Report

Section 1: methods & metrics

- Describe what methods you have used, how they are independent from one another, what your workflow was, how you performed the cross-correlation between your methods. If applicable, please report estimated performance metrics of your methods, such as accuracy, sensitivity, false-discovery rate, etc., and how those metrics were obtained (e.g. cross-validation). Please provide key references if available.

Motivation

A major difficulty in developing a treatment for a novel pathogen, such as SARS-CoV-2, is the significant time investment required to produce bioactive ligands. A typical drug discovery cycle starts with a bioactive “hit” showing modest activity. The structure of this compound is then modified to improve efficacy, bioavailability, and safety. In the case of a global pandemic, there is a need to greatly expedite this process in order to create a valid treatment as soon as possible. This is further complicated by the limited amount of data available for a novel pathogen. Considering this, our main goal in this challenge was to develop a method for rapid bioactive ligand generation that could make use of as much available data as possible. Our proposed method was designed to consider the 2-dimensional (2D) structure of molecules in combination with a protein’s primary amino acid sequence. Not only does this provide advantages regarding computational efficiency, it also ensures that we are not dependent on 3D crystallographic data, as acquiring a protein’s crystal structure can be equally difficult and time consuming. As a pandemic-like situation evolves rapidly, it is imperative that our method be able to adapt to additional data as it is reported.

Pipeline description

We designed our workflow (Fig 1) based on the two molecules published by Dai et al.[1]. The existence of two compounds, each with an IC_{50} below 100 nM, allowed us to construct a ligand based virtual screening pipeline. Following the similarity principle[2], structurally similar molecules are more likely to have similar properties—in our case it follows that the expectation of finding active molecules similar to the Dai compounds is higher than in the overall chemical space.
To provide a list of 10,000 ranked molecules according to the challenge’s rules, we designed a workflow based on three methods that we used in cascade.
The goal of the first method (Method 1, Fig 1a) was to ensure that molecules screened have a good chance to be synthesizable. This was achieved through the use of a predictive synthesizability score. Molecules predicted to be difficult to synthesize were omitted, as the potential time and energy expenditure required to synthesize and test these compounds made them less viable candidates. The role of the second method (Method 2, Fig 1a) was to constrain

the ~1 billion molecules virtual library to only those compounds which were close in the chemical space to the Dai template ligands. This represents a region of chemical space which was most applicable to our predictive model. We defined the threshold of this method such that 10,000 molecules were left for the final ranking. To provide the final list of ranked molecules, we used an ensemble of deep learning models trained to predict if a given protein-molecule pair would bind together (Method 3, Fig 1d).

Overall, the three methods are independent from one another and utilize different types of information relevant to the challenge. Method 1 and 2 are rule-based while Method 3 is data driven. Method 1 estimates the molecules' synthesizability, whilst Method 2 ranks them according to their structural similarity to the known active template ligands. In contrast, Method 3 is a template-free method, which considers the structural information of the target protein in order to perform ranking. Furthemore, Method 1 differs from Method 2 as it is used to estimate the feasibility of producing the molecules whereas Method 2 estimates their relative position in chemical space.

