain.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3712255.3734339

1 INTRODUCTION

Machine learning (ML) has been empowering biochemical discovery with predictive methods to accelerate its internal pipelines – e.g., computationally prescreening molecules with adequate properties, proposing new biochemical molecules (e.g., using generative AI, GenAI), and designing new compounds to serve as drugs [1, 15].

In this context, this work primarily focuses on absorption, distribution, metabolism, excretion, and toxicity (ADMET) properties of biochemical compounds due to their relationships with drug design and optimisation [7, 24, 27, 32]. ADMET properties, related to the pharmacokinetics and toxicity of compounds, provide an understanding of how the biochemical compounds move into, through, and out of the body. In addition, these properties assess how the molecules affect the human body by measuring how toxic/safe they are to cells, organs and the genome.

There has been an extensive work on ML for ADMET property prediction [3, 6, 7, 11, 14, 17, 24, 27, 31–33], which links the molecule structure(s) and substructure(s) with ADMET properties, but in

GECCO '25 Companion, July 14–18, 2025, Malaga, Spain

ACM ISBN 979-8-4007-1464-1/2025/07
https://doi.org/10.1145/3712255.3734339

general their methods – and respective models, web servers and/or tools – rely on static and, consequently, non-customisable predictive models. Given the large number of compounds generated daily and the need to predict their properties, this is a significant limitation to the application of these methods to real-world biochemical data.

Automated machine learning (AutoML) [20] has emerged as an efficient alternative to deal with this issue, creating methods that recommend customised ML algorithms and pipelines, which will deliver targeted biochemical property prediction models to the biochemical (property) data at hand. Nevertheless, AutoML has not been broadly used in the field of computational chemistry and cheminformatics, despite its considerable popularity, with only a few recent efforts in this interdisciplinary field [5, 8, 10, 16, 23, 30].

In addition, one of the goals of AutoML is to democratise ML, as it facilitates the use of ML methods, particularly by users with little or no expertise in ML. This is particularly relevant for biochemistry researchers analysing biochemical properties of biomolecules (e.g., ADMET properties). Despite that, most AutoML methods still act as a *black box*, where AutoML algorithmic decisions are not properly interpreted and explained, and so users typically do not understand why an ML algorithm or pipeline has been selected instead of another. Therefore, inexperienced ML users or researchers might lack trust in the recommendations of a *black box* AutoML method.

To mitigate this problem, this work proposes an evolutionary AutoML method for biochemical property prediction which outputs an interpretable model that helps to better understand the ML pipelines produced by the AutoML method. Our proposed method is based on an interpretable Bayesian network classifier (BNC)-based model [13] and a grammar-based genetic programming (GGP) method [22]. While we used a GGP method to evolve valid ML pipelines customised to a biochemical property prediction task, we also employed a BNC model to guide and interpret this evolutionary method's decisions, leading to a broad understanding of the AutoML's recommendations by identifyingthe main components of the proposed AutoML search space that affect the ML pipelines' predictive performance.

12 biochemical ADMET property prediction benchmark datasets were used to validate the proposed model-based GGP method and compare it with alternative methods (i.e., the standard version of the GGP method [8], the pkCSM method [27] and the XGBoost method [2]). The achieved results of this AutoML method on these 12 molecular datasets highlight its predictive power since they show its superior performance in eight (8) out of 12 datasets against alternative methods. These promising results indicate that incorporating the BNC model to guide the GGP method led to a better AutoML search and optimisation. Moreover, the use of a Bayesian network model to assist the GGP algorithm was also relevant to comprehending the AutoML choices of ML pipeline components during the method's evolutionary search, as will be shown in a case study with a benchmarking dataset.

2 RELATED WORK

This section reviews a few works on AutoML related to cheminformatics and/or computational chemistry, focusing on recent works that build and recommend customised predictive pipelines based on biochemical data.

Recently, a few AutoML works have made efforts to automate and, consequently, customise cheminformatics or computational chemistry pipelines through search and optimisation methods, such as the work of de Sá and Ascher [8], AutoQSAR [10], ZairaChem [30], Uni-QSAR [16], QSARtuna [23], and Deepmol [5].

de Sá and Ascher [8] introduced the first evolutionary AutoML method to build and recommend tailored predictive pipelines for small molecule pharmacokinetic data. These pipelines included molecule feature extraction (to define the features for each molecule), feature scaling, feature selection, ML algorithms and hyperparameter optimisation. This AutoML method is based on a context-free grammar [29], which is used by an evolutionary method (i.e., a traditional GGP method) to generate individuals (candidate solutions), perform genetic operations that modify candidate solutions, and guide the evolutionary method in producing only valid solutions (ML pipelines).

AutoQSAR [10], on the other hand, utilises an accuracy score to rank ML pipelines aiming to solve a quantitative structure-activity relationship (QSAR) problem. Nevertheless, AutoQSAR relies on an exhaustive search, which makes it unable to scale to larger datasets. Following a distinct approach, ZairaChem [30] follows open-source ideas to deliver a robust AutoML package for drug screening, employing five optimisation methods for this purpose independently and targeting different objectives (e.g., predictive performance, interpretability and robustness). Although interpretability is used in ZairaChem, it is applied to the final ML models, which is distinct from this paper, which proposes a method that is able to interpret the AutoML-generated pipelines.

Uni-QSAR [16] and QSARtuna [23] are both automated QSAR frameworks for molecule property prediction. In the case of Uni-QSAR, a stacking method is employed to combine the solutions of several ML models and predict molecule properties as a result. By contrast, QSARtuna takes advantage of Bayesian optimisation for the same task. QSARtuna also offers explainability based on the final models found, which is a similar aspect observed in ZairaChem.

Finally, Correia et al. [5] and Li et al. [21] proposed Deepmol and Model Training Engine (MTE), respectively. Both Deepmol and MTE are AutoML frameworks for computational chemistry, considering both traditional machine learning and deep learning models. These frameworks are defined by Bayesian optimisation algorithms that search for and optimise pipelines in the context of drug discovery problems. Deepmol's and MTE's search spaces incorporate a list of options, such as standardisation, feature extraction, feature scaling and selection, machine learning modelling and methods for coping with class imbalance. Similar to ZairaChem and QSARtuna, Deepmol offers explainability, but only to the final predictive model, derived from the AutoML pipeline recommendation.

