

AutoMolDesigner for Antibiotic Discovery: An AI-based Open-source Software for Automated Design of Small-molecule Antibiotics

Tao Shen[¶], Jiale Guo[¶], Zunsheng Han[¶], Gao Zhang[¶], Qingxin Liu^{¶[○]}, Xinxin Si[○], Dongmei Wang^{¶}, Song Wu^{¶*}, Jie Xia^{¶*}*

[¶]State Key Laboratory of Bioactive Substance and Function of Natural Medicines, Institute of Materia Medica, Chinese Academy of Medical Sciences and Peking Union Medical College, Beijing 100050, China; [○]School of Pharmacy, Jiangsu Ocean University, Lianyungang, Jiangsu 222005, China; *corresponding authors.

Correspondence should be addressed to J.X. (jie.william.xia@hotmail.com)

Dr. Jie Xia

Institute of Materia Medica,
Chinese Academy of Medical Sciences
No. 2 Nanwei Road
Beijing 100050, China

ABSTRACT: Discovery of small-molecule antibiotics with novel chemotypes serves as one of the essential strategies to address antibiotic resistance. Although a considerable number of computational tools committed to molecular design have been reported, there is a deficit in the holistic and efficient tool specifically developed for small-molecule antibiotic discovery. To address this issue, we report AutoMolDesigner, a computational modeling software dedicated to small-molecule antibiotic design. It is a generalized framework comprising two functional modules, i.e., generative deep learning-enabled molecular generation and automated machine learning based-antibacterial activity/property prediction, wherein individually trained models and curated datasets are out-of-the-box for whole cell-based antibiotic screening and design. It is open-source thus allows for the incorporation of new features for flexible use. Unlike most software programs based on Linux and command lines, this application equipped with Qt-based graphical user interface can be run on personal computers with multiple operating systems, making it much easier to use for experimental scientists. The software and related materials are freely available at GitHub (<https://github.com/taoshen99/AutoMolDesigner>) and Zenodo (<https://zenodo.org/record/8366085>).

KEYWORDS: antibiotic discovery; *de novo* drug design; automated machine learning; graphical user interface

INTRODUCTION

The evolutionary multidrug-resistant (MDR) bacteria have become an emerging threat to public health¹. Among these bacteria, the ESKAPE (*Enterococcus faecium*, *Staphylococcus aureus*, *Klebsiella pneumoniae*, *Acinetobacter baumannii*, *Pseudomonas aeruginosa*, and *Enterobacter*) pathogens² account for many fatal infections. Known chemotypes of antibiotics rapidly lose efficacy to these pathogens due to the developed resistance mechanisms through natural selection³. Accordingly, novel chemotypes of antibiotics that are widely acknowledged as effective to overcome antibiotic resistance are at an urgent need. In history, novel-chemotype antibiotics are mainly discovered by three strategies. The first strategy dates back to half a century ago, the exploitation of natural products (secondary metabolites) isolated from environmental microbes contributed to most of the small-molecule antibiotics in clinic⁴. However, this strategy is becoming insufficient now since the same chemotype (skeleton) is repetitively discovered and structural modification does not lead to better antibiotics than the natural product itself⁵. With the emerging of combinatorial chemistry at the beginning of the 21st century, the second strategy was target assay-based high-throughput screening (HTS)¹ of large-scale chemical library. The target-based approach achieved notable success in drug research and development for other diseases. Unfortunately, this strategy seems not fruitful in antibiotic discovery, as only three new chemotypes of antibiotics received approval⁶⁻⁸ in the last 20 years and two of them are still derived from natural products. The third strategy is phenotype-based antibiotic screening, in particular the whole cell-based antibiotic screening, i.e. directly screening chemical libraries for bioactive compounds against the whole bacteria instead of a protein target, which holds the promise to circumvent the pitfall of target assay-based screenings that do not consider antibiotic-likeness⁹.

Machine learning (ML), as the core of AI, is comprised of three basic subfields including supervised learning, unsupervised learning and reinforcement learning (RL). Correspondingly, the application of ML to whole cell-based antibiotic discovery can be divided into two aspects according to the applied algorithms¹⁰. Quantitative structure–activity relationships (QSAR) modeling (also called molecular bioactivity prediction) is the most important supervised learning task in this field¹¹. It utilizes whole-cell activity prediction models to prioritize compounds from chemical libraries. A few cases in antibacterial hit identification have been reported. For example, Wang *et al.* utilized molecular descriptors and fingerprints to characterize known 2,066 compounds with activity against methicillin-resistant *S. aureus*, and they built models with four classical ML algorithms including naive Bayesian, support vector machine, recursive partitioning, and k-nearest neighbors¹². The best models were eventually applied to virtual screening of a chemical library comprised of about 7,500 compounds, which contributed to several antibacterial hits. Recently, molecular property prediction based on deep learning-enabled automated featurization was reported¹³. Chemprop is developed to extract bond-level topological information from graph representation based on the algorithm of directed-message passing neural network (D-MPNN)¹⁴. With two kinds of cell-based antibiotic datasets compiled for model training, Chemprop was respectively applied to virtual screening campaigns for new antibiotics. As a result, the research teams have discovered two novel hits which are either broad-spectrum antibiotics¹⁵ or narrow-spectrum antibiotics against *A. baumannii* with a new mode of action¹⁶. With the renaissance of artificial intelligence (AI), *de novo* design of antibacterial molecules becomes an emerging field, wherein self-supervised learning, a kind of unsupervised learning, has been leveraged for molecular generation. Currently, most research is focused on *de novo* design of antimicrobial peptides and those applied strategies are pretty effective¹⁷. As few cases of success

were reported, deep learning-based design of small-molecule antibiotics seems quite challenging. Segler *et al.* pioneered the framework of recurrent neural network (RNN)-based chemical language models (CLMs) for small-molecule generation¹⁸. In their study, the *in silico* design of novel small-molecules antibiotics against *S. aureus*. was used to illustrate the “Design-Synthesis-Test Cycle”. The ML-based prediction model and the transfer learning strategy were coupled to generate a focused chemical library including 60,988 virtual molecules with potential antibacterial activity. This important work provided an early insight into the small-molecule antibiotic design through generative deep learning. However, their ML-based prediction model covered antibacterial activity only and neglected antibiotic-likeness, thus the designed molecules may not be ideal. In terms of public availability, no source code or application was provided. Recently, Sowmya *et al.* developed a semi-supervised deep molecular generation model to explore structure-based molecular design for *Mycobacterium tuberculosis*. A key ML model for drug-target affinity prediction was used to reward the generative process through RL¹⁹. They have shown that more than 75% generated molecules are highly analogous to known inhibitors and contained the essential pharmacophore features that would interact with the binding site residues of chorismate mutase. Unfortunately, they did not take the whole-cell antibacterial activity into consideration but only emphasized target binding of the designed molecules. Therefore, it is not convincing that the designed molecules by the model are active for the bacteria. Similar to the work of Segler *et al.*, their model was not made publicly available as well.

To the best of our knowledge, the current development of AI-based computational tools for antibiotic design and screening is far from satisfaction. Two main issues are as follows: (1) All the whole cell-based models in previous study are mainly focused on antibacterial activity profiling while lack the consideration of antibiotic-likeness²⁰, a significantly important aspect of antibiotics.

This aspect is also overlooked in molecular generation-based small-molecule antibiotic design from a target structure-centered perspective. (2) The computational tools and models for antibiotic design and screening are not easy to access by users. Most of them required programming skills and high-performance server to deploy for application. Recently, some web servers such as AIScaffold²¹ and Chemistry42²² were developed for molecular design. Basically, they are applicable to the building of antibiotic-focused AI models, but they all require commercial licenses.

To address these issues, we developed AutoMolDesigner for small-molecule antibiotic design and screening, which consisted of two functional modules, i.e., a remastered RNN aiming at focused library generation for small-molecule antibiotics and an automated machine learning (AutoML) framework named AutoGluon²³ for molecular property prediction. Both classification and regression models are out-of-the-box for accurate prediction of antibacterial activity and antibiotic-likeness, with cytotoxicity and plasma protein binding (PPB) as representatives. To make the models easy to use, we incorporated the modules including data curation, model evaluation and molecular visualization into AutoMolDesigner. In addition, we designed an intuitive graphical user interface (GUI) for both Windows and MacOS platform, along with the corresponding command-line interface (CLI). This software was also tested on personal computers with various device configurations. It can be run on a laptop without independent graphics processing unit (GPU) and achieve close performance to that on high-performance servers.

RESULTS AND DISCUSSION

Three main features of AutoMolDesigner are summarized below:

- (1) A generalized framework was implemented to realize automated molecular design through focused library generation and molecular property prediction;

- (2) Specific datasets and models for antibiotic discovery were established for the out-of-box use.
- (3) Software operating logics were simplified, intuitive GUI was provided and fluent user experience on personal computer was guaranteed.

Figure 1 depicts the workflow of this software, and corresponding interfaces of two main sub-tools are displayed in Figure S1. Herein, we take the automated design of antibiotic-like small molecules with activity against *Escherichia coli* (*E. coli*) as a case to demonstrate the usage of the two tools.

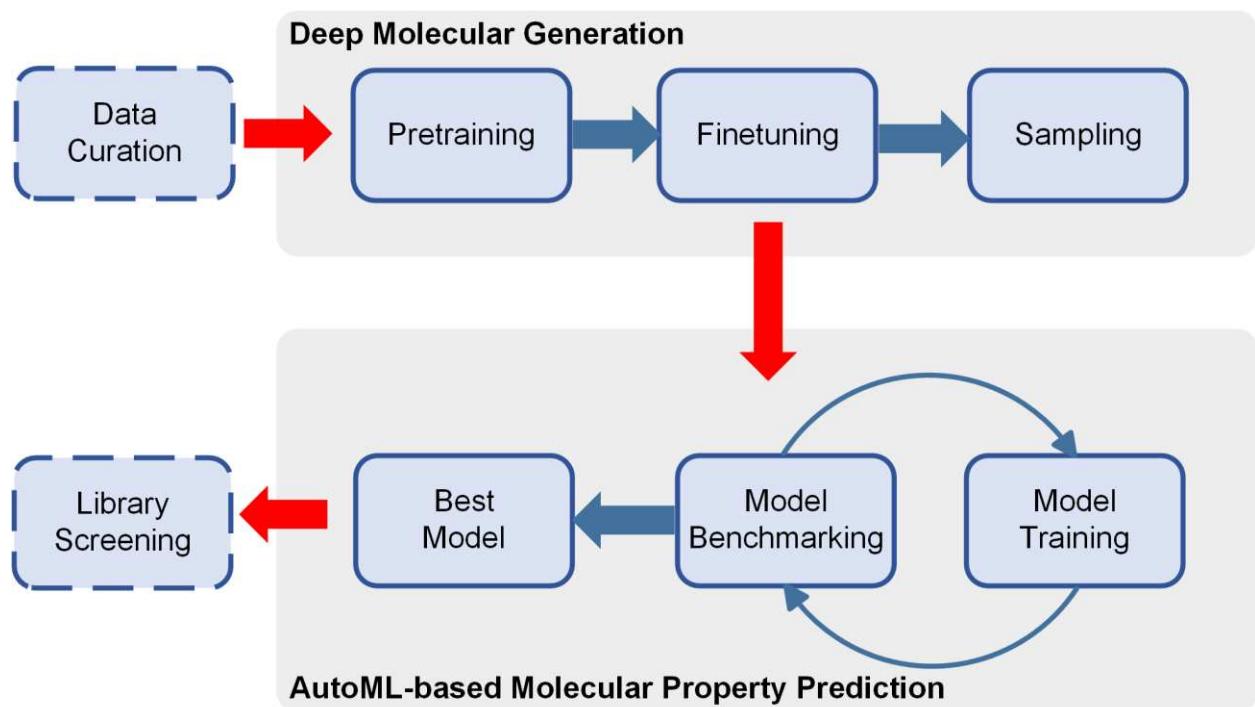


Figure 1. Workflow of automated molecular design implemented in AutoMolDesigner.