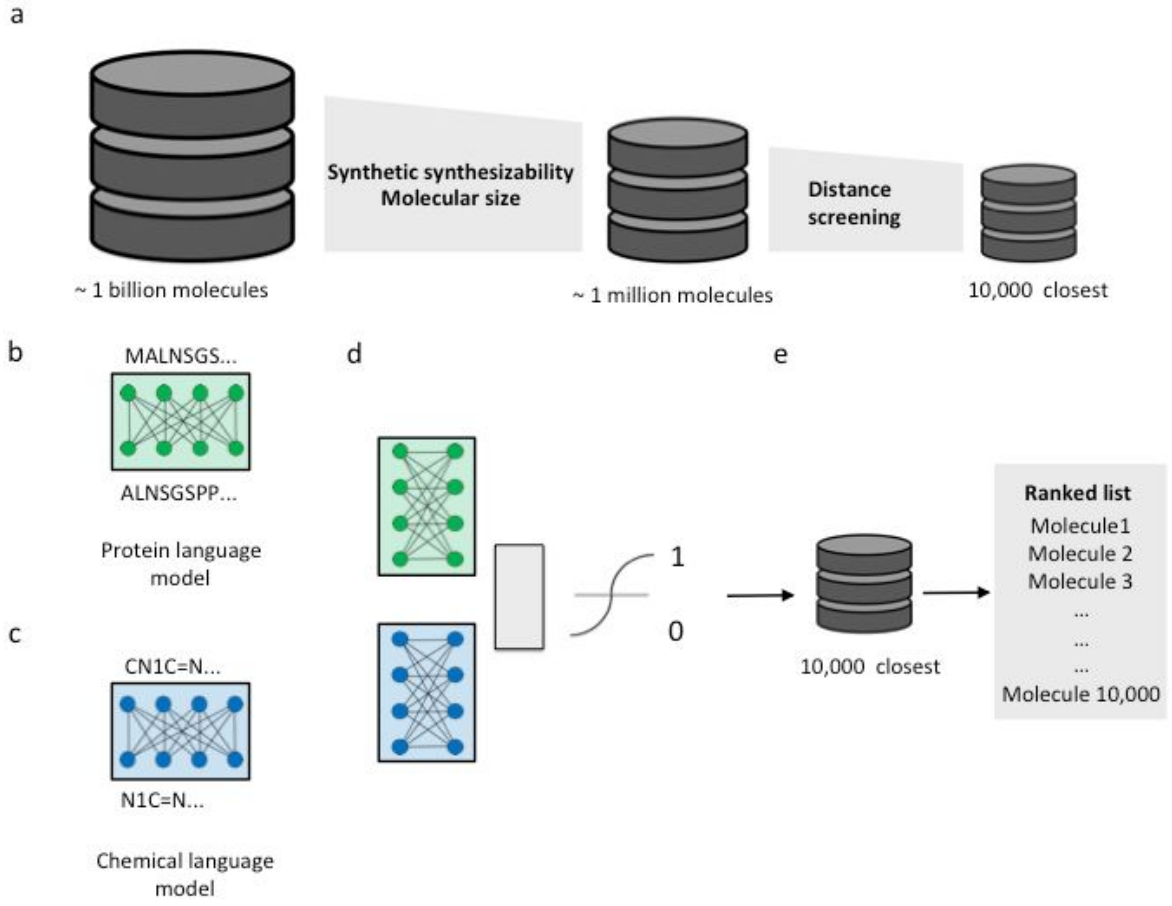

Figure 1. Workflow overview. **a,** The virtual library of more than one billion molecules is pruned by molecular size and synthetic accessibility before being reduced to the 10,000 closest molecules to the Dai compounds. **b,** A protein language model is pretrained on data from Swiss-prot. **c,** A chemical language model is pretrained on molecules encoded as SMILES strings from ChEMBL. **d,** Both language models are joined together and fine-tuned with an

additional top layer to predict if a given protein-molecule pair will interact with an activity below 100 nM. **e,** The final model is trained and used as an ensemble to rank the 10,000 closest molecules.

Methods

Method 1. The aim of the first method (Fig 1a) was to increase the likelihood that the pool of molecules we ranked would be synthesizable. We used the synthetic accessibility score (SAscore)[3], which assigns a score between 1 and 10 to a molecule which is reflective of its synthetic accessibility. Molecules close to 1 are easy to synthesize, whereas those close to 10 are hard. We defined our SAscore threshold such that the two Dai compounds would have passed the Method 1 screen within a small margin.

Method 2. The second method (Fig 1a) was used to constrain the chemical space to the vicinity of the Dai compounds. Following the similarity principle, we would expect that compounds in this region were more likely to be active. For this method, we represented molecules as their Morgan fingerprint[4] and defined similarity as the mean Tanimoto distance between the two Dai compounds. The Morgan fingerprint was used as it had been previously shown to outperform other structural fingerprints[5]. The distance threshold was defined to keep only the 10,000 most similar molecules for the final step of the workflow.