We emphasise that all the above described AutoML methods for biochemical property prediction do not have an AutoML interpretation component in their workflow, acting like an AutoML *black box* method. When interpretability (or explainability) is incorporated into previous methods (e.g., ZairaChem, QSARtuna and Deepmol), this is performed on the final predictive models produced when running the ML pipeline output by the AutoML method, with no explanation of ML pipelines as a whole. In contrast, we propose the first evolutionary AutoML method that outputs an interpretable

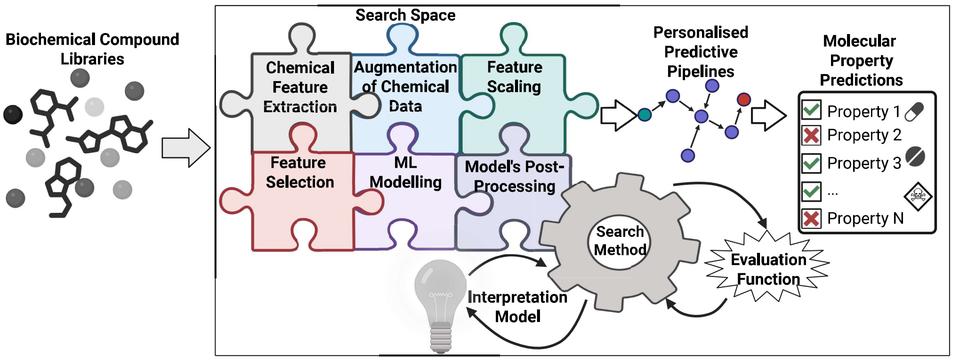

Figure 1: The conceptual workflow for the proposed AutoML method to create ML pipelines customised to the input dataset, targeting ADMET biochemical compound property prediction.

model (based on a BNC) that identifies which ML pipeline components are affecting the predictive performance of the ML pipelines created by the AutoML search. In the next section, we describe the proposed evolutionary AutoML method's workflow in the context of biochemical property prediction.

3 AN EVOLUTIONARY AUTOML METHOD FOR ADMET PROPERTY PREDICTION WITH ML PIPELINE INTERPRETABILITY

Figure 1 illustrates the general conceptual workflow followed by the proposed AutoML method[1]. It starts by receiving the biochemical molecules library and the targeting properties as input. Each molecule in the input set contains its associated (ADMET) property labels. Next, from a defined search space, our AutoML search method identifies the most suitable combination of ML building blocks (i.e., the optimal ML pipeline) for a given biochemical property prediction task (i.e., an input dataset). This search incorporates, e.g., options for biochemical feature extraction, data preprocessing, ML algorithm selection with hyperparameter optimisation, and model post-processing techniques. All these building blocks are automatically selected by the proposed AutoML method, which uses an evaluation function to output the best-performing and customised predictive pipeline to the input biochemical data. The output ML pipeline can then perform predictions (from a learned model) on new molecules. Note that, during the AutoML process, the proposed AutoML method iteratively builds and updates a Bayesian network-based interpretable model representing statistical dependencies between the main ML pipeline components (i.e., biochemical feature extraction techniques, preprocessing methods, ML algorithms) and a pipeline's predictive performance. This model is used to partially guide the generation of new individuals at each generation and to identify the main ML pipeline components affecting the ML pipelines' predictive performance – and, as a consequence, the AutoML predictive performance.

Next, we present details on the main components of the proposed AutoML method. First, we describe its search space, highlighting its main ML building blocks, hyperparameters and algorithm options. Second, we introduce the Bayesian network-based and the evolutionary search method, which finds and optimises ML pipelines in the predefined search space of biochemical property prediction tasks. Finally, we describe its evaluation (or fitness) function.

3.1 Search Space

A context-free grammar [29] defines the search space employed by the proposed AutoML method, consisting of five biochemical extraction methods (and their combinations), five scaling techniques (and the option of not using any scaling on the input dataset), six feature selection approaches (and the option of not using any data feature selection on the dataset) and 10 machine learning algorithms.

An excerpt of the grammar defining the proposed AutoML search space can be found in Frame 1[2]. The grammar is formally presented as a four-tuple <N, T, P, S>, where: N is a set of non-terminals; T is a set of terminals; P is a set of production rules; and S (a member of N) is the start symbol. The production rules in the grammar derive the language by combining the grammar symbols. In addition, the symbol "|" represents a choice, and the non-terminals surrounded by the symbols "[" and "]" are optional, i.e., they can appear or not in the production rules.

The start rule – <Start> in the grammar shown in Frame 1 – defines the four main components of the biochemical property prediction pipeline: (i) molecular representation (captured by the non-terminal <feature_extraction>), (ii) feature scaling, (iii) feature selection, and (iv) machine learning modelling (represented by the non-terminal <ML_algorithms>).

For molecular representation – which relates to the biochemical feature extraction from molecules – 31 different combinations of biochemical compound representation techniques are available. These fall into five main categories that relate to their respective algorithms: molecular descriptors, advanced molecular descriptors, graph-based signatures, fragments, and toxicophores [7, 27]. This search space component essentially determines the features used to characterise compounds based on their biochemical structure. It is worth noting that the AutoML search space can result in a maximum of 266 features based on the combination of biochemical

[1] The official source code for this AutoML method is available at https://github.com/alexgcsa/ecxai_workshop_2025