Tool I: Generating Focused Library for Small-molecule Antibiotics

Data Curation. Before model training, data curation implemented in the first tab named “Data Preparation” should be conducted first to provide accurate and uniform data for ML model. Apart

from the default curation such as de-duplication, salt removal, and molecular standardization, the other two custom functionalities are provided as well. First, neutralization can be operated on molecules that are used for molecular generation. Ionization states at a certain pH range can be enumerated by Dimorphite-DL²⁴, an open-source Python package for drug-like molecular ionization. This functionality is indispensable for molecular property prediction. Second, molecular data augmentation is available, which performs the simplified molecular input line entry system (SMILES) enumeration. It has been proved effective to enhance model performance for both supervised and unsupervised tasks, especially for molecular property prediction²⁵ and transfer learning in low data regimes²⁶.

We retrieved a large amount of bioactive data from ChEMBL database²⁷ (version 32, <https://www.ebi.ac.uk/chembl/>, accessed March 2023) for model pretraining. The default data curations, molecular neutralization and augmentation by ten folds were conducted to give 7,230,290 data points (“Dataset for Pretraining”). In terms of data used for model finetuning (also named transfer learning), we retrieved active molecules whose measured minimum inhibitory concentration (MIC) values against *E. coli* were no greater than 32 $\mu\text{g}/\text{mL}$ from ChEMBL database (version 32, accessed March 2023). Apart from the same curation as that for the pretraining data, we additionally removed the molecules whose molecular weight is greater than 600 and number of rotatable bonds is more than 20 by referring to the compilation of MUBD-HDACs²⁸. A total of 13,243 data points were obtained and their distribution can be visualized from Figure S2. The MIC value of 1 $\mu\text{g}/\text{mL}$ (-log₂^{MIC} value of 0) was used as the cutoff to discriminate between the compounds with “moderate activity” and those with “high activity”. The molecules labeled as “high activity” were later used to train the molecular property prediction model. The molecules labeled as “moderate activity” accounted for about two thirds of all compounds. The

structural clustering of all the 13,243 molecules based on Morgan2 (Extended-Connectivity Fingerprints with radius 2, ECFP_4) fingerprints²⁹ was carried out, resulting in a total of 570 clusters. Table S2 lists the cluster attribute of each molecule. Moreover, a set of molecules constituted by the compounds with the lowest MIC values from 6 largest clusters along with the compounds from 6 smallest clusters (size = 1) are displayed in Figure S3. As expected, 6 compounds from those large clusters all belong to well-studied chemotypes of small-molecule antibiotics including quinolone, β -lactam, thiazolidinone, propargyl alcohol and diacetylene. In comparison, the other 6 compounds from the smaller clusters have novel scaffolds but show less potency. The above data indicate the dataset is diverse. It not only covers the well-known chemotypes, but also includes rare but structurally novel antibiotics. It should be noted that only the active molecule with the lowest MIC value from each cluster was used for transfer learning. Thus, the diverse subset (“Dataset for Finetuning”) was composed of 570 compounds.

Molecular Generation. By referring to several recently reported prospective campaigns that made full use of deep molecular generation^{30,31}, we designed the main interface named molecular generation (MG) based on an autoregressive model that took a remastered RNN as its core network. Three sub-modules that executed model pretraining, model finetuning and molecular sampling constituted the whole workflow of molecular generation, and they were implemented in three independent tabs, i.e., “MG Pretrain”, “MG Finetune” and “MG Sample”.

Here, we demonstrate the molecular generation beginning with “MG Pretrain”. This submodule aims at learning generalized features of drug-like molecules to enable the unbiased production of stochastic molecules with possible bioactivity. Despite multiple hyperparameter options are available at this tab, only advanced users are recommended to adjust the architecture. Users who focus on practice are suggested to use the generalized model parameters that we have

provided at GitHub. The model was pretrained with the “Dataset for Pretraining”, and its architecture is described in the section of “Materials and Methods, Deep Molecular Generation” in Supporting Information. To follow the routine of the scientific community, we performed benchmarking of our MG model with MOSES³². Table S1 shows that our remastered RNN was superior to CharRNN but fell behind MolGPT³³ in molecular validity. It generated more novel molecules and achieved lower Frechet ChemNet Distance (FCD) score compared with MolGPT. Lower value of FCD score indicates the better capability of model in learning dataset statistics. Notably, MolGPT contained far more network parameters than RNN as its basic architecture was generative-pretraining Transformers, which is computationally expensive and basically trained on high-performance servers. Our model performance was close with these two models in terms of other metrics, indicating the robustness of our remastered MG model implementation.

Following the model pretraining, it came to the transfer learning that plays a key role in practice of molecular generation. Deep learning model attempts to learn from the user-customized data that has shared features (for example, with bioactivity against a specific target, protein family or even phenotype). Ideally, the direction of molecular generation will be steered towards the desired chemical space. It is worth noting that number of iteration (epoch) could be customized in this tab but early stopping technique was also provided to prevent overfitting, thus model training may stop before the preset epoch is met. Considering that model finetuning is the most demanding task on computing capacity in practice, we compared runtime performance on various device configurations using exactly the same parameters and finetuning dataset (“Dataset for Finetuning”). Table S3 shows that it took only 222 seconds to complete this task on a Windows laptop with a common central processing unit (CPU), and 35 seconds more time on an Apple computer, indicating the acceptable running speed of our software. The high-performance server only ran for

7 seconds to complete the task, indicating the introduction of GPU could lead to obvious improvement. The parameters of the finetuned model are also available at GitHub.

At last, “MG Sample” submodule was developed to sample new molecules from the pretrained or finetuned model. The “Temperature Sampling” function is borrowed from other CLMs to improve the sampling process³¹. More molecules with unusual atom types and novel scaffolds will be generated at lower temperature while invalid molecules may also be generated more frequently. Since not every SMILES sampled from the trained models can be converted to valid molecules, the “MG Post-process” module is provided here to measure quality of the generated molecules and perform data curation. The metrics³² including “Validity” (fraction of valid molecules in all generated molecules) and “Uniqueness” (fraction of non-duplicated molecules in all valid molecules) will be shown in a message box after the removal of invalid SMILES and the duplicates. The metric of “Novelty” will also be shown by calculating the number of unique SMILES after de-duplication from the “Dataset for Finetuning” (Figure S4). By default, the “MG Sample” submodule sampled 30,000 molecules by using the trained model.

Library Analysis. To analyze the focused library generated for antibiotics against *E. coli.*, we used the “MG Sample” submodule to sample 1,000 molecules and then used uniform manifold approximation and projection³⁴ (UMAP) to visualize the chemical space characterized by Functional-Class Fingerprints with radius 3 (FCFP_6). The chemical space covered 967 valid molecules sampled from the pretrained model, 910 valid and de-duplicated molecules sampled from the finetuned model and 570 molecules from “Dataset for Finetuning”. Figure S5 shows that most molecules sampled from the finetuned model were adjacent to those known antibiotics in chemical space, proving that the model was able to capture the antibiotic-specific features through transfer learning. Notably, a few molecules generated by finetuning occupied the space of the

molecules generated by pretraining but were not that far from the space of known antibiotics, thus promising small-molecule antibiotics with novel chemotypes may be discovered from these molecules.

Tool II: AutoML-based Molecular Property Prediction for Small-molecule Antibiotics

Although Tool I could generate antibiotic-like molecules as a focused library, it is still necessary to narrow down the chemical space with ML-based molecular property prediction. Herein, we applied AutoML for our purpose³⁵. Such a discipline aims at building task-oriented ML models and further maximizing overall performance with ensemble learning methods including bagging, boosting and stacking. Its main advantages include minimal human intervention, highly parallel model training and optimization as well as universal applicability. Among the AutoML frameworks, AutoGluon is an open-source application developed and distributed by Amazon Co. Ltd., which has achieved state-of-the-art performance on the tabular data benchmarks²³. More importantly, this tool was recently proved applicable to predict pharmacokinetics-related molecular properties³⁶. Accordingly, AutoGluon combined with molecular descriptors/fingerprints was used here to build antibiotic-related molecular property prediction models. As shown in Figure S1B and S6, the “AutoGluon” sub-interface contains two modules, i.e., “Model Training” and “Model Prediction”.

Model Training. The “Model Training” module provides five kinds of molecular descriptors/fingerprints, i.e., RDKit 2D¹⁴, normalized RDKit 2D, ECFP_4, FCFP_6 and MACCS structural keys. The other adjustable arguments are derived from AutoGluon. The “Model Quality Preset” argument determines model performance controlling number of trained models and hyperparameter configurations. This “Model Quality Preset” has four levels ranging from low

level to high level, i.e., “medium_quality”, “good_quality”, “high_quality” and “best_quality”. More computational resources are required if the level is higher. “Evaluation Metric” argument defines the metrics for model training, i.e., F1 score and area under receiver operation characteristic curve (AUROC) for classification and mean absolute error (MAE) and root mean squared error (RMSE) for regression. The “Time Limit” argument sets the maximum amount of time for model training. The training will be forced to stop if the elapsed time is beyond the limit. The “Deployment optimization” argument is used to reduce computational cost by the removal of redundant models but may slightly reduce model performance.

Tabular data is accepted for model training. Its first column is headed with “SMILES” containing molecules represented by SMILES. The second column is headed with “label” containing either 0/1 for binary classification or continuous values for regression. Raw data should be firstly prepared by “Data preparation” module, where Comma-Separated Values (*.csv) and Microsoft Excel (*.xlsx) are two supported formats for input. By default, the whole data are used for training while random splitting of the datasets into a training set and a test set is also available. All the trained ML models are saved under the directory named “ag_models”.

Model Prediction. There are two modes in “Model Prediction” module. For retrospective prediction, users can provide tabular data in the same format as the training data. The model performance is measured by confusion matrix, accuracy, AUROC, Matthews correlation coefficient (MCC) and F1 score for classification and MAE, RMSE, R square (R^2) and median absolute error (MedianAE). For prospective prediction, a plain text file containing molecules represented by SMILES (*.smi, one sequence per row) is required. Both a csv file and a Structure Data File (*.sdf) are used to save the predicted values. To demonstrate the functionality of the

AutoGluon-based ML tools, three cases for predicting antibacterial activity, cytotoxicity and PPB are discussed below.

