Team name	PhD Students Institute for Machine Learning & LIT AI Lab, Johannes Kepler University Linz
Team member(s) (firstname lastname; …)	Peter Ruch Hamid Eghbal-zadeh Christina Halmich Johannes Schimunek Philipp Seidl Helga Ludwig Andreas Mayr Andreu Vall Elisabeth Rumetshofer Michael Widrich Philipp Renz
Affiliation	Institute for Machine Learning, Johannes Kepler University Linz LIT AI Lab, Johannes Kepler University Linz Institute of Computational Perception, Johannes Kepler University Linz
Contact email	ruch@ml.jku.at
Contact phone number (optional)	
Protein targets (for example: 3CLPro/Nsp5, BoAT1, Fc Receptor, Furin, IL6R, M protein, Nspx, OrfXx, N, E, etc…) | 3 required	3CLPro PLpro RdRP

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.

We performed a ligand-based screening [1, 2] using **five independent machine learning methods for compound selection** namely RandomForests, Gradient Boosting and Self-Normalizing Networks [5] trained on chemical descriptor data as well as LSTM [13] based sequence models for learning directly from the molecular SMILES representation.

RandomForests and Gradient Boosting are decision tree-based methods that differ in the way the trees are constructed and how results are aggregated. Self-Normalizing Networks are simple feed-forward neural networks that use the SeLU activation function. The LSTM based models are autoregressive neural networks that are used for processing sequence data (i.e. it uses SMILES strings instead of chemical descriptor vectors as input). Both LSTMs are further trained using different training strategies (fine-tuning versus from scratch).

The models were trained on a dataset based on data for five PubChem assays (see Section 2). The best model was selected based on best AUROC on the validation set (AUROC scores for every fold and assay were higher than 0.65 for all models).

Methods:

RandomForest:

We trained a single task classification Random Forest for every assay using the Random Forest implementation available in Scikit-Learn [8]. Hyperparameters were selected via Hyperparameter search using the HyperOpt [9] library.

Gradient Boosting:

For every assay we trained a Gradient Boosting model using the DART [6] algorithm available in LightGBM [7]. Hyperparameter search was performed via Bayesian optimization using the Ax library and Ray Tune [10].

Self-Normalizing Networks:

We trained self-normalizing neural networks [5] utilizing the SeLU activation function in the multi-task setting. The model was implemented in PyTorch, the Hyperparameter search was performed via a combination of Ax and Ray Tune [10].

SMILES - LSTM (transfer-learning):

This approach utilized a transfer-learning strategy where a neural network trained on a combination of ChEMBL, PubChem