[2] The complete grammar is available at: https://github.com/alexgcsa/ecxai_workshop_2025/blob/main/grammar/automl.bnf.

```
<Start> ::= <feature_extraction> [<feature_scaling>] [<feature_selection>] <ML_algorithms>
<feature_extraction> ::= General_Descriptors | Advanced_Descriptors | Graph-based_Signatures | Toxicophores | Fragments | General_Descriptors Advanced_Descriptors | General_Descriptors Graph-based_Signatures | ... | General_Descriptors Advanced_Descriptors Graph-based_Signatures Toxicophores Fragments
<feature_scaling> ::= <Normalizer> | <MinMaxScaler> | <MaxAbsScaler> | <RobustScaler> | <StandardScaler>
<Normalizer> ::= Normalizer <norm>
<norm> ::= l1 | l2 | max
...
<StandardScaler> ::= StddScaler <with_mean> <with_std>
<with_mean> ::= True | False
<with_std> ::= True | False
<feature_selection> ::= <Variance_Threshold> | <Select_Percentile> | <Select_FPR> | <Select_FWE> | <Select_FDR> | <Select_RFE>
<Variance_Threshold> ::= VarianceThreshold <threshold>
<threshold> ::= 0.0 | 0.05 | 0.10 | 0.15 | ... | 0.85 | 0.90 | 0.95 | 1.0
...
<ML_algorithms> ::= <AdaBoost> | <Decision_Tree> | <Extra_Tree> | <Random_Forest> | <Extra_Trees> | <Gradient_Boosting> | <Neural_Networks> | <SVM> | <NuSVM> | <XGBoost>
<AdaBoost> ::= AdaBoost <algorithm> <n_estimators> <learning_rate>
<algorithm> ::= SAMME.R | SAMME
<n_estimators> ::= 5 | 10 | 15 | 20 | ...| 300 | 500 | 550 | ... | 950 | 1000 | 1500 | 2000 | 2500 | 3000
<learning_rate> ::= 0.01 | 0.02 | 0.03 | ... | 2.0
...
<XGBoost> ::= XGBoost <n_estimators> <max_depth> <max_leaves> <learning_rate>
<max_depth> ::= 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | None
<max_leaves> = 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10
```

Frame 1: An excerpt of the proposed AutoML grammar.

feature extraction algorithms, which derive features from molecules. Nevertheless, it is part of the AutoML method's task to learn the best-performing combination of feature extraction algorithms to be included in the ML pipeline.

Feature scaling is handled using standard approaches from the scikit-learn library [25, 28], including *Normalizer*, Min-Max Scaler, Max Abs Scaler, Robust Scaler, and Standard Scaler. This step modifies the numerical representation of biochemical compounds, ensuring that feature values are appropriately scaled. However, the grammar also allows the option of bypassing feature scaling, as specified in the <Start> rule.

Feature selection is another key component involving the selection of relevant features using methods from scikit-learn [28]. The grammar includes Variance Threshold, Select Percentile, Recursive Feature Elimination (RFE) and selection methods based on False Discovery Rate (FDR), False Positive Rate (FPR), and Family-Wise Error Rate (FWE). It is worth noting that RFE requires a predictive model on top of the feature selection method. As with scaling, the grammar provides the flexibility to proceed without applying any feature selection method, as defined in the <Start> rule.

The grammar is designed exclusively for classification pipelines [28]. At present, ML modelling consists of 10 algorithms implemented in scikit-learn and other software [2, 25]: Decision Tree, Extremely Randomised Tree (Extra Tree), Extremely Randomised Trees (Extra Trees), Random Forest, Adaptive Boosting (AdaBoost), Gradient Boosting, Neural Networks (using Multi-Layered Perceptron), Support Vector Machines (SVM), Nu-Support Vector Machines (NuSVM) and Extreme Gradient Boosting (XGBoost). Future work includes extending this search space for regression problems.

For all three out of the four main types of components (i.e., feature scaling, feature selection and ML modelling), hyperparameter optimisation is also applied to each level, ensuring the search for hyperparameters' best values is based on the input biochemical data. Considering all available choices and their respective hyperparameters, the AutoML grammar comprises 59 non-redundant production rules, with 58 non-terminals and 389 terminals.

3.2 Search Method

The proposed search method is partially inspired by the Bayesian Optimisation Algorithm (BOA) [26] (which is a specific type of evolutionary algorithm that should *not* be confused with the standard "Bayesian Optimisation" method in AutoML [19, 20]). More precisely, the proposed AutoML search method is a hybrid between BOA and GGP, in the sense that a proportion of the new individuals created at each generation are created by GGP operators, whilst the remaining proportion of individuals is created by sampling from a Bayesian network in a BOA-like style. This Bayesian network will also be used to create an interpretable model (Figure 1) that helps to understand which ML pipeline components influence the overall predictive accuracy of the searched ML pipelines.

Figure 2 illustrates the workflow followed by the proposed AutoML method. First, a dataset of biochemical compounds is set as input to the AutoML method. This dataset could represent any particular biochemical compound property, including but not limited to ADMET properties. Next, the evolutionary method gets its population of individuals (candidate solutions, in our case, machine learning pipelines) initialised at random but following the grammar rules mentioned in Section 3.1. All individuals are converted into scikit-learn pipelines to be evaluated. Based on the results of the evaluation (see Section 3.3 for details) at each generation, we train a Bayesian network classifier (BNC) model [13] representing statistical dependencies between the ML pipelines' components and the pipelines' predictive accuracy.

To learn this BNC model, the AutoML method constructs a binary dataset, where each instance represents an ML pipeline (an individual in the population). Each binary predictive feature represents whether or not a specific ML pipeline component (molecular feature extraction, feature scaling, feature selection and ML technique) is included in each ML pipeline, and the binary target variable indicates whether a pipeline's predictive accuracy is good or bad, which