Antibacterial Activity. The aforementioned studies have proven the feasibility of discovering new antibiotics using bacterial phenotype-based ML models. In this case, 4,774 known antibiotics with “high activity” (MIC: less than 1 μ g/mL) against *E. coli*. were used for modeling. They were firstly used to make benchmarking sets by MUBD-DecoyMaker2.0³⁷, a publicly accessible GUI application focused on making trustworthy datasets for virtual screening. The benchmark for model training was comprised of 501 positive data points (unbiased ligands) and 19,539 negative data points (unbiased decoys), which simulated the virtual screening scenario in real world (i.e., a low hit rate in HTS, Table S4). Since Chemprop is an established ML tool for antibacterial activity prediction¹⁴, we carried out comparative studies between this tool and AutoGluon. Table S5 shows that the best performance was achieved by Morgan2-based AutoGluon model, with the AUC of 0.937 and MCC of 0.687, which significantly outperformed the best Chemprop model, with the AUC of 0.877 and MCC of 0.552. Notably, the best Chemprop model was enhanced by ensemble learning and molecular-level feature¹⁴ (normalized RDKit 2D). In comparison, we set the “Time limit” argument of the best AutoGluon model to 3,600 seconds, indicating that the model performance could be further improved if extra time is allocated. In terms of runtime, it took 27 seconds to train RDKit 2D-based AutoGluon model with “medium quality” but the model’s performance had already been comparable to the best Chemprop model that cost 3,201 seconds for training. The best AutoGluon model that required the most computational resources only took 17 seconds to perform prediction. In comparison, the best Chemprop model cost 121 seconds for prediction.

Cytotoxicity. It is an unwanted property for antibiotic discovery, and it has become a key factor leading to the failed clinical trials of antibiotics³⁸. Considering that some toxicity-related benchmarking sets are not only obsolescent for practical use but also has limited correlation to pharmacological toxicology³⁹, we chose to retrieve training data from a recently published platform named DeepCancerMap⁴⁰. It contains a subset comprised of 10,861 binary inhibitory labels on 32 normal cell lines from 9,668 compounds. With the same dataset, it become natural to compare model performance between AutoGluon and FP-GNN used in DeepCancerMap. Table S6 shows that for AutoGluon model training, RDKit 2D descriptor was slightly superior to Morgan2 fingerprint, with the MCC value of 0.754 versus 0.752. The best AutoGluon model significantly outperformed non-ensembled FP-GNN for which the MCC was only 0.536. However, we noticed that the AUC value of FP-GNN was greater than the highest AUC value achieved by AutoGluon, with the value of 0.914 versus 0.866.

Plasma Protein Binding. It is widely acknowledged that for small-molecule antibiotics, only free fraction of molecules can exert antibacterial effect, thus PPB can affect *in vivo* efficacy of antibiotics⁴¹. Therefore, accurate prediction of PPB can improve efficiency of screening candidates with appropriate binding affinity for plasma protein. In this case, a curated PPB dataset with 3,921 data points was retrieved from a recent study that reported IDL-PPBopt⁴², a deep learning model with graph attention mechanism for regressive prediction on PPB fraction value of compounds. AutoGluon was trained on this dataset and compared with IDL-PPBopt. Table S7 shows that RDKit 2D-based AutoGluon model achieved the best performance, with the MAE value of 0.062, RMSE value of 0.105 and R^2 value of 0.861, which was superior to IDL-PPBopt model for which MAE was 0.075, RMSE was 0.112 and R^2 was 0.841. However, it should be noted that ensemble

learning was basically included for model training in AutoGluon while IDL-PPBopt was not ensembled.

Practical Application of Automated Molecular Design to Small-molecule Antibiotics. This section aims at simulating the application scenario of two tools in real-world drug discovery. 10,000 molecules sampled from the finetuned model were protonated between pH 7.3 and pH 7.5 to give 36,364 data points, which constituted the virtual library. The best configuration for each molecular property prediction model was adopted to train a prospective model with the whole dataset. Table S8 shows the performance of each model on the validation set. Then the models were applied to screening the virtual library. Figure S7 displays 9 potential antibacterial compounds for *E. coli*. with high probability of being active, and their cytotoxicity probability and PPB fraction are also annotated. All these compounds are not indexed in CAS SciFinder⁴³ (accessed September 2023). 9,598 molecules with their predicted values are listed in Table S9, among which 42 molecules were predicted active (activity probability > 0.5), non-toxic (cytotoxicity probability < 0.5) and having appropriate PPB (predicted PPB fraction < 0.9), thus represented promising small-molecule antibiotics. Additionally, all the above predictions were performed on various device configurations to compare the runtime. Table S10 shows that even the most time-consuming task, i.e., PPB prediction, only cost about half an hour on an Apple computer, indicating that it is feasible to leverage these models for large-scale libraries on personal computer. Notably, the exploitation of GPU may not significantly accelerate the prediction. We found that most of the models trained with AutoGluon were CPU-intensive. To facilitate visualization of molecules, we designed a pop-up interface named “Molecular Visualization” (cf. Figure S8), where users can save either high-resolution bitmap or structure file of molecules.

IMPLEMENTATION

AutoMolDesigner is an open-source software written in Python3. Freely accessible Qt Designer software was used to develop its GUI. The functionalities were realized by invoking application programming interfaces of PySide6. Another Python library named PyInstaller (version 5.12.0) was used to pack executable versions for both MacOS and Windows platform. The RNN based on PyTorch (version 1.13.1) was used for language modeling to implement deep molecular generation. The AutoML framework named AutoGluon (version 0.8.2) was adopted for molecular property prediction. Scikit-learn Python package (version 1.2.2) was used to calculate metrics for classification and regression⁴⁴. A Python package named Descriptastorus (version 2.6.1) was used to compute all the molecular descriptors. RDKit⁴⁵ Python package (version 2023.3.3) was used to perform basic molecular operations and visualization while MolVS (version 0.1.1) and Dimorphite-DL (version 1.3.2) was used for salt removal, molecular standardization, charge neutralization or protonation at a certain pH range.

CONCLUSIONS

Combating antibiotic resistance somewhat relies on the discovery of small-molecule drugs with novel chemotypes. However, few user-friendly tools are publicly available for molecular design of small-molecule antibiotics. In this study, we presented an easy-to-use software dedicated to the AI-driven automated design of novel small-molecule antibiotics. We proposed an automated workflow that combined deep molecular generation with AutoML-based molecular property prediction. The benchmarking studies have proved that our tool is either superior or comparable to the other established models. The practical application has further shown that this software can produce novel molecules with high probability of being active, low probability of being toxic and appropriate PPB fraction. Notably, our software not only focuses on designing small molecules with whole cell-based antibacterial activity but also takes high antibiotic-likeness into

consideration. In contrast, target-based computational methods, which leverage either experimentally-determined or computationally-modeled structures, have limited applicability in antibiotic discovery⁴⁶. Ease of use is the other main feature of our software. It is equipped with an intuitive GUI that has been tested on mainstream operating systems. We have shown that users can have acceptable experience by using personal computers with limited computational capacity. Moreover, the curated datasets, the parameters of those trained models and user manual are made publicly available. We expect its wide application in novel small-molecule antibiotic discovery.

Currently, we have established the workflow for designing and screening small-molecule antibiotics. According to the aforementioned demonstration, this workflow could also be applied to other ESKAPE pathogens and other antibiotic-like properties. Our future work will be expanding the curated datasets and trained models for small-molecule antibiotics discovery and developing more useful modules of the software.

ASSOCIATED CONTENT

Data and Software Availability

The CLI version along with its source code is available at the open-source GitHub repository (<https://github.com/taoshen99/AutoMolDesigner>). The packaged GUI versions with user manual is free of charge at Zenodo (<https://zenodo.org/record/8366085>), where all the parameters of the trained models and the curated datasets that can be used for prospective study are also provided.

Supporting Information

Materials and Methods, Software interface figures, data visualization figures, benchmarking results tables (Supporting_Information.docx); datasets made in model training and prediction results (Supplementary_Table.xlsx).

AUTHOR INFORMATION

Corresponding Authors

*E-mail: J.X. (jie.william.xia@hotmail.com), S.W. (ws@imm.ac.cn), D.W. (wangdmchina@imm.ac.cn)

Address: Jie Xia, Song Wu and Dongmei Wang, Institute of Materia Medica, Chinese Academy of Medical Sciences and Peking Union Medical College, Beijing 100050, China

Authors

Address: Tao Shen, Jiale Guo, Zunsheng Han, Gao Zhang and Qingxin Liu, Institute of Materia Medica, Chinese Academy of Medical Sciences and Peking Union Medical College, Beijing

100050, China; Qingxin Liu and Xinxin Si, School of Pharmacy, Jiangsu Ocean University, Lianyungang, Jiangsu 222005, China

ORCID

Jie Xia: 0000-0002-9567-3763

Tao Shen: 0009-0005-7086-8803

Jiale Guo: 0009-0002-6580-2584

Zunsheng Han: 0009-0001-0852-3135

Author Contributions

J.X. led the project, with the support from D.W. and S.W.. T.S. developed the source code and GUI. J.G. reviewed the code and made improvements. T.S., J.G. and Z.H. performed data collection and curation, and built ML models. T.S. and J.X. jointly performed the data analysis. J.G., Z.H., G.Z., Q.L. and X.S. tested the software and provided invaluable feedbacks. T.S. and J.X. wrote the manuscript. All authors reviewed the final version of the manuscript and provided important revisions.

Notes

The authors declare no competing financial interest.

Acknowledgements

This study was funded by the CAMS Innovation Fund for Medical Sciences (Grant No. 2021-I2M-1-069), the National Science and Technology Major Projects for Major New Drugs Innovation and Development (Grant No. 2018ZX09711001-012-003) and the Program for Foreign

Talent of Ministry of Science and Technology of the People's Republic of China (Grant No. G2021194015L).