Method 3. To rank the top 10,000 molecules against our protein target of interest, we trained a deep learning model to predict if a given protein-molecule piar would bind—the so-called Drug Target Interaction (DTI) task [6] (Fig 1d). Specifically, we trained our model to predict if a given molecule would bind to the respective protein with an affinity less than or equal to 100 nM. Following methods used to rank documents in the natural language processing community [cite], we used the output score of the last layer of the deep learning model to rank the compounds. Moreover, as it has been shown that an ensemble of deep learning models gives a prediction boost over a single instance of the model[7], we used the mean of an ensemble of five models for the final prediction. Of note, the ensemble consists of the same model trained five different times on the same data—the predictions' differences between each model instance is due to the random initialization of the deep learning model. This differing initialization leads to slightly different solutions for a given training procedure and training data.
To train our model, we used data from BindingDB[8], which reports a triplet of the protein, molecule and measured activity between them. After filtering for protein-molecule pairs having a reported K_d and processing the data, we were left with ~30,000 triplets for training. To permit deep learning models training, which are notoriously data hungry[9], we used pretraining and fine-tuning[10].
As a pretraining objective, both models were trained with the so-called language model approach, which consist of iteratively predicting the next token in a sequence given the previous tokens[11]. We pretrained a model for each type of input, namely, a model that takes as input a protein amino acid sequence (Fig 1b) and another model which is trained on the SMILES string representation of a molecule[12] (Fig 1c). The protein language model (PLM) was trained on

proteins from the Swiss-Prot database[13] while the chemical language model (CLM) was trained on molecules from ChEMBL27[14].
Those two pretrained models were combined into a single model, with the addition of a top layer to output a binding prediction. The full model was fine-tuned to predict if a protein-molecule pair would have an activity below 100 nM. To rank molecules against each other, the output value—which is given after the sigmoid activation function—of the model for a given prediction was used.

BindingDB processing. We restricted our protein-molecule pair selection to those with a reported K_d. We limited the proteins' size to an amino acid length of 1000 and the SMILES string representation to molecules of 100 tokens. We binarized the activity values into two categories; interaction between a protein and a molecule was defined as active if the reported activity was below 100 nM, and inactive otherwise. All protein-molecule pairs having entries in both binarized activity groups (active and inactive) were completely removed. Duplicate entries were also removed, leaving only unique triplets.

ChEMBL27 processing. All molecules were extracted as SMILES strings and canonicalized with RDKit[15]. Only molecules with most representative tokens from ChEMBL were kept (see code repository). All molecules represented by a SMILES string longer than 100 tokens were not kept, resulting in a final dataset size for training of 1,725,404 unique SMILES strings.

Swiss-prot processing. We downloaded Swiss-prot, a manually annotated and reviewed subset of UniProt, and kept only sequences with known amino acids. Protein length in terms of the number of amino acids was limited to 1000, resulting in 454,512 unique protein sequences.

Data split for language models. To train the chemical language model, we split our processed ChEMBL27 randomly and used 95% of the data for training, and 5% for validation. The same procedure was used to split our processed data from Swiss-prot to train the protein language model.

Data split for Model 3 training. To train our deep learning model to predict protein-molecule interactions, we used a cross-validation approach. Given that the potential protein targets are dissimilar compared to data in BindingDB, we defined our cross-validation splits such that all proteins within the validation split for a given cross-validation fold have a high degree of dissimilarity towards the proteins used for training. Although this construction makes the prediction task harder for our model, it better reflects the deployment task of the protein targets linked to the coronavirus.
To cluster similar proteins together, we used MMseqs2, an ultra fast and sensitive sequence search and clustering suite[16]. We set a similarity threshold of 80% to cluster the proteins. To create the cross-validation folds, we randomly sampled the cluster representative sequence, and took all the other members of that cluster in the fold it was randomly assigned to.

Section 2: targets

- Describe for each protein target: why you chose it, from which source you obtained it (e.g., insidecorona.net / covid.molssi.org / rcsb.org) and why this is the best quality structure, if any pre-processing (e.g., energy minimization, residue correction, alternative folding, …) was performed.

We selected only one target, the main protease of SARS-CoV-2, Mpro. Our choice to submit a solution to this challenge only for one—and not three—proteins involved with Covid-19 follows the computational approach we chose to use. Our approach was designed to be data-driven, exploiting existing activity data in order to make an accurate estimation of a molecule's potential use as a covid treatment. At the beginning of the project, no known actives against covid-19 had been reported, and hence, we intended to solely make use of the protein-molecule interaction information. However, during the course of the challenge *Dai et al.* produced two compounds showing nanomolar activity towards a covid specific target, namely, the Mpro protease. In light of these developments, we sought to integrate this data within our method. While we indeed could have provided a ranked list for two other targets, the imbalance in information relevant to our target problem meant that selecting such targets would not have been scientifically motivated for our data-driven approach. Therefore, we made the difficult—but we hope, justified –choice to submit only one list.
In addition to providing bioactive ligands as a reference, our choice to focus on the Mpro target allowed us to make use of the PostEra dataset (https://postera.ai/covid/activity_data), providing us with the opportunity to create a relevant test set for our selected target For the ranking experiment, we used the protein sequence as reported by *Dai et al*. with the PDB accession number 6LZE.