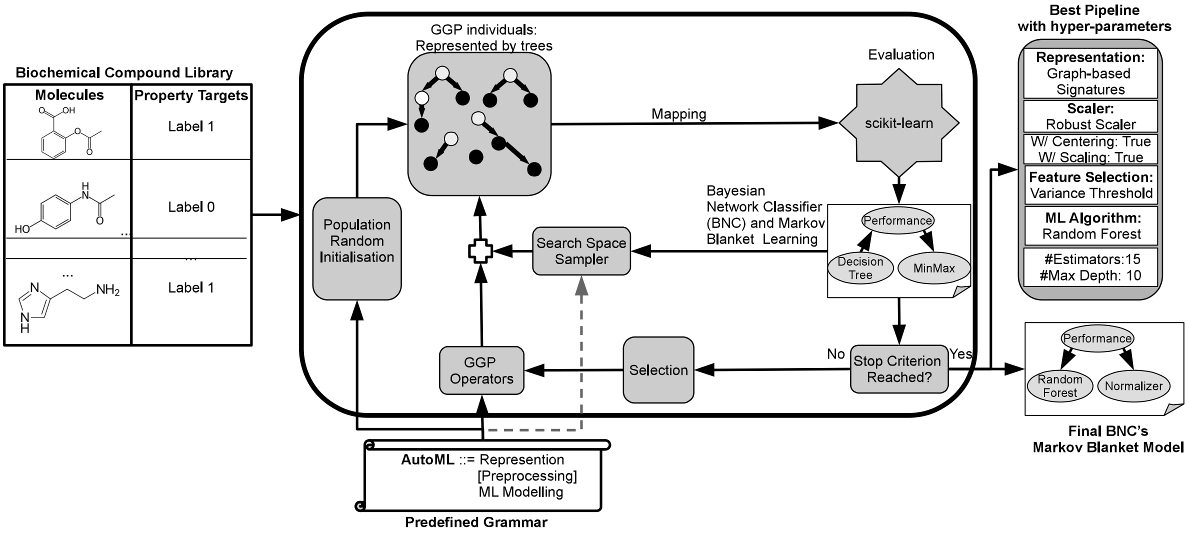
Biochemical Compound Library
Molecules
Property Targets
Label 1
Label 0
Label 1
...
GGP individuals: Represented by trees
Population Random Initialisation
Mapping
Evaluation
scikit-learn
Bayesian Network Classifier (BNC) and Markov Blanket Learning
Performance
Decision Tree
MinMax
Search Space Sampler
GGP Operators
Selection
No
Stop Criterion Reached?
Yes
AutoML ::= Represention [Preprocessing] ML Modelling
Predefined Grammar
Best Pipeline with hyper-parameters
Representation: Graph-based Signatures
Scaler: Robust Scaler
W/ Centering: True W/ Scaling: True
Feature Selection: Variance Threshold
ML Algorithm: Random Forest
#Estimators:15 #Max Depth: 10
Performance
Random Forest
Normalizer
Final BNC's Markov Blanket Model

Figure 2: The grammar-based genetic programming (GGP) method with a Bayesian Network Classifier to search for ML pipelines in the context of biochemical property prediction, particularly for ADMET property prediction.

is defined based on a ranking of all ML pipelines based on their fitness values (see Section 3.3), as follows. A pipeline's accuracy is deemed good if its percentile in the fitness-based ranking is in the range 80%-100% (i.e., it is among the best 20% of all pipelines). A pipeline's accuracy is deemed bad if its percentile in the fitness-based ranking is in the range 0%-60%. We did not consider pipelines with percentiles between 60% and 80% for reducing the level of uncertainty while modelling the pipelines' predictive performance.

From the learned BNC, our AutoML method takes the Markov Blanket [13] of the target node to consider only the most important ML building blocks affecting the AutoML performance. In a Bayesian network, the Markov Blanket of a node is the set of the nodes corresponding to the node's parents, its children and the other parents of its children. The AutoML uses the Markov Blanket of the target node to sample in areas of the search space affecting ML pipelines' predictive performance. To build the BNC from the above described constructed dataset (linking the components of the pipelines generated by the AutoML method until the current generation with the pipelines' predictive performance), we used a hill-climbing algorithm with a Bayesian information criterion as the scoring metric. We used the aGrUM/pyAgrum's implementation for this step [12] to train and build the BNC.

A proportion of the new population will have ML pipelines (individuals) sampled based on the BNC's Markov Blanket, which respects the grammar rules, as indicated in Figure 2 by the Search Space Sampler. Note that, at each generation, both the new individuals (pipelines) created by this sampling and the new individuals created by the GPP algorithm are part of the evolutionary process. The next step involves checking whether the stopping criterion has been met. If it has not, the pipelines (i.e., individuals) undergo selection based on their fitness values, via tournament selection, followed by the application of GGP operators, specifically Whigham's crossover followed by mutation [22]. If both operators are used, mutation modifies the recombined pipelines resulting from the crossover operation. Both crossover and mutation adhere to the grammar constraints, ensuring that only valid individuals are generated. Additionally, elitism is employed, preserving the top n pipelines (individuals) from the previous generation to maintain high-performing solutions.

This evolutionary process, illustrated in Figure 2, continues until the stopping criterion is met. Then, the best-performing pipeline from the final population is returned, along with its optimal hyperparameter values. The final BNC's Markov blanket model is also output to the user, identifying which AutoML search space algorithmic components from generated ML pipelines are affecting the pipelines' predictive performance.

3.3 Fitness Function

The designed fitness function for the proposed AutoML method relies on a repeated k-fold cross-validation procedure to define the fitness of the ML pipeline. More precisely, we run k-fold cross-validation m times on the training set, each time using a different random seed to divide the training set into k folds. I.e., in each cross-validation, an ML pipeline is run k times, each time using a different fold as the validation set and the other $k-1$ folds as the learning set, so that in total each ML pipeline (GGP individual) is run $k \times m$ times, and the average of performance scores over the $k \times m$ validation sets is used as the fitness of the pipeline. In this fitness function, we defined $k = 3$ and $m = 3$. This approach reduces the chances of overfitting during model selection since data is resampled in each iteration of the repeated k-fold cross-validation.

To assess the quality of each pipeline, the fitness function is defined as the average of Matthew's correlation coefficient (MCC) [4] over the repeated cross-validation procedures. MCC is a widely used performance metric in classification tasks, particularly valuable in cases of data imbalance due to its robustness in evaluating model performance, which is typically the case for biochemical datasets. The MCC formula is given in Equation 1.

$$MCC = \frac{((TP \times TN) - (FP \times FN))}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{1}$$

In Equation 1, TP (true positives) is the number of molecules labelled as one (1) and correctly predicted as one (1), TN (true negatives) is the number of molecules labelled as zero (0) and correctly predicted as zer (0), FP (false positives) the number of molecules labelled as zero (0) but incorrectly predicted as one (1), and FN (false negatives) the number of molecules labelled as one (1) but incorrectly predicted as 0 (zero).