REFERENCES

- (1) Tommasi, R.; Brown, D. G.; Walkup, G. K.; Manchester, J. I.; Miller, A. A., ESKAPEing the labyrinth of antibacterial discovery. *Nat. Rev. Drug Discov.* **2015**, *14* (8), 529-42.
- (2) Oliveira, D. M. P. D.; Forde, B. M.; Kidd, T. J.; Harris, P. N. A.; Schembri, M. A.; Beatson, S. A.; Paterson, D. L.; Walker, M. J., Antimicrobial Resistance in ESKAPE Pathogens. *Clin. Microbiol. Rev.* **2020**, *33* (3), 10.1128/cmr.00181-19.
- (3) Cook, M. A.; Wright, G. D., The past, present, and future of antibiotics. *Sci. Transl. Med.* **2022**, *14* (657), eabo7793.
- (4) Wright, G. D., Opportunities for natural products in 21st century antibiotic discovery. *Nat. Prod. Rep.* **2017**, *34* (7), 694-701.
- (5) Cox, G.; Sieron, A.; King, A. M.; De Pascale, G.; Pawlowski, A. C.; Koteva, K.; Wright, G. D., A Common Platform for Antibiotic Dereplication and Adjuvant Discovery. *Cell Chem. Biol.* **2017**, *24* (1), 98-109.
- (6) Ford, W. C.; Zurenko, E. G.; Barbachyn, R. M., The Discovery of Linezolid, the First Oxazolidinone Antibacterial Agent. *Curr. Drug Targets Infect. Disord.* **2001**, *1* (2), 181-99.
- (7) Humphries, R. M.; Pollett, S.; Sakoulas, G., A Current Perspective on Daptomycin for the Clinical Microbiologist. *Clin. Microbiol. Rev.* **2013**, *26* (4), 759-80.
- (8) Daum, R. S.; Kar, S.; Kirkpatrick, P., Retapamulin. *Nat. Rev. Drug Discov.* **2007**, *6* (11), 865-6.
- (9) Grzelak, E. M.; Choules, M. P.; Gao, W.; Cai, G.; Wan, B.; Wang, Y.; McAlpine, J. B.; Cheng, J.; Jin, Y.; Lee, H.; et al., Strategies in anti-Mycobacterium tuberculosis drug discovery based on phenotypic screening. *J. Antibiot.* **2019**, *72* (10), 719-28.
- (10) Melo, M. C. R.; Maasch, J. R. M. A.; de la Fuente-Nunez, C., Accelerating antibiotic discovery through artificial intelligence. *Commun. Biol.* **2021**, *4* (1), 1050.
- (11) Tropsha, A., Best Practices for QSAR Model Development, Validation, and Exploitation. *Mol. Inform.* **2010**, *29* (6-7), 476-88.
- (12) Wang, L.; Le, X.; Li, L.; Ju, Y.; Lin, Z.; Gu, Q.; Xu, J., Discovering New Agents Active against Methicillin-Resistant *Staphylococcus aureus* with Ligand-Based Approaches. *J. Chem. Inf. Model.* **2014**, *54* (11), 3186-97.
- (13) Liu, G.; Stokes, J. M., A brief guide to machine learning for antibiotic discovery. *Curr. Opin. Microbiol.* **2022**, *69*, 102190.
- (14) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; et al., Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model.* **2019**, *59* (8), 3370-88.
- (15) Stokes, J. M.; Yang, K.; Swanson, K.; Jin, W.; Cubillos-Ruiz, A.; Donghia, N. M.; MacNair, C. R.; French, S.; Carfrae, L. A.; Bloom-Ackermann, Z.; et al., A Deep Learning Approach to Antibiotic Discovery. *Cell* **2020**, *180* (4), 688-702.
- (16) Liu, G.; Catacutan, D. B.; Rathod, K.; Swanson, K.; Jin, W.; Mohammed, J. C.; Chiappino-Pepe, A.; Syed, S. A.; Fragis, M.; Rachwalski, K.; et al., Deep learning-guided discovery of an antibiotic targeting *Acinetobacter baumannii*. *Nat. Chem. Biol.* **2023**.

(17) Tucs, A.; Tran, D. P.; Yumoto, A.; Ito, Y.; Uzawa, T.; Tsuda, K., Generating Ampicillin-Level Antimicrobial Peptides with Activity-Aware Generative Adversarial Networks. *ACS Omega* **2020**, *5* (36), 22847-51.

(18) Segler, M. H. S.; Kogej, T.; Tyrchan, C.; Waller, M. P., Generating Focused Molecule Libraries for Drug Discovery with Recurrent Neural Networks. *ACS Cent Sci* **2018**, *4* (1), 120-31.

(19) Krishnan, S. R.; Bung, N.; Padhi, S.; Bulusu, G.; Misra, P.; Pal, M.; Oruganti, S.; Srinivasan, R.; Roy, A., De novo design of anti-tuberculosis agents using a structure-based deep learning method. *J. Mol. Graphics Model.* **2023**, *118*, 108361.

(20) Mansbach, R. A.; Leus, I. V.; Mehla, J.; Lopez, C. A.; Walker, J. K.; Rybenkov, V. V.; Hengartner, N. W.; Zgurskaya, H. I.; Gnanakaran, S., Machine Learning Algorithm Identifies an Antibiotic Vocabulary for Permeating Gram-Negative Bacteria. *J. Chem. Inf. Model.* **2020**, *60* (6), 2838-47.

(21) Lai, J.; Li, X.; Wang, Y.; Yin, S.; Zhou, J.; Liu, Z., AI Scaffold: A Web-Based Tool for Scaffold Diversification Using Deep Learning. *J. Chem. Inf. Model.* **2021**, *61* (1), 1-6.

(22) Ivanenkov, Y. A.; Polykovskiy, D.; Bezrukov, D.; Zagribelnyy, B.; Aladinskiy, V.; Kamya, P.; Aliper, A.; Ren, F.; Zhavoronkov, A., Chemistry42: An AI-Driven Platform for Molecular Design and Optimization. *J. Chem. Inf. Model.* **2023**, *63* (3), 695-701.

(23) Erickson, N.; Mueller, J.; Shirkov, A.; Zhang, H.; Larroy, P.; Li, M.; Smola, A., Autogluontabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505* **2020**.

(24) Ropp, P. J.; Kaminsky, J. C.; Yablonski, S.; Durrant, J. D., Dimorphite-DL: an open-source program for enumerating the ionization states of drug-like small molecules. *J. Cheminform.* **2019**, *11* (1), 14.

(25) Karpov, P.; Godin, G.; Tetko, I. V., Transformer-CNN: Swiss knife for QSAR modeling and interpretation. *J. Cheminform.* **2020**, *12* (1), 17.

(26) Moret, M.; Friedrich, L.; Grisoni, F.; Merk, D.; Schneider, G., Generative molecular design in low data regimes. *Nat. Mach. Intell.* **2020**, *2* (3), 171-80.

(27) Mendez, D.; Gaulton, A.; Bento, A. P.; Chambers, J.; De Veij, M.; Felix, E.; Magarinos, M. P.; Mosquera, J. F.; Mutowo, P.; Nowotka, M.; et al., ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Res.* **2019**, *47* (D1), D930-D40.

(28) Xia, J.; Tilahun, E. L.; Kebede, E. H.; Reid, T. E.; Zhang, L.; Wang, X. S., Comparative modeling and benchmarking data sets for human histone deacetylases and sirtuin families. *J. Chem. Inf. Model.* **2015**, *55* (2), 374-88.

(29) Rogers, D.; Hahn, M., Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **2010**, *50* (5), 742-54.

(30) Li, Y. S.; Zhang, L. T.; Wang, Y. F.; Zou, J.; Yang, R.; Luo, X.; Wu, C.; Yang, W.; Tian, C.; Xu, H.; et al., Generative deep learning enables the discovery of a potent and selective RIPK1 inhibitor. *Nat. Commun.* **2022**, *13* (1), 6891.

(31) Moret, M.; Pachon Angona, I.; Cotos, L.; Yan, S.; Atz, K.; Brunner, C.; Baumgartner, M.; Grisoni, F.; Schneider, G., Leveraging molecular structure and bioactivity with chemical language models for de novo drug design. *Nat. Commun.* **2023**, *14* (1), 114.

(32) Polykovskiy, D.; Zhebrak, A.; Sanchez-Lengeling, B.; Golovanov, S.; Tatanov, O.; Belyaev, S.; Kurbanov, R.; Artamonov, A.; Aladinskiy, V.; Veselov, M.; et al., Molecular Sets (MOSES): A Benchmarking Platform for Molecular Generation Models. *Front. Pharmacol.* **2020**, *11*, 565644.

(33) Bagal, V.; Aggarwal, R.; Vinod, P. K.; Priyakumar, U. D., MolGPT: Molecular Generation Using a Transformer-Decoder Model. *J. Chem. Inf. Model.* **2022**, *62* (9), 2064-76.

(34) McInnes, L.; Healy, J.; Melville, J., Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* **2018**.

(35) Vaccaro, L.; Sansonetti, G.; Micarelli, A., An Empirical Review of Automated Machine Learning. *Computers* **2021**, *10* (1), 11.

(36) Stoyanova, R.; Katzberger, P. M.; Komissarov, L.; Khadhraoui, A.; Sach-Peltason, L.; Groebke Zbinden, K.; Schindler, T.; Manevski, N., Computational Predictions of Nonclinical Pharmacokinetics at the Drug Design Stage. *J. Chem. Inf. Model.* **2023**, *63* (2), 442-58.

(37) Xia, J.; Li, S.; Ding, Y.; Wu, S.; Wang, X. S., MUBD-DecoyMaker 2.0: A Python GUI Application to Generate Maximal Unbiased Benchmarking Data Sets for Virtual Drug Screening. *Mol. Inform.* **2020**, *39* (4), e1900151.

(38) Cohen, F.; Aggen, J. B.; Andrews, L. D.; Assar, Z.; Boggs, J.; Choi, T.; Dozzo, P.; Easterday, A. N.; Haglund, C. M.; Hildebrandt, D. J.; et al., Optimization of LpxC Inhibitors for Antibacterial Activity and Cardiovascular Safety. *ChemMedChem* **2019**, *14* (16), 1560-72.

(39) Richard, A. M.; Huang, R.; Waidyanatha, S.; Shinn, P.; Collins, B. J.; Thillainadarajah, I.; Grulke, C. M.; Williams, A. J.; Lougee, R. R.; Judson, R. S.; et al., The Tox21 10K Compound Library: Collaborative Chemistry Advancing Toxicology. *Chem. Res. Toxicol.* **2021**, *34* (2), 189-216.

(40) Wu, J.; Xiao, Y.; Lin, M.; Cai, H.; Zhao, D.; Li, Y.; Luo, H.; Tang, C.; Wang, L., DeepCancerMap: A versatile deep learning platform for target- and cell-based anticancer drug discovery. *Eur. J. Med. Chem.* **2023**, *255*, 115401.

(41) Schmidt, S.; Röck, K.; Sahre, M.; Burkhardt, O.; Brunner, M.; Lobmeyer, M. T.; Derendorf, H., Effect of Protein Binding on the Pharmacological Activity of Highly Bound Antibiotics. *Antimicrob. Agents Chemother.* **2008**, *52* (11), 3994-4000.

(42) Lou, C.; Yang, H.; Wang, J.; Huang, M.; Li, W.; Liu, G.; Lee, P. W.; Tang, Y., IDL-PPBopt: A Strategy for Prediction and Optimization of Human Plasma Protein Binding of Compounds via an Interpretable Deep Learning Method. *J. Chem. Inf. Model.* **2022**, *62* (11), 2788-99.

(43) Gabrielson, S. W., SciFinder. *J. Med. Libr. Assoc.* **2018**, *106* (4), 588.

(44) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V., Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825-30.

(45) Landrum, G. *RDKit: Open-Source Cheminformatics Software*. RDKit, 2023. <https://www.rdkit.org/> (accessed on 29 March 2023).

(46) Wong, F.; Krishnan, A.; Zheng, E. J.; Stärk, H.; Manson, A. L.; Earl, A. M.; Jaakkola, T.; Collins, J. J., Benchmarking AlphaFold-enabled molecular docking predictions for antibiotic discovery. *Mol. Syst. Biol.* **2022**, *18* (9), e11081.

Supporting Information

AutoMolDesigner for Antibiotic Discovery: An AI-based Open-source Software for Automated Design of Small-molecule Antibiotics

Tao Shen[¶], Jiale Guo[¶], Zunsheng Han[¶], Gao Zhang[¶], Qingxin Liu^{¶[○]}, Xinxin Si[○], Dongmei Wang^{¶}, Song Wu^{¶*}, Jie Xia^{¶*}*

[¶]State Key Laboratory of Bioactive Substance and Function of Natural Medicines, Institute of Materia Medica, Chinese Academy of Medical Sciences and Peking Union Medical College, Beijing 100050, China; [○]School of Pharmacy, Jiangsu Ocean University, Lianyungang, Jiangsu 222005, China; *corresponding authors.