Section 3: libraries

- Describe which libraries you have used, how they were combined, if any compounds were removed / added, why additions are relevant, any unique features of your library, etc. Please provide the sources you obtained the libraries from (if publicly available). Describe the procedure of data preparation (removal of duplicates, standardization, etc). Indicate if different libraries were used for different targets, and why. If possible, provide a download link to your version of the library.

We selected the following libraries: ZINC, SWEETLEAD, JEDI-AMS and CAS antiviral as advised by JEDI. All molecules were encoded as canonical SMILES strings with RDKit. SMILES strings with selective tokens were kept, and limited to a size of 100 tokens.

Section 4: results

- Briefly describe your key findings, any interesting trends in your data, a description of your top 5 compounds for each target. If possible, provide a link to a code and/or data repository. Please do not submit randomly selected compounds!

Deep learning model training. We explored multiple training configurations to train our deep learning ensemble to predict activity between a molecule and protein.We found that pretraining both the protein and the chemical language model—that is, the parts of the deep learning model accepting the inputs (Fig b,c and d)—was beneficial in terms of the standard deviation over the three cross-validation folds (Fig. 2a) compared to the full model trained from scratch (Fig. 2c). Moreover, fine-tuning the entire model to predict activity, rather than freezing the pretrained models and training only the added layer, proved to be beneficial (Fig. 2d). Finally, despite a significant divergence between the training and the validation losses (Fig 2a), we found that adding more regularization to the network was not yielding improved losses (Fig 2b).
It has to be noted that a level of overfitting (Fig. 2a) is expected with our cross-validation split construction as the validation set is constructed such that the proteins in the validation set possess a certain degree of dissimilarity towards the training proteins.

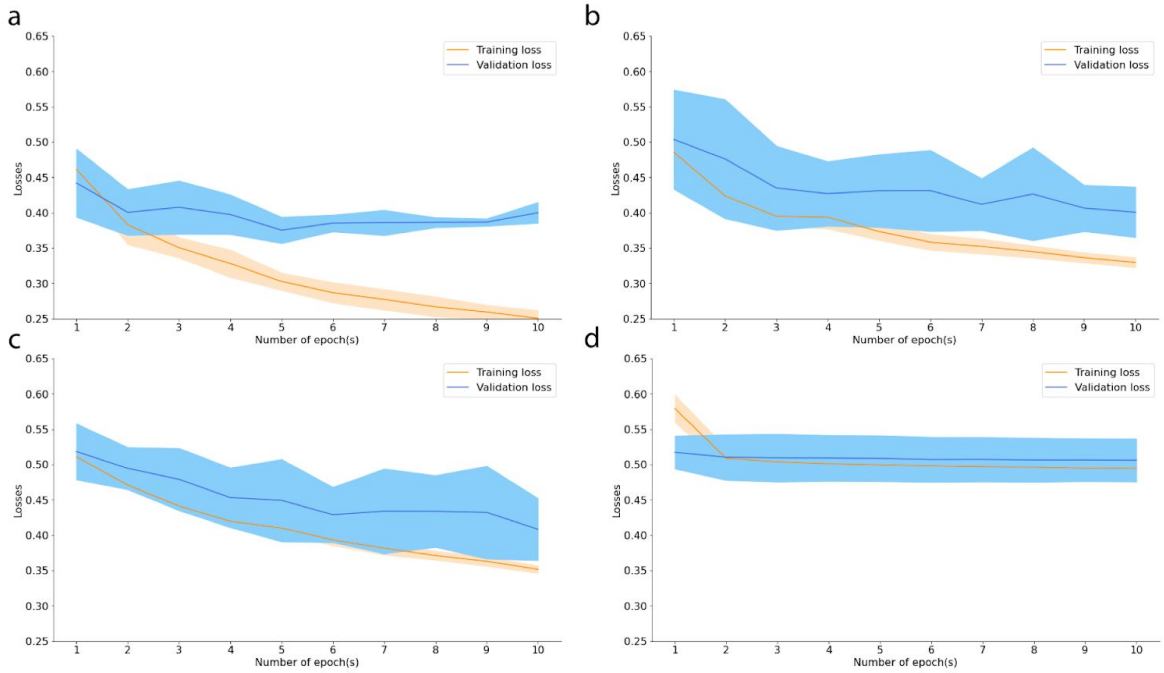

Figure 2. Cross-validation losses for the deep learning model (Model 3). **a,** Model fine-tuned to predict activity with both the protein and the chemical language models pretrained. **b,** Model fine-tuned to predict activity with both the protein and the chemical language model pretrained with more regularization. **c,** Model trained without pretrained protein and chemical language models. **d,** Model trained to predict activity with both the protein and the chemical language models pretrained, but with both pretrained language models frozen during activity prediction.