MCC accounts for all four types of classification outcomes (TP, TN, FP, and FN), providing a balanced evaluation even in imbalanced learning scenarios. As a correlation coefficient, MCC ranges from -1.0 to +1.0, where +1.0 indicates a perfect positive correlation, -1.0 represents a complete inverse correlation, and 0.0 suggests that the classifier's predictions are no better than random guessing.

4 EXPERIMENTS

This section outlines the key aspects of the AutoML experiments for biochemical data, including a description of the datasets (Section 4.1), the configuration of the grammar-based genetic programming (GGP) method with the Bayesian Network Classifier (Section 4.2), and the benchmarking with alternative methods (Section 4.3).

4.1 Biochemical ADMET Datasets

Twelve (12) biochemical datasets relating to ADMET properties were used to evaluate the performance of the proposed AutoML method. These datasets represent binary classification tasks related to absorption, metabolism, and excretion based on experimental *in vivo* or *in vitro* tests on small molecules. The number of molecules (# Molecules) (i.e., instances) in these datasets varies significantly, ranging from 404 to 18,558. This variation in dataset size presents a challenge for our AutoML method, requiring it to effectively adapt and optimise the search process for different pipeline configurations, including biochemical feature extraction approaches that will lead to complete classification datasets. Table 1 encompasses the employed datasets and their characteristics. It is important to mention that since the AutoML method will select the combination of feature extraction algorithms, Table 1 omits the number of features for each dataset because this will be decided while executing each ML pipeline over the AutoML search and optimisation.

Note that although we used these 12 datasets to validate the AutoML method, any other biochemical predictive task would be suitable to be used. We plan to assess the performance of our AutoML method in a broader range of datasets in future work.

Each dataset was split into two data subsets: training and test data, using a stratified approach. 90% of each dataset (training set) was allocated for searching the optimal pipeline for the respective ADMET dataset, while the remaining 10% (test set) was reserved to evaluate the predictive accuracy of the final selected pipeline.

4.2 Parameter Configuration

The GGP (with BNC) parameters were configured as follows. The entire GGP evolutionary process with a population of 100 individuals, representing ML pipelines, runs for one hour. Each individual has a time budget of 5 minutes to run. Since we are using multiprocessing (#processes = 10), there is a chance of the individual running for more than 5 minutes because of multiprocessing synchronisation. When this occurs, the individual score (representing the ML pipeline score) is multiplied by 0.7 to penalise its execution. Nevertheless, when multiprocessing synchronism happens and the individual/pipeline has not completed its run, its MCC is set to 0.0.

Table 1: Description of the 12 binary classification datasets used in the experiments related to ADMET properties.

ID	Dataset	Abbreviation	Category	# Molecules
1	Caco-2 permeability	Caco-2	Absorption	663
2	P-glycoprotein I Inhibitor	PGP I Inhibitor	Absorption	1223
3	P-glycoprotein II Inhibitor	PGP II Inhibitor	Absorption	1023
4	P-glycoprotein I Substrate	PGP I Substrate	Absorption	1272
5	Skin Permeability	Skin Perm.	Absorption	404
6	Cytochrome P450 CYP2C9 Inhibitor	CYP2C0 Inhibitor	Metabolism	14,706
7	Cytochrome P450 CYP2C19 Inhibitor	CYP2C19 Inhibitor	Metabolism	14,572
8	Cytochrome P450 CYP2D6 Inhibitor	CYP2D6 Inhibitor	Metabolism	14,738
9	Cytochrome P450 CYP2D6 Substrate	CYP2D6 Substrate	Metabolism	666
10	Cytochrome P450 CYP3A4 Inhibitor	CYP3A4 Inhibitor	Metabolism	18,558
11	Cytochrome P450 CYP3A4 Substrate	CYP3A4 Substrate	Metabolism	669
12	Renal Organic Cation Transporter 2 Substrate	OCT2 Substrate	Excretion	904

Given a tournament of size 2, crossover and mutation GGP operators can be applied to selected individuals in the population independently, with probability rates of 0.80 and 0.15, respectively. These operations can also happen jointly, where crossover is followed by mutation at a rate of 0.05. These operator types were set and implemented aiming to increase search diversity. At each generation, the best current individual is also kept for the next generation (i.e., the elitism size is equal to 1).

From the individuals sampled from the BNC's Markov Blanket, we select 10% of the population size (i.e. 10) of new individuals (ML pipelines) to compose the new population. Finally, to reduce the risk of premature convergence, we add a random pipeline into the population if we have cases where at least 70% of the ML pipelines in the population are the same, indicating convergence. Note that the population size is always kept the same, meaning that new individuals are added into the new population (based on elitism, BNC's Markov Blanket convergence) before the loop to derive new (child) individuals via crossover and mutation operators. Table 2 highlights these parameters for the proposed AutoML method.

Table 2: The GGP parameters for the proposed evolutionary AutoML method for evolving a population of machine learning pipelines for biochemical property prediction.

Parameter	Value
Population Size	100
Stopping Criterion to Run the Entire GGP (with BNC)	1 hour
Individual's time budget	5 minutes
Tournament Selection Size	2
Crossover Probability	0.80
Mutation Probability	0.15
Crossover and Mutation Probability	0.05
Elitism Size	1
BNC Sampled Population Percentage	10 (10%)
Population Similarity Rate to Add Random Individuals	70%

4.3 Benchmarking

We evaluated the best ML pipelines discovered by the proposed AutoML method (described in Section 3) by comparing them against the results of three alternative methods, as detailed below. For all methods, for each dataset, the results reported in Section 5 refer to the MCC values obtained by those methods on the same test set (10% of the full dataset), but there are some differences in the methodology used to train some methods, as follows.

First, the results of the proposed AutoML method were compared against the results of the GGP method proposed by de Sá and Ascher [8]. Both methods were trained as follows. Each method was run 20 times on the training set, and the best ML pipeline (with the highest fitness on the training set) of each run was identified. Then, the best overall ML pipeline (again, with the highest fitness on the training set) among those 20 best-of-run pipelines was selected as the final solution, and its predictive performance (MCC value) was measured on the test set. Hence, the comparison between the results of the proposed AutoML and the method in [8] is fair, as both used the same training methodology and the same test set.