Correspondence should be addressed to J.X. (jie.william.xia@hotmail.com)

Dr. Jie Xia

Institute of Materia Medica,

Chinese Academy of Medical Sciences

No. 2 Nanwei Road

Beijing 100050, China

Table of Contents

| | |
|--|--------------------------|
| Figure S1. Two functional windows of AutoMolDesigner. (A) Main window: deep molecular generation for <i>de novo</i> drug design; (B) Pop-up window: automated machine learning for molecular property prediction. | S4 |
| Figure S2. Distribution of antibacterial activity (minimum inhibitory concentration, MIC, $\mu\text{g/mL}$). The value was transformed as $-\log_2^{\text{MIC}}$. | S4 |
| Figure S3. Compounds annotated with MIC, cluster index and cluster size from 12 selected clusters. The cluster index can be referred to Table S2. | S5 |
| Figure S4. “Validity”, “Uniqueness” and “Novelty” of 1,000 molecules sampled from the finetuned model at “MG Post-process” tab. | S6 |
| Figure S5. UMAP visualization of molecules embedded in chemical space characterized by FCFP_6. 967 molecules sampled from the pretrained model are colored blue, 910 molecules sampled from the finetuned model are colored orange and 570 molecules from “Datasets for Finetuning” are colored dark red. | S6 |
| Figure S6. “Model Prediction” interface of the AutoGluon module. | S7 |
| Figure S7. 9 designed antibacterial compounds with high active probability, alongside the cytotoxicity probability and PPB fraction. | S8 |
| Figure S8. “Molecular Visualization” interface popping up from the main interface. | S9 |
| Table S1. Performance comparison between our remastered RNN from AutoMolDesigner and other deep molecular generation models, i.e., CharRNN and MolGPT, with MOSES as the benchmark. The best value for each metric is highlighted in bold. | S10 |
| Table S2. Structural clustering of the molecules for finetuning in molecular generation with Morgan2-based Butina algorithm. | Supplementary_Table.xlsx |

Table S3. Runtime of model finetuning with various device configurations. Configuration (A)

Ubuntu server: CPU [Intel(R) Core(TM) i9-12900K @ 3.20GHz] with GPU [NVIDIA GA102 (GeForce RTX 3090 24GB)]; (B) Ubuntu server: CPU [Intel(R) Core(TM) i9-12900K @ 3.20GHz]; (C) Windows laptop: CPU [AMD Ryzen(TM) 7 5800H @ 3.20GHz] with GPU [NVIDIA GA106 (GeForce RTX 3050Ti Mobile 4GB)]; (D) Windows laptop: CPU [AMD Ryzen(TM) 7 5800H @ 3.20GHz]; (E) Apple Mac mini: CPU (Apple M2 @ 3.49GHz). S10

Table S4. MUBD2.0 dataset for antibiotics against *E. coli*. *Supplementary_Table.xlsx*

Table S5. Antibacterial activity prediction – comparative study between Chemprop and AutoGluon. The best value for each metric is highlighted in bold. S11

Table S6. Cytotoxicity prediction – comparative study between FP-GNN and AutoGluon. This benchmark contained 32 tasks, and their results were averaged for comparison. The best value for each metric is highlighted in bold. S12

Table S7. Plasma protein binding prediction – comparative study between IDL-PPBopt and AutoGluon. The best value for each metric is highlighted in bold. S12

Table S8. Validation set performance of three molecular property prediction models applied to screening the virtual library. S13

Table S9. 9,598 compounds with activity probability (*E. coli*), cytotoxicity probability and predicted PPB fraction. Among all ionization states of a molecule, the one with the highest activity probability (*E. coli*) was kept. *Supplementary_Table.xlsx*

Table S10. Runtime of predicting 36,364 data points on three tasks with various device configurations (see definitions in Table S3). S13

Materials and Methods S14

References S20

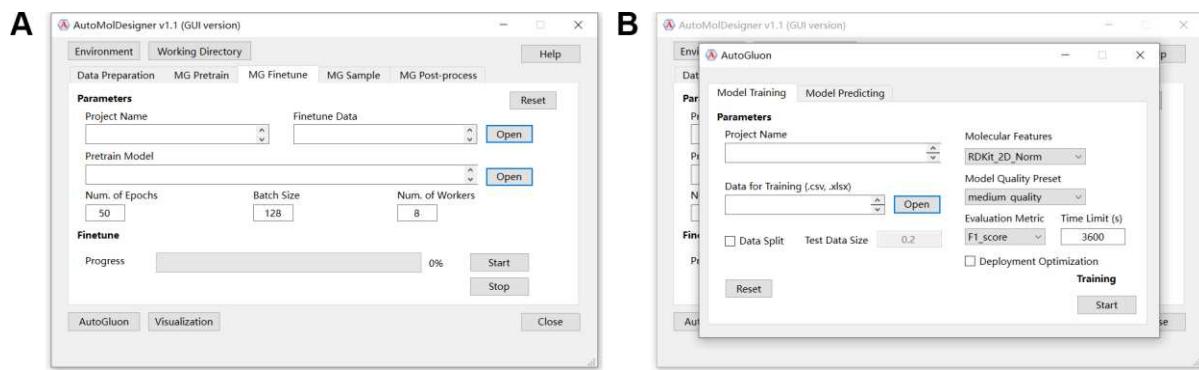


Figure S1. Two functional windows of AutoMolDesigner. **(A)** Main window: deep molecular generation for *de novo* drug design; **(B)** Pop-up window: automated machine learning for molecular property prediction.

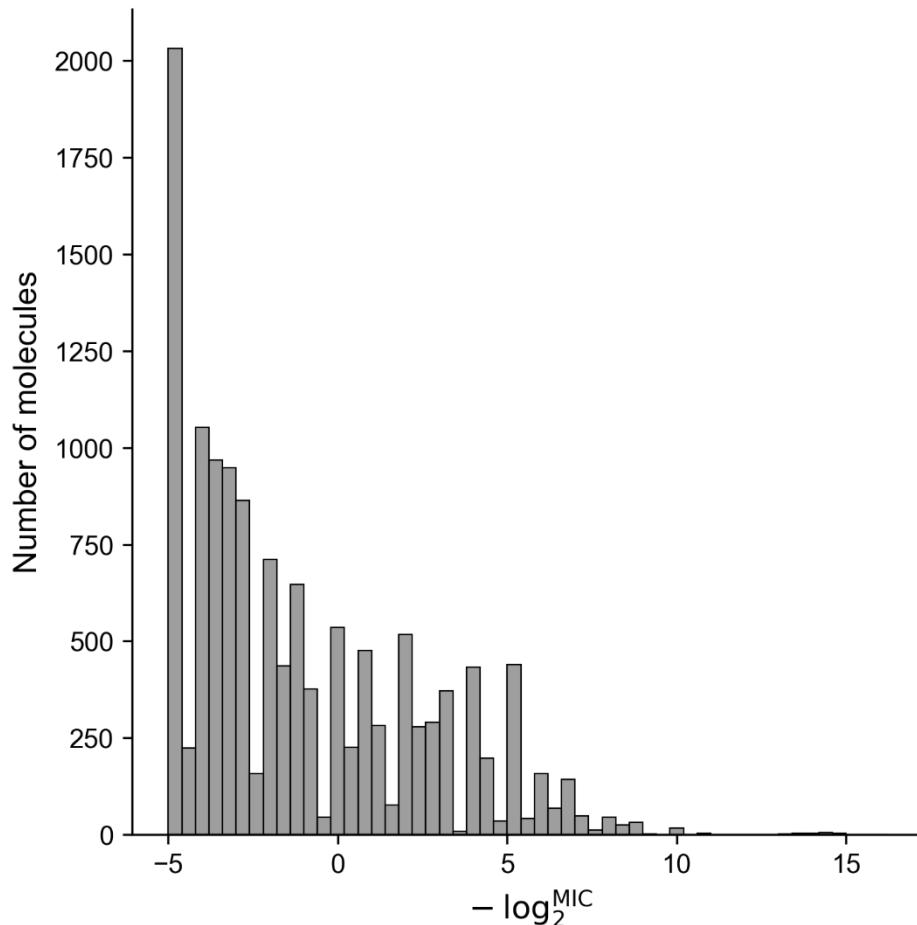


Figure S2. Distribution of antibacterial activity (minimum inhibitory concentration, MIC, $\mu\text{g/mL}$).

The value was transformed as $-\log_2^{\text{MIC}}$.

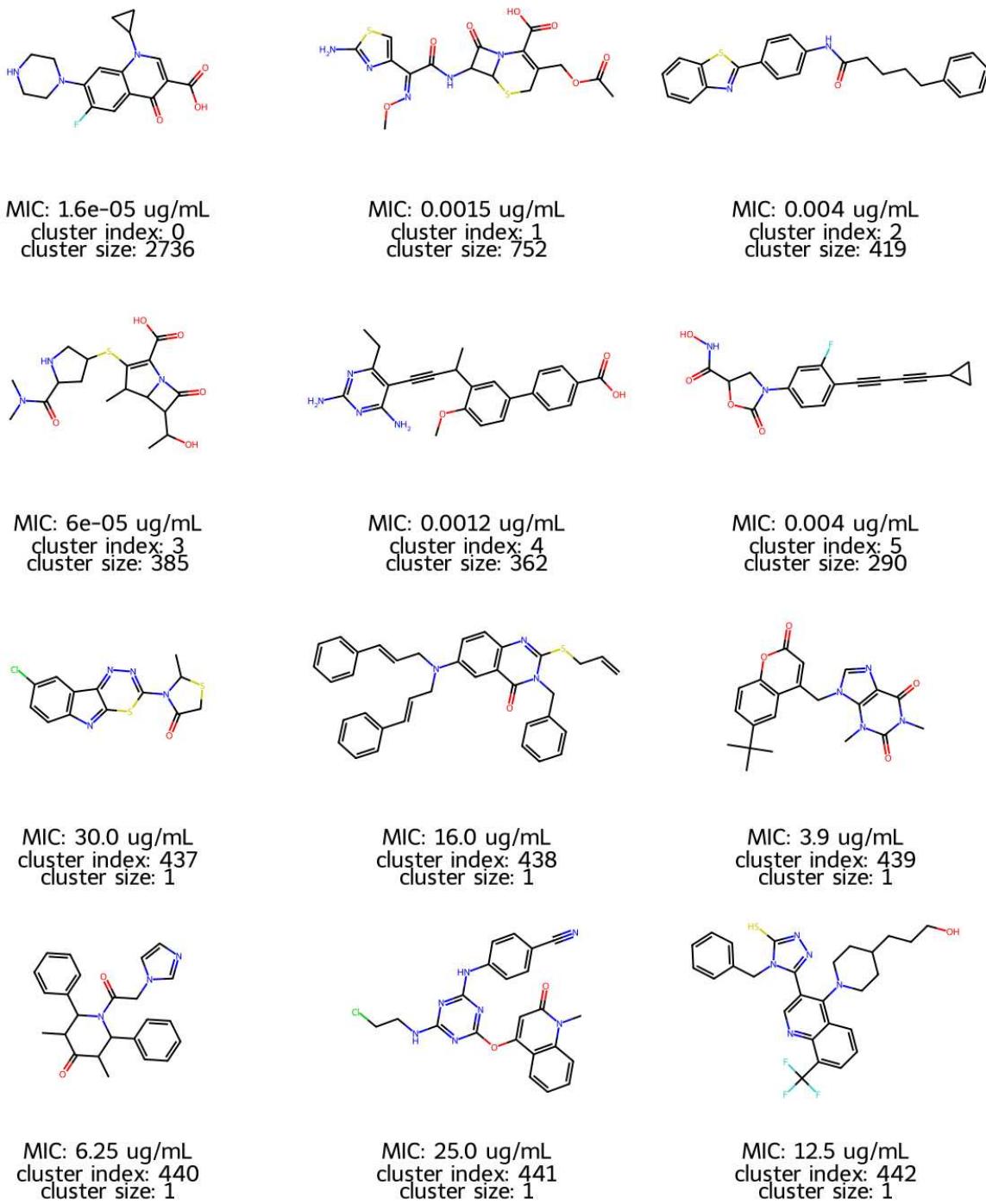


Figure S3. Compounds annotated with MIC, cluster index and cluster size from 12 selected clusters. The cluster index can be referred to Table S2.