Deep learning model selection. Based on the results of the cross-validation, we trained a model on the entire data for ten epochs (Fig. 2a). Based on the validation loss, we selected the first five epochs for model selection. To select our model, we used the data made available by PostEra. We compared the Spearman rank correlation of each model against the reported experimental results. We decided to pick the model at epoch 2, as the model at epoch 5 does not show a significant improvement, but does require more training time (Table 2).

Table 1: Spearman rank correlation of the ensemble of five deep learning models on the PostEra data. The mean of the ensembles is reported.

Epoch number	Spearman rank correlation
1	0.387
2	0.439
3	0.430
4	0.412
5	0.458

Benchmarking deep learning approach against other methods. As a quality control of our approach, we compared the performance of our model in terms of Spearman rank correlation to that of a series of molecular descriptors and fingerprints on the PostEra data.
The two Dai compounds were used as a reference in order to rank the PostEra data. For the molecular fingerprints, the average Tanimoto distance (1 - Tanimoto coefficient) to the Dai compounds was used. For molecule properties, such molecular weight or number of hydrogen bond donors, we measured the average absolute difference in the value with respect to the two Dai compounds. We found that the deep learning model ensemble performs better than all tested methods with a Spearman rank correlation of 0.439. The second best method is based on the Morgan fingerprint with a Spearman rank correlation of 0.236.
Based on those results, we validated the use of our deep learning ensemble to finalise the ranking of the top 10,000 compounds.

Table 2: Spearman rank correlation of the deep learning model ensemble compared with a series of molecular descriptors and properties, calculated on the PostEra data.

Methods	Spearman rank correlation

MACCS	0.014
Morgan fingerprint	0.236
CATS	-0.028
Molecular weight	-0.140
ClogP	-0.000
FeatMorgan fingerprint	0.121
Layered fingerprint	0.055
Atom-pair fingerprint	-0.042
Avalon fingerprint	-0.076
Torsional fingerprint	-0.050
Hydrogen bond acceptor	0.158
Hydrogen bond donor	0.137
Aliphatic heterocycles	-0.043
Aromatic heterocycles	-0.032
Deep learning ensemble @epoch 2	0.439