The results of our AutoML method were also compared against the results of pkCSM [27], a well-known method for predicting biochemical ADMET properties. pkCSM models were trained and assessed using the same training and test sets used by our AutoML method. To learn a pkCSM model for each of the datasets in Table 1, we used the same ML algorithm (and hyperparameter settings) that was used in previous work for each of those datasets – namely: Logistic Model Tree for the Caco-2 dataset, Simple Logistic for the CYP2D6 Substrate and CYP3A4 Substrate datasets, and Random Forest for the other nine (9) datasets in Table 1. Note that the comparison between our AutoML method and pkCSM is not completely fair because pkCSM was not run 20 times on the training set. However, the comparison is broadly fair because both our AutoML method and pkCSM used the same training set to learn predictive models, and their models were evaluated on the same test set.

The results of the proposed AutoML method were also compared against the results of XGBoost [2] with default hyperparameter settings. XGBoost was chosen due to its widespread use in machine learning and its frequent application in predictive modelling. This comparison is not completely fair because XGBoost was applied to the training set only once, rather than 20 times as for the proposed AutoML method, but the comparison is broadly fair because both methods were applied to the same training set and their results were evaluated on the same test set.

Finally, to statistically compare all four (4) methods, we applied Iman-Davenport's modification of Friedman's test [9]. If the test yielded significant results, we conducted a Nemenyi *post hoc* test to perform pairwise method comparisons.

5 RESULTS

First, in Section 5.1, we compare the predictive performance of 4 methods: the proposed AutoML method, the work of de Sá and Ascher [8], pkCSM [27], and XGBoost [2]. Next, in Section 5.2, we analyse a case study on the dataset *Caco-2* to understand how the evolutionary process is taking advantage of the BNC to evolve ML pipelines better and to interpret the AutoML decisions.

5.1 Benchmarking against Alternative Methods

The MCC values of the proposed AutoML method and of the above 3 alternative methods are shown in Table 3. Notice that our evolutionary AutoML method achieved the highest average MCC (0.618) across all datasets, outperforming de Sá and Ascher (0.530), pkCSM (0.520), and XGBoost (0.497). In terms of average ranking, our AutoML method is also ranked as the best method (rank 1.75), followed by de Sá and Ascher (2024) (2.25), pkCSM (2.50) and XGBoost (3.50).

In addition, as shown in Table 3, our AutoML method outperformed all other methods in eight (8) out of 12 datasets (Caco-2, PGP II Substrate, Skin Permeability, CYP2C9 Inhibitor, CYP2C19 Inhibitor, CYP2D6 Substrate, CYP3A4 Inhibitor and OCT2 Substrate). It had a particularly strong lead in CYP3A4 Inhibitor (0.958), where the second-best method (de Sá and Ascher) achieved only 0.590.

Table 3: Predictive accuracy (MCC values) results on the test set for the four compared methods (see Subsection 4.3).

ID	Dataset	Proposed Method	de Sá and Ascher, 2024	pkCSM	XGBoost
1	Caco-2	**0.641**	0.610	0.613	0.579
2	PGP I Inhibitor	0.799	0.837	0.854	0.820
3	PGP II Inhibitor	0.703	0.783	0.816	0.696
4	PGP II Substrate	**0.367**	0.289	0.351	0.232
5	Skin Perm.	**0.588**	0.394	0.462	0.368
6	CYP2C9 Inhibitor	**0.668**	0.615	0.572	0.553
7	CYP2C19 Inhibitor	**0.764**	0.647	0.641	0.590
8	CYP2D6 Inhibitor	0.540	0.556	0.521	0.488
9	CYP2D6 Substrate	**0.581**	0.334	0.197	0.267
10	CYP3A4 Inhibitor	**0.958**	0.590	0.542	0.534
11	CYP3A4 Substrate	0.145	0.274	0.290	0.440
12	OCT2 Substrate	**0.661**	0.427	0.384	0.402
Average		**0.618**	0.530	0.520	0.497
Ranking		**1.75**	2.25	2.50	3.50

Regarding statistical analysis, the Friedman Test (with Iman-Davenport correction) reports a statistically significant difference (p-value =0.004307), indicating that at least one method performs significantly differently from the others. Hence, we proceed with a *post hoc* analysis with the Nemenyi Test for pairwise method comparisons. First, we calculate the critical difference [9], which is 1.407. Given the ranking differences between each pair of methods in Table 3, we noted that the proposed AutoML method's superiority over XGBoost (whose ranking difference to our AutoML method is 1.75) is statistically significant. There was no statistically significant difference between the rankings of the other pairs of methods.

Furthermore, recall that our proposed evolutionary AutoML method was designed not only to deliver good predictive performance, but also to explain the produced ML pipelines across the AutoML process. With Bayesian Network Classifiers (BNCs) and their respective Markov Blankets guiding the decisions of the evolutionary AutoML process, the proposed method allows us to obtain a better understanding of that process, as discussed in Section 5.2.