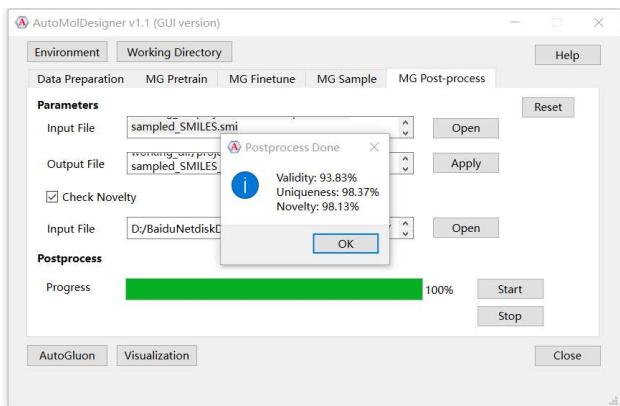


Figure S4. “Validity”, “Uniqueness” and “Novelty” of 1,000 molecules sampled from the finetuned model at “MG Post-process” tab.

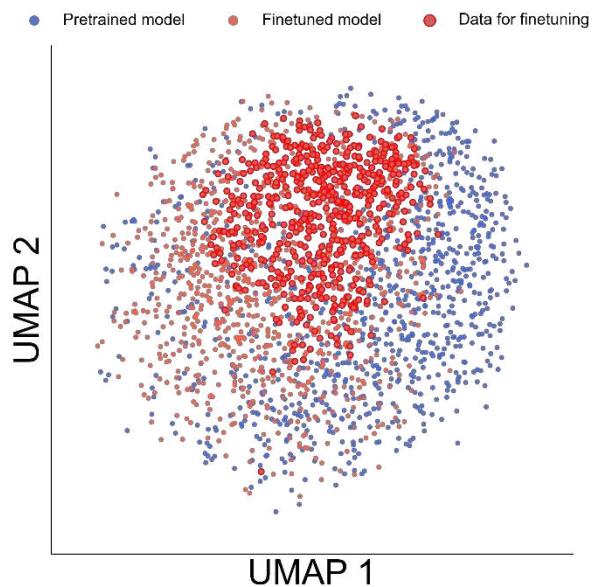


Figure S5. UMAP visualization of molecules embedded in chemical space characterized by FCFP_6. 967 molecules sampled from the pretrained model are colored blue, 910 molecules sampled from the finetuned model are colored orange and 570 molecules from “Datasets for Finetuning” are colored dark red.

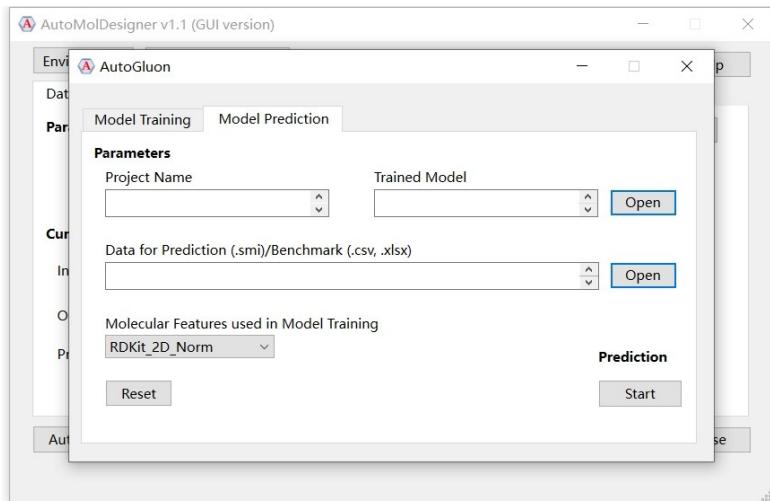


Figure S6. “Model Prediction” interface of the AutoGluon module.

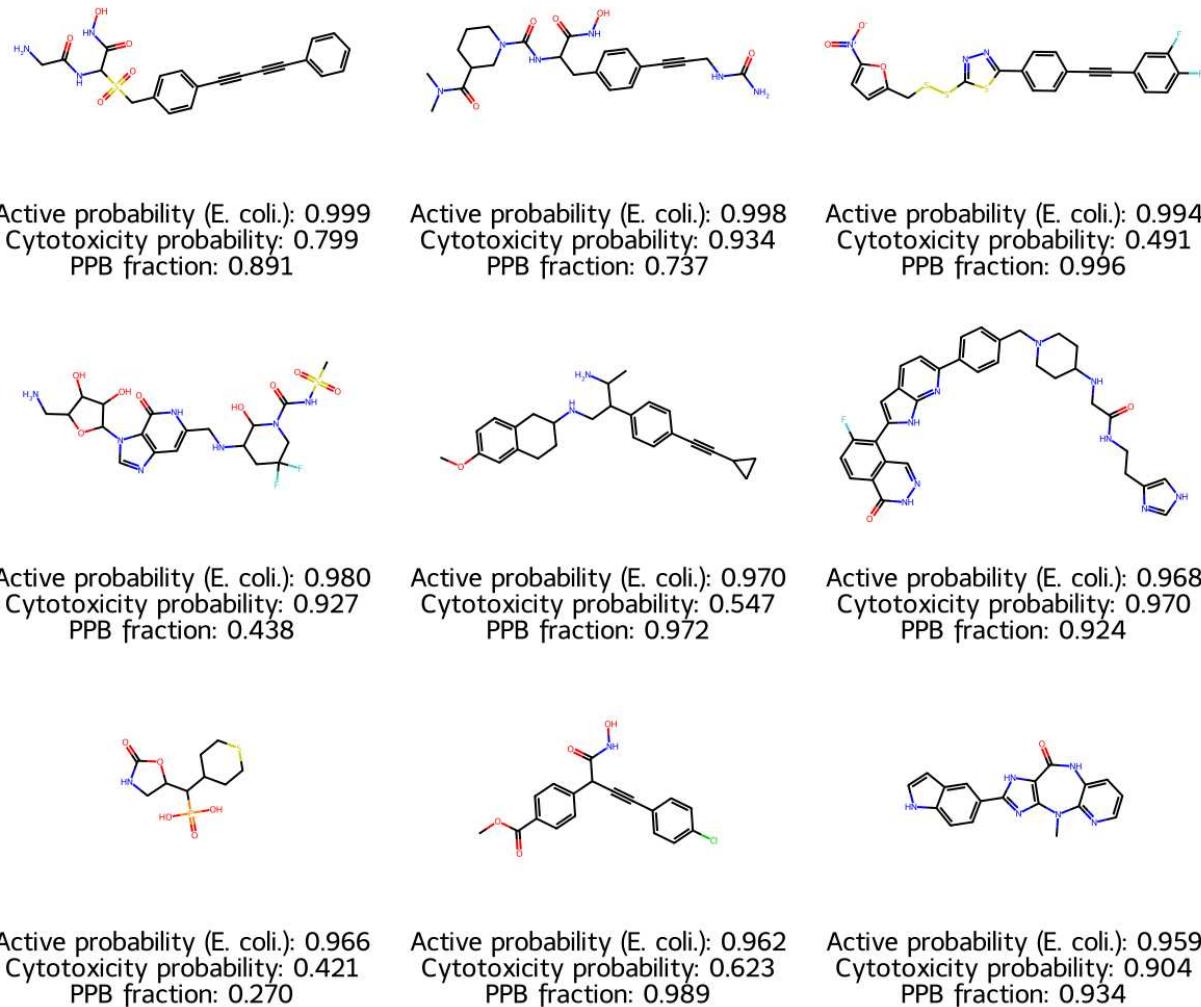


Figure S7. 9 designed antibacterial compounds with high active probability, alongside the cytotoxicity probability and PPB fraction.

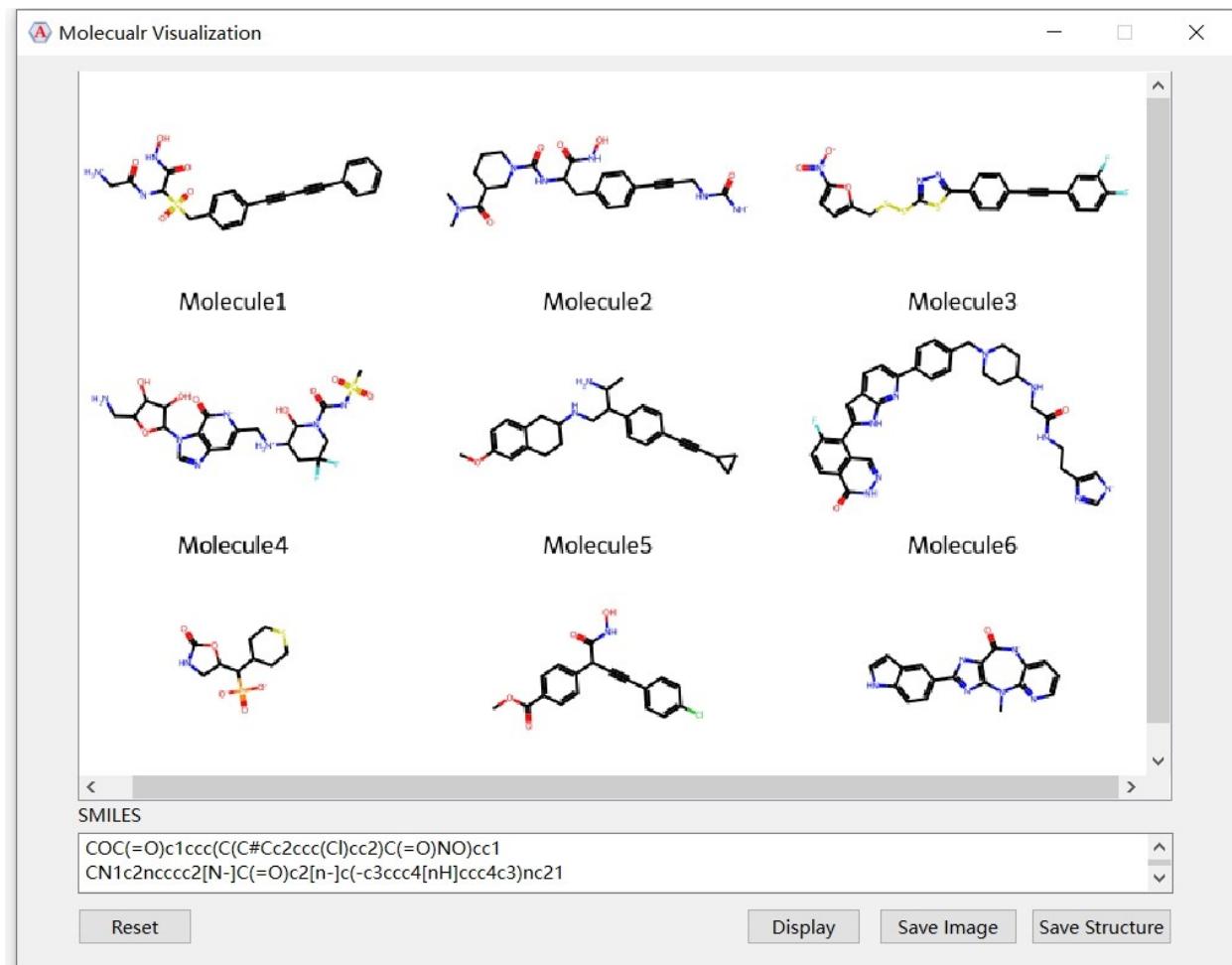


Figure S8. “Molecular Visualization” interface popping up from the main interface.