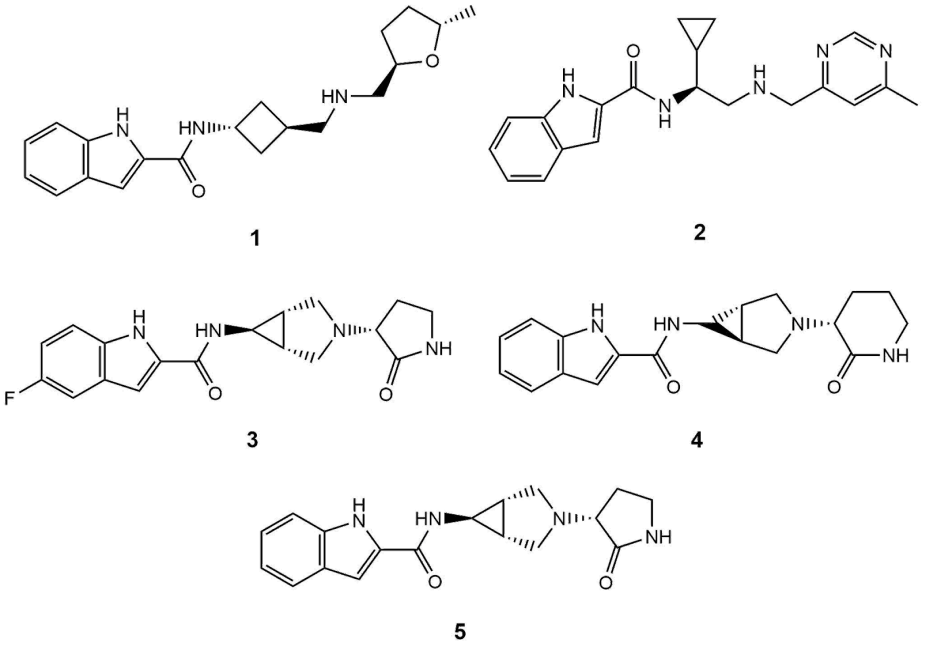

Figure 3. Top five molecules ranked by the ensemble of deep-learning models following the application of the synthetic accessibility score and the narrowing of chemical space by the Morgan fingerprint distance to the Dai templates.

Fig. 3 shows the top five compounds ranked by our deep learning model. Common among each of the molecules, as well as the Dai compounds, is the presence of an indole ring with an adjacent amide group. Compounds **3**-**5** also possess a lactam ring, which in the Dai publication was shown to form a hydrogen bonding interaction with the histidine side chains within the Mpro active site. In contrast to the Dai compounds, our proposed structures lacked the key aldehyde warhead which facilitated the C-S covalent linkage. Due to the associated toxicities inherent with reactive aldehyde groups[17,18], we saw the exclusion of this moiety as advantageous, as it reduced the likelihood of potential toxicity, thus increasing the compounds applicability as potential drug candidates.

Code availability

Script to reproduce the experiments are available in the following GitHub repository: https://github.com/michael1788/jedichallenge_submission/

References

1. Dai, W. *et al.* Structure-based design of antiviral drug candidates targeting the SARS-CoV-2 main protease. *Science* **368**, 1331–1335 (2020).

2. Johnson, M. A. & Maggiora, G. M. Concepts and applications of molecular similarity. (1990).
3. Ertl, P. & Schuffenhauer, A. Estimation of synthetic accessibility score of drug-like molecules based on molecular complexity and fragment contributions. *J. Cheminform.* **1**, 8 (2009).
4. Rogers, D. & Hahn, M. Extended-connectivity fingerprints. *J. Chem. Inf. Model.* (2010).
5. O'Boyle, N. M. & Sayle, R. A. Comparing structural fingerprints using a literature-based similarity benchmark. *J. Cheminform.* **8**, 36 (2016).
6. Lee, I., Keum, J. & Nam, H. DeepConv-DTI: Prediction of drug-target interactions via deep learning with convolution on protein sequences. *PLoS Comput. Biol.* **15**, e1007129 (2019).
7. Nogueira, R., Jiang, Z. & Lin, J. Document Ranking with a Pretrained Sequence-to-Sequence Model. *Preprint at http://arxiv.org/abs/2003.06713* (2020).
8. Liu, T., Lin, Y., Wen, X., Jorissen, R. N. & Gilson, M. K. BindingDB: a web-accessible database of experimentally determined protein-ligand binding affinities. *Nucleic Acids Res.* **35**, D198–201 (2007).
9. Brown, T. B. *et al.* Language Models are Few-Shot Learners. *Preprint at http://arxiv.org/abs/2005.14165* (2020).
10. Howard, J. & Ruder, S. Universal Language Model Fine-tuning for Text Classification. *Preprint at http://arxiv.org/abs/1801.06146* (2018).
11. Bengio, Y., Ducharme, R., Vincent, P. & Jauvin, C. A Neural Probabilistic Language Model. *J. Mach. Learn. Res.* **3**, 1137–1155 (2003).
12. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **28**, 31–36 (1988).
13. Boeckmann, B. *et al.* The SWISS-PROT protein knowledgebase and its supplement

TrEMBL in 2003. *Nucleic Acids Res.* **31**, 365–370 (2003).

14. Gaulton, A. *et al.* The ChEMBL database in 2017. *Nucleic Acids Res.* **45**, D945–D954 (2016).
15. Landrum, G. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling. (2013).
16. Steinegger, M. & Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.* **35**, 1026–1028 (2017).
17. LoPachin, R. M. & Gavin, T. Molecular mechanisms of aldehyde toxicity: a chemical perspective. *Chem. Res. Toxicol.* **27**, 1081–1091 (2014).
18. Ahmed Laskar, A. & Younus, H. Aldehyde toxicity and metabolism: the role of aldehyde dehydrogenases in detoxification, drug resistance and carcinogenesis. *Drug Metab. Rev.* **51**, 42–64 (2019).