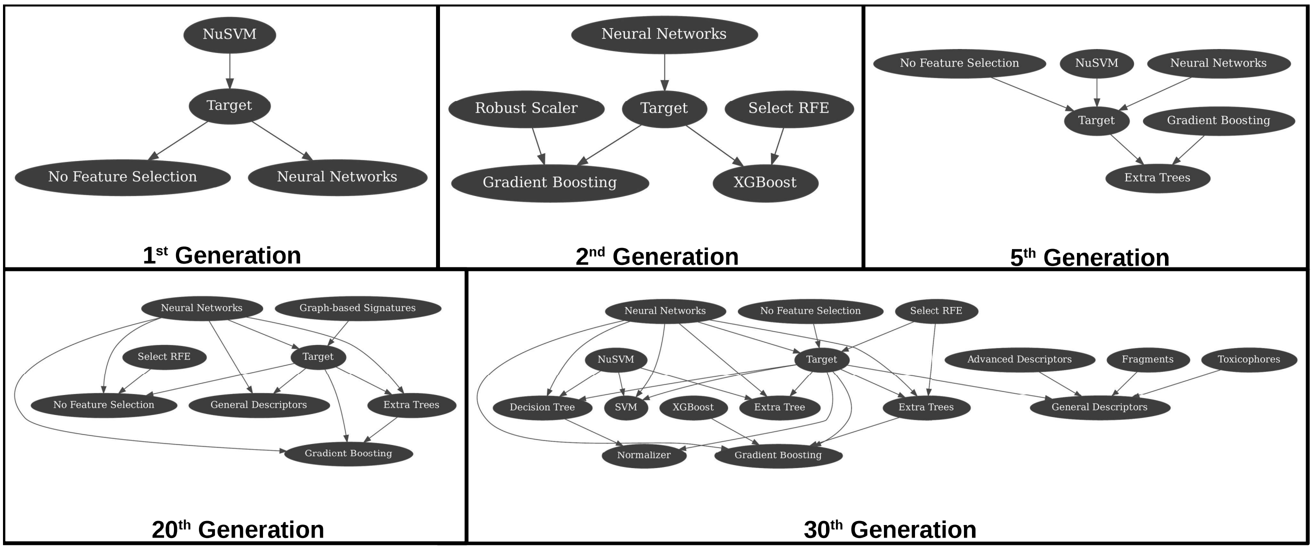

Figure 3: The built Markov Blankets of the Bayesian networks across generations of a run of our evolutionary AutoML method on the Caco-2 dataset.

5.2 Interpreting the ML Pipelines Found by AutoML through Markov Blankets

Figure 3 depicts the Markov Blankets of the target node in the Bayesian Network Classifiers (BNCs) throughout the evolutionary process of the proposed AutoML method in a run on the Caco-2 dataset. Apart from using these BNCs to sample from regions of the search space that are more prominent to improve predictive performance, they can also be used to understand the AutoML pipeline decisions that the GGP algorithm is (broadly) making over time.

We can observe in Figure 3 (for an example run) that the BNC's Markov Blankets evolve from simpler to more complex, as data regarding ML pipeline evaluations targeting ADMET properties increases over time. For example, in the first generation, performing feature selection is not in the Markov blanket of the target variable (which maps the accuracy of the ML pipelines). So, feature selection is not affecting the target variable, as opposed to the second generation. However, there is an alternation of pipeline components over time. The 30th generation shows the power of the choice of ML algorithms to affect the pipelines' predictive accuracy, as most of the nodes of the Markov blanket of the target variable are actually ML algorithm components of a pipeline, even if using or not using feature selection (Recursive Feature Elimination and No Feature Selection are present in the 30th generation).

6 CONCLUSIONS AND FUTURE WORK

This paper introduced a novel, robust evolutionary AutoML method for predicting biochemical (ADMET) properties. Our method was designed not only to recommend good predictive pipelines – composed of molecular representation, scaling, feature selection and ML modelling components – but also to have its pipeline decisions easily explained by an evolutionary algorithm guided by a BNC.

Preliminary results on 12 ADMET datasets demonstrate our proposed method's capabilities in selecting and configuring ML pipelines for ADMET predictive tasks, although not limited to them. The performed analysis of our AutoML method's results against alternative methods in terms of Matthew's correlation coefficient ranks it as the best method in outputting predictive pipelines that will yield good models and results. Our results also indicate the positive side of using a Bayesian network to model ML pipeline performance and use it to both sample from the search space and (broadly) explain AutoML decisions.

Although these results indicate a good step forward in proposing new specific AutoML methods for cheminformatics and biochemical property prediction problems, we still plan to improve the search and optimisation methods in the evolutionary AutoML method to improve its predictive performance. In fact, a future study could be to design the BNC differently. For instance, instead of interactively and locally building the Bayesian Networks across the evolutionary process, we could perform a complete study *a priori* and use it to build a better model to guide our method's evolutionary process.

Moreover, we plan to compare our AutoML method to other methods in future work, such as ZairaChem [30], Uni-QSAR [16], QSARtuna [23], Deepmol [5] and Model Training Engine (MTE) [21], to have more comprehensive results on our method's performance. In this evaluation, we will standardise the comparison by using the Therapeutics Data Commons ADMET datasets [18].

FUNDING

This work was funded by an Investigator Grant from the National Health and Medical Research Council of Australia (GNT1174405), the Victorian Government's Operational Infrastructure Support Program, CNPq, and FAPEMIG.

REFERENCES

[1] Carrie Arnold. 2023. Inside the nascent industry of AI-designed drugs. *Nature Medicine* 29 (2023), 1292–1295.

[2] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: a scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 785–794.

[3] Feixiong Cheng, Weihua Li, Yadi Zhou, Jie Shen, Zengrui Wu, Guixia Liu, Philip W. Lee, and Yun Tang. 2012. admetSAR: a comprehensive source and free tool for assessment of chemical ADMET properties. *Journal of Chemical Information and Modeling* (2012), 3099––3105.

[4] Davide Chicco and Giuseppe Jurman. 2020. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* 21 (2020), 1–13.

[5] João Correia, João Capela, and Miguel Rocha. 2024. DeepMol: An Automated Machine and Deep Learning Framework for Computational Chemistry. *Journal of Cheminformatics* 16, 1 (2024).

[6] Antoine Daina, Olivier Michielin, and Vincent Zoete. 2017. SwissADME: a free web tool to evaluate pharmacokinetics, drug-likeness and medicinal chemistry friendliness of small molecules. *Scientific Reports* 7, 1 (2017), 42717.

[7] Alex G.C. de Sá, Yangyang Long, Stephanie Portelli, Douglas E.V. Pires, and David B. Ascher. 2022. toxCSM: comprehensive prediction of small molecule toxicity profiles. *Briefings in Bioinformatics* 23, 5 (2022), bbac337.

[8] Alex G. C. de Sá and David B. Ascher. 2024. Towards Evolutionary-based Automated Machine Learning for Small Molecule Pharmacokinetic Prediction. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, 1544–1553.

[9] Janez Demšar. 2006. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research* 7 (2006), 1–30.

[10] Steven L Dixon, Jianxin Duan, Ethan Smith, Christopher D Von Bargen, Woody Sherman, and Matthew P Repasky. 2016. AutoQSAR: an automated machine learning tool for best-practice quantitative structure-activity relationship modeling. *Future Medicinal Chemistry* 8, 15 (2016), 1825–1839.