Table S1. Performance comparison between our remastered RNN from AutoMolDesigner and other deep molecular generation models, i.e., CharRNN and MolGPT, with MOSES as the benchmark. The best value for each metric is highlighted in bold.

| Models | validity | unique @10k | novelty | IntDiv ₁ | IntDiv ₂ | FCD /Test |
|----------------------|--------------|----------------|--------------|---------------------|---------------------|---------------|
| CharRNN ^a | 0.975 | 0.999 | 0.842 | 0.856 | 0.85 | 0.0732 |
| MolGPT ^a | 0.994 | 1.0 | 0.797 | 0.857 | 0.851 | 0.067 |
| AutoMolDesigner | 0.988 | 0.999 | 0.820 | 0.856 | 0.85 | 0.0628 |

^aThe model performance was retrieved from the work that reported MolGPT¹.

Table S3. Runtime of model finetuning with various device configurations. Configuration **(A)** Ubuntu server: CPU [Intel(R) Core(TM) i9-12900K @ 3.20GHz] with GPU [NVIDIA GA102 (GeForce RTX 3090 24GB)]; **(B)** Ubuntu server: CPU [Intel(R) Core(TM) i9-12900K @ 3.20GHz]; **(C)** Windows laptop: CPU [AMD Ryzen(TM) 7 5800H @ 3.20GHz] with GPU [NVIDIA GA106 (GeForce RTX 3050Ti Mobile 4GB)]; **(D)** Windows laptop: CPU [AMD Ryzen(TM) 7 5800H @ 3.20GHz]; **(E)** Apple Mac mini: CPU (Apple M2 @ 3.49GHz).

| Runtime (s) | A | B | C | D | E |
|---------------|---|----|-----|-----|-----|
| One iteration | 1 | 4 | 22 | 26 | 31 |
| Full task | 7 | 34 | 167 | 222 | 257 |

Table S5. Antibacterial activity prediction – comparative study between Chemprop and AutoGluon. The best value for each metric is highlighted in bold.

| Models | ACC ^a | AUC ^b | MCC ^c | F1 score | Training time ^g (s) | Predicting time (s) |
|--------------------------------|------------------|------------------|------------------|--------------|--------------------------------|---------------------|
| Chemprop | | | | | | |
| baseline | 0.974 | 0.852 | 0.443 | 0.456 | 155 | 4 |
| baseline ensemble ^d | 0.979 | 0.873 | 0.524 | 0.533 | 2782 | 48 |
| RDKit 2D feature ^e | 0.980 | 0.864 | 0.520 | 0.521 | 457 | 79 |
| RDKit 2D feature ensemble | 0.981 | 0.877 | 0.552 | 0.557 | 3201 | 121 |
| Morgan2 feature | 0.978 | 0.903 | 0.506 | 0.511 | 161 | 5 |
| Morgan2 feature ensemble | 0.979 | 0.914 | 0.532 | 0.540 | 2925 | 52 |
| AutoGluon | | | | | | |
| RDKit_2D ^f | baseline | 0.983 | 0.879 | 0.572 | 0.555 | 27 |
| | best | 0.981 | 0.930 | 0.567 | 0.573 | 1687 |
| Morgan 2 | baseline | 0.982 | 0.870 | 0.544 | 0.520 | 67 |
| | best | 0.987 | 0.937 | 0.687 | 0.671 | 3365 |
| | | | | | | 17 |

^aACC: Accuracy. ^bAUC: Area under receiver operating characteristic. ^cMCC: Matthews correlation coefficient. ^dThe ensemble consisted of 20 independently trained models.

^eChemprop was enhanced by molecular-level features that are additional molecular descriptors. ^fThis molecular descriptor named normalized RDKit_2D was the same as that used for Chemprop enhancement. ^gFor Chemprop, the training time did not include the time for hyperparameters optimization.

Table S6. Cytotoxicity prediction – comparative study between FP-GNN and AutoGluon. This benchmark contained 32 tasks, and their results were averaged for comparison. The best value for each metric is highlighted in bold.

| Models | Sensitivity | Specificity | ACC ^a | AUC ^b | MCC ^c |
|-----------------------|-------------|--------------|------------------|------------------|------------------|
| FP-GNN | | | | | |
| Reported model | 0.813 | 0.423 | 0.722 | 0.914 | 0.536 |
| AutoGluon | | | | | |
| RDKit_2D ^d | baseline | 0.952 | 0.643 | 0.853 | 0.818 |
| | best | 0.953 | 0.662 | 0.877 | 0.866 |
| Morgan2 | baseline | 0.948 | 0.593 | 0.837 | 0.819 |
| | best | 0.962 | 0.632 | 0.872 | 0.864 |

^aACC: Accuracy. ^bAUC: Area under receiver operating characteristic. ^cMCC: Matthews correlation coefficient. ^dNormalized version.

Table S7. Plasma protein binding prediction – comparative study between IDL-PPBopt and AutoGluon. The best value for each metric is highlighted in bold.

| Models | MAE ^a | RMSE ^b | R ² ^c | Training time (s) | Predicting time (s) |
|-----------------------|------------------|-------------------|-----------------------------|-------------------|---------------------|
| IDL-PPBopt | | | | | |
| Reported model | 0.075 | 0.112 | 0.841 | - ^e | 2 |
| AutoGluon | | | | | |
| RDKit_2D ^d | baseline | 0.068 | 0.110 | 0.846 | 25 |
| | best | 0.062 | 0.105 | 0.861 | 3511 |
| Morgan2 | baseline | 0.084 | 0.134 | 0.772 | 146 |
| | best | 0.074 | 0.124 | 0.806 | 3542 |

^aMAE: Mean absolute error. ^bRMSE: root mean squared error. ^cR²: R square. ^dNormalized version.

^eThe official script for training is not available in the published work.

Table S8. Validation set performance of three molecular property prediction models applied to screening the virtual library.

| Metric ^a | Antibacterial activity | Cytotoxicity | Plasma protein binding |
|---------------------|------------------------|----------------|------------------------|
| F1 score | 0.645 | 0.894 | - ^b |
| MAE | - ^b | - ^b | 0.062 |

^aThe “Evaluation Metric” argument for AutoGluon model training. ^bNot applied.

Table S10. Runtime of predicting 36,364 data points on three tasks with various device configurations (see definitions in Table S3).

| Runtime (s) | A | B | C | D | E |
|------------------------|-----|-----|------|------|------|
| Antibacterial activity | 77 | 80 | 252 | 254 | 106 |
| Cytotoxicity | 167 | 170 | 300 | 313 | 211 |
| Plasma protein binding | 666 | 660 | 1006 | 1029 | 1744 |

MATERIALS AND METHODS

Data Collection and Curation.

In the section of “Tool I: Generating Focused Library for Small-molecule Antibiotics”, all the training data were retrieved from ChEMBL² (version 32, <https://www.ebi.ac.uk/chembl/>, accessed March 2023). For pretraining, 764,373 small molecules (SMILES) whose annotated bioactivities (“K_d”, “K_i”, “K_b”, “IC50”, “EC50”) were no higher than 10 μM were obtained³. After the standard curation including de-duplication, salts removal and molecular standardization, they were further augmented by 10 folds to give 7,230,290 data points (“Dataset for Pretraining”). For finetuning, 15,795 small molecules whose annotated minimum inhibitory concentration (MIC) values were no higher than 32 μg/mL for *Escherichia coli* (*E. coli*) were obtained after “NaN” dropping and de-duplication. Apart from the standard curation, molecules whose molecular weight were greater than 600 or number of rotatable bonds were more than 20^{4,5} were also removed to give 13,243 data points. Structural clustering based on Morgan2 fingerprints using Butina algorithm with 0.75 (Tanimoto similarity) as the threshold was carried out to give 570 clusters⁶. The most active molecule from each cluster was collected to constitute the final finetuning datasets (“Dataset for Finetuning”).

In the section of “Tool II: AutoML-based Molecular Property Prediction for Small-molecule Antibiotics”, MUBD-DecoyMaker2.0⁷ was used to make the dataset used for training and predicting antibacterial activity against *E. coli*. Briefly, molecules labeled with “high activity” (MIC ≤ 1 μg/mL) from 13,243 data points were protonated at pH ranging from 7.3 to 7.5 by Discovery Studio⁸ (version 2016) to give 5,021 data points, which served as the input for MUBD-DecoyMaker2.0. The final MUBD dataset was comprised of 501 unbiased ligands and 19,539 unbiased decoys. The dataset used for cytotoxicity prediction was retrieved from a subset in

DeepCancerMap⁹. This server contains 32 subsets comprised of compounds with annotated binary toxicity labels against normal cell lines. We followed its train-test splitting way to train 32 individual models. The dataset used for plasma protein binding (PPB) prediction was basically retrieved from IDL-PPBopt¹⁰, which contains 3,921 records. Its published splits of training, validating and testing sets were strictly followed.

There were two sets of molecules generated by the pretrained and finetuned model, respectively. One set contained 1,000 data points that were used for chemical space visualization. The other set of molecules including 10,000 data points was generated by the finetuned model for the simulated experiment of virtual screening. All the post-processing procedures implemented in “MG Post-process” tab were performed on these sets of molecules to remove invalid and duplicate molecules. Moreover, the recalled molecules in “Dataset for Finetuning” were also removed to ensure novelty. The data points for the simulated experiment of virtual screening were also protonated at pH ranging between 7.3 and 7.5. Eventually, 36,364 data points were generated.

Deep Molecular Generation

The deep molecular generation implemented in our software program belonged to the well-established chemical language model^{11,12}. Briefly, SMILES sequences were firstly tokenized to atomic symbols that constituted a vocabulary together with tokens for starting, ending and padding. Next, an embedding layer of 128 dimensions followed by three-layer long-short term memory (LSTM) with 512 units were used to process these tensors, and a final dense layer with the same dimension as the length of the vocabulary was used to put out un-softmaxed value for each token. Dropout and layer normalization and gradient clipping were used to prevent overfitting. The model was trained in an auto-regressive manner with cross entropy loss, default Adam optimizer in

PyToch and learning rate schedule of cosine annealing ($T_{\text{max}} = 64$ for pretraining and $T_{\text{max}} = 32$ for finetuning). For pretraining, the dataset was randomly split into a training set and a validation set with the ratio of 99:1 while for finetuning, the ratio was 4:1. The batch size for pretraining (512) was also larger than that for finetuning (128). Moreover, the parameters of the embedding layer learned from model pretraining were frozen during backward propagation in model finetuning. For model sampling, the SMILES sequence was generated token by token until the ending token occurred or the maximum token length of 128 was reached. A widely used technique named “Temperature sampling”¹¹ was also included to control molecular novelty by stretching the probability distribution of candidate tokens.