[11] Jie Dong, Ning-Ning Wang, Zhi-Jiang Yao, Lin Zhang, Yan Cheng, Defang Ouyang, Ai-Ping Lu, and Dong-Sheng Cao. 2018. ADMETlab: a platform for systematic ADMET evaluation based on a comprehensively collected ADMET database. *Journal of Cheminformatics* 10 (2018), 1–11.

[12] Gaspard Ducamp, Christophe Gonzales, and Pierre-Henri Wuillemin. 2020. aGrUM/pyAgrum: a toolbox to build models and algorithms for Probabilistic Graphical Models in Python. In *International Conference on Probabilistic Graphical Models*. PMLR.

[13] Nir Friedman, Dan Geiger, and Moises Goldszmidt. 1997. Bayesian network classifiers. *Machine learning* 29 (1997), 131–163.

[14] Li Fu, Shaohua Shi, Jiacai Yi, Ningning Wang, Yuanhang He, Zhenxing Wu, Jinfu Peng, Youchao Deng, Wenxuan Wang, Chengkun Wu, et al. 2024. ADMETlab 3.0: an updated comprehensive online ADMET prediction platform enhanced with broader coverage, improved performance, API functionality and decision support. *Nucleic Acids Research* (2024), gkae236.

[15] Amit Gangwal and Antonio Lavecchia. 2024. Unleashing the power of generative AI in drug discovery. *Drug Discovery Today* 29, 6 (2024), 103992.

[16] Zhifeng Gao, Xiaohong Ji, Guojiang Zhao, Hongshuai Wang, Hang Zheng, Guolin Ke, and Linfeng Zhang. 2023. Uni-QSAR: an Auto-ML Tool for Molecular Property Prediction. *arXiv preprint arXiv:2304.12239* (2023).

[17] Yaxin Gu, Zhuohang Yu, Yimeng Wang, Long Chen, Chaofeng Lou, Chen Yang, Weihua Li, Guixia Liu, and Yun Tang. 2024. admetSAR3.0: a comprehensive platform for exploration, prediction and optimization of chemical ADMET properties. *Nucleic Acids Research* (2024), gkae298.

[18] Kexin Huang, Tianfan Fu, Wenhao Gao, Yue Zhao, Yusuf Roohani, Jure Leskovec, Connor W Coley, Cao Xiao, Jimeng Sun, and Marinka Zitnik. 2021. Therapeutics Data Commons: Machine learning datasets and tasks for drug discovery and development. *arXiv preprint arXiv:2102.09548* (2021).

[19] Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. 2011. Sequential model-based optimization for general algorithm configuration. In *Proceedings of the International Conference on Learning and Intelligent Optimization*. Springer, 507–523.

[20] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. 2019. *Automated machine learning: methods, systems, challenges*. Springer Nature.

[21] Xinqi Li, Sergio Pascual-Diaz, Calum Hand, Rory Garland, Waseem Abbas, Faiz M. Khan, Nikhil M. Das, Vedant Desai, Mohamed AbouZleikha, and Matthew Clark. 2025. Scalable Drug Property Prediction via Automated Machine Learning. *ChemRxiv* (2025).

[22] Robert I McKay, Nguyen Xuan Hoai, Peter Alexander Whigham, Yin Shan, and Michael O'neill. 2010. Grammar-based genetic programming: a survey. *Genetic Programming and Evolvable Machines* 11 (2010), 365–396.

[23] Lewis Mervin, Alexey Voronov, Mikhail Kabeshov, and Ola Engkvist. 2024. QSARtuna: an automated QSAR modeling platform for molecular property prediction in drug design. *Journal of Chemical Information and Modeling* 64, 14 (2024), 5365–5374.

[24] YooChan Myung, Alex G. C. de Sá, and David B. Ascher. 2024. Deep-PK: deep learning for small molecule pharmacokinetic and toxicity prediction. *Nucleic Acids Research* (2024), gkae254.

[25] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.

[26] Martin Pelikan and Martin Pelikan. 2005. Bayesian Optimization Algorithm. *Hierarchical Bayesian optimization algorithm: toward a new generation of evolutionary algorithms* (2005), 31–48.

[27] Douglas E.V. Pires, Tom L. Blundell, and David B. Ascher. 2015. pkCSM: predicting small-molecule pharmacokinetic and toxicity properties using graph-based signatures. *Journal of Medicinal Chemistry* 58, 9 (2015), 4066–4072.

[28] Sebastian Raschka and Vahid Mirjalili. 2019. *Python machine learning*. Packt publishing ltd.

[29] Michael Sipser. 2012. *Introduction to the theory of computation* (3rd ed.). Cengage Learning.

[30] Gemma Turon, Jason Hlozek, John G Woodland, Ankur Kumar, Kelly Chibale, and Miquel Duran-Frigola. 2023. First fully-automated AI/ML virtual screening cascade implemented at a drug discovery centre in Africa. *Nature Communications* 14, 1 (2023), 5736.

[31] Yu Wei, Shanshan Li, Zhonglin Li, Ziwei Wan, and Jianping Lin. 2022. Interpretable-ADMET: a web service for ADMET prediction and optimization based on deep neural representation. *Bioinformatics* 38, 10 (2022), 2863–2871.

[32] Guoli Xiong, Zhenxing Wu, Jiacai Yi, Li Fu, Zhijiang Yang, Changyu Hsieh, Mingzhu Yin, Xiangxiang Zeng, Chengkun Wu, Aiping Lu, et al. 2021. ADMETlab 2.0: an integrated online platform for accurate and comprehensive predictions of ADMET properties. *Nucleic Acids Research* 49, W1 (2021), W5–W14.

[33] Hongbin Yang, Chaofeng Lou, Lixia Sun, Jie Li, Yingchun Cai, Zhuang Wang, Weihua Li, Guixia Liu, and Yun Tang. 2019. admetSAR 2.0: web-service for prediction and optimization of chemical ADMET properties. *Bioinformatics* 35, 6 (2019), 1067–1069.