Molecular Property Prediction

The molecular descriptors and other optional arguments for AutoGluon¹³ model training and prediction are illustrated in the main text. Herein, we describe the detailed settings of retrospective studies for three molecular property prediction models. In all these studies, AutoGluon models adopted normalized RDKit 2D descriptors and Morgan2 fingerprints. Two training modes were applied, i.e., “baseline” mode denoting that model was trained in the default configuration provided by AutoGluon, and “best” mode denoting that model was trained in the configuration of “Time limit (s)” = 3600 and “Model Quality Presets” = “best model quality”. All evaluation metrics were calculated by Scikit-learn¹⁴ Python package (version 1.2.2). For antibacterial activity prediction, hyperparameter optimization was firstly performed for Chemprop¹⁵ with 80% molecules from the MUBD dataset as the training set and through five-fold cross-validation with 10 iterations. Afterwards, the test set (20% remaining molecules) was used to compare the performances of optimized Chemprop and AutoGluon models. For cytotoxicity prediction, the average performance of FP-GNN on 32 tasks were used for comparison, and the AutoGluon

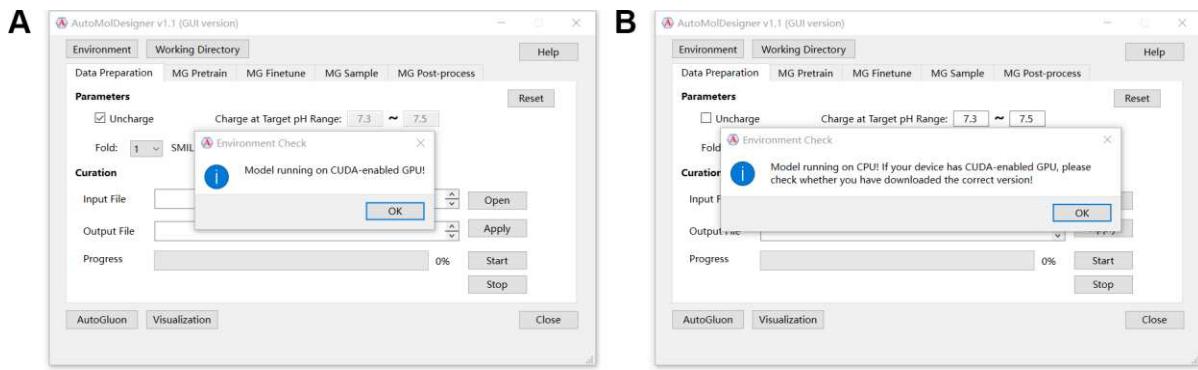


Figure S9. Detect running environment. **(A)** Model runs on CUDA-enabled GPU. **(B)** Model runs on CPU.

models were benchmarked in the same manner. For PPB fraction prediction, the performance of IDL-PPBopt¹⁰ was retrieved from the published results while AutoGluon was benchmarked with the training and test sets from IDL-PPBopt. All the benchmarking experiments were conducted in triplicate. It is worth noting that the last three prospective models trained with the full datasets were optimized by “Deployment Optimization”.

Other Useful Modules

The usage and implementation of other useful modules including data curation, model benchmarking and molecular visualization are described below. To begin with, there is a button located at the upper left corner of the “Environment” interface. It can be clicked to check whether the application is central processing unit (CPU) version or graphical processing unit (GPU) version. This functionality was realized by the PyTorch function in checking availability of compute unified device architecture (CUDA). The model will be run on CPU if CUDA-enabled GPU is not available (Figure S9). The second button called “Working Directory” can be used to set the current working directory. As a result, all the data generated will be saved under the directory of “your working directory/projects/your project name”. It should be noted that the “Reset” button is

available in each tab for resetting. In terms of “Data preparation” tab, several class methods of MolVS (version 0.1.1) and Dimorphite-DL¹⁶ (version 1.3.2) including “LargestFragmentChooser().choose()”, “Uncharger().uncharge()”, “Standardizer().standardize()”, and “DimorphiteDL().protonate()” were used to implement salts removal, charge neutralization, standard molecular standardization and protonation at a certain pH range. Moreover, “SmilesEnumerator().randomize_smiles()” that is a class method implemented in a standalone Python script called “SmilesEnumerator.py” was used to obtain the single randomized SMILES of a molecule. For “MG Post-process” tab, all the curation is based on Python built-in functions for data manipulation. “Molecular Visualization” tab is another supplementary module that provides a convenient way to show the designed molecules, and it was implemented by the “rdkit.Chem.Draw.MolsToGridImage()” function. Additionally, all the other molecular manipulations such as basic sanitization and molecular format conversion between SMILES and Structural Data were all made by RDKit¹⁷ (version 2023.3.3). With its built-in dataframe, Pandas (version 1.5.3) was used to process tabular data in the form of Comma-Separated Values (*.csv) or Microsoft Excel (*.xlsx).

Graphical User Interface Implementation

As mentioned in the section of “Implementation”, the graphical user interface (GUI) of AutoMolDesigner was designed based on Qt platform wrapped by PySide6 (version 6.5.1.1). The GUI code, the functional source code, and the related packages were packed into an executable and distributable standalone software by PyInstaller (version 5.12.0) for both MacOS and Windows operating systems (for Linux users, GUI version is currently not available, as command-line interface version that basically has the same operating logics can serve as an alternative). In detail, each long-term task will run on an independent thread through “QThread” and

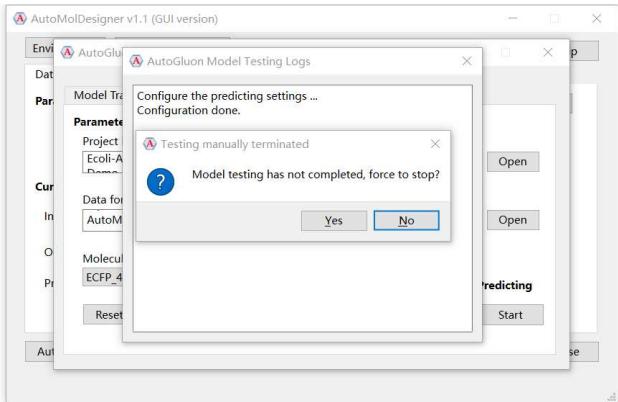


Figure S10. The message box pops up from the “AutoGluon” interface when users try to manually terminate the task.

“QThreadPool” classes of PySide6 to maintain responsiveness of the application. For molecular generation in the main interface, the progress bar is provided for each tab to indicate the progress of current task, while the adjacent “Stop” button can be used for manual termination. But it should be noted that pretraining and finetuning tasks will not stop immediately after the click of this button and will stop only when the current iteration is completed. For molecular property prediction in the “AutoGluon” interface, a logging dialogue will pop up during model training or prediction. If users try to close this dialogue before the task is completed, a message box will pop up as a reminder. Only if further confirmation is made according to the message box, the task can be manually terminated (Figure S10).

REFERENCES

- (1) Bagal, V.; Aggarwal, R.; Vinod, P. K.; Priyakumar, U. D., MolGPT: Molecular Generation Using a Transformer-Decoder Model. *J. Chem. Inf. Model.* **2022**, *62* (9), 2064-76.
- (2) Mendez, D.; Gaulton, A.; Bento, A. P.; Chambers, J.; De Veij, M.; Felix, E.; Magarinos, M. P.; Mosquera, J. F.; Mutowo, P.; Nowotka, M.; et al., ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Res.* **2019**, *47* (D1), D930-D40.
- (3) Wong, F.; Krishnan, A.; Zheng, E. J.; Stärk, H.; Manson, A. L.; Earl, A. M.; Jaakkola, T.; Collins, J. J., Benchmarking AlphaFold-enabled molecular docking predictions for antibiotic discovery. *Mol. Syst. Biol.* **2022**, *18* (9), e11081.
- (4) Mysinger, M. M.; Carchia, M.; Irwin, J. J.; Shoichet, B. K., Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. *J. Med. Chem.* **2012**, *55* (14), 6582-94.
- (5) Xia, J.; Tilahun, E. L.; Kebede, E. H.; Reid, T. E.; Zhang, L.; Wang, X. S., Comparative modeling and benchmarking data sets for human histone deacetylases and sirtuin families. *J. Chem. Inf. Model.* **2015**, *55* (2), 374-88.
- (6) Butina, D., Unsupervised Data Base Clustering Based on Daylight's Fingerprint and Tanimoto Similarity: A Fast and Automated Way To Cluster Small and Large Data Sets. *J. Chem. Inf. Comput. Sci.* **1999**, *39* (4), 747-50.
- (7) Xia, J.; Li, S.; Ding, Y.; Wu, S.; Wang, X. S., MUBD-DecoyMaker 2.0: A Python GUI Application to Generate Maximal Unbiased Benchmarking Data Sets for Virtual Drug Screening. *Mol. Inform.* **2020**, *39* (4), e1900151.
- (8) *Discovery Studio*: BIOVIA, 2016, <https://www.3ds.com/products-services/biovia/products/molecular-modeling-simulation/biovia-discovery-studio> (accessed on 5 April 2023).
- (9) Wu, J.; Xiao, Y.; Lin, M.; Cai, H.; Zhao, D.; Li, Y.; Luo, H.; Tang, C.; Wang, L., DeepCancerMap: A versatile deep learning platform for target- and cell-based anticancer drug discovery. *Eur. J. Med. Chem.* **2023**, *255*, 115401.
- (10) Lou, C.; Yang, H.; Wang, J.; Huang, M.; Li, W.; Liu, G.; Lee, P. W.; Tang, Y., IDL-PPBopt: A Strategy for Prediction and Optimization of Human Plasma Protein Binding of Compounds via an Interpretable Deep Learning Method. *J. Chem. Inf. Model.* **2022**, *62* (11), 2788-99.
- (11) Gupta, A.; Müller, A. T.; Huisman, B. J. H.; Fuchs, J. A.; Schneider, P.; Schneider, G., Generative Recurrent Networks for De Novo Drug Design. *Mol. Inform.* **2018**, *37* (1-2), 1700111.
- (12) Moret, M.; Friedrich, L.; Grisoni, F.; Merk, D.; Schneider, G., Generative molecular design in low data regimes. *Nat. Mach. Intell.* **2020**, *2* (3), 171-80.
- (13) Erickson, N.; Mueller, J.; Shirkov, A.; Zhang, H.; Larroy, P.; Li, M.; Smola, A., Autogluontabular: Robust and accurate automl for structured data. *arXiv preprint arXiv:2003.06505* **2020**.
- (14) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V., Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825-30.
- (15) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; et al., Analyzing Learned Molecular Representations for Property Prediction. *J. Chem. Inf. Model.* **2019**, *59* (8), 3370-88.
- (16) Ropp, P. J.; Kaminsky, J. C.; Yablonski, S.; Durrant, J. D., Dimorphite-DL: an open-source program for enumerating the ionization states of drug-like small molecules. *J. Cheminform.* **2019**, *11* (1), 14.

(17) Landrum, G. *RDKit: Open-Source Cheminformatics Software*. RDKit, 2023. <https://www.rdkit.org/> (accessed on 29 March 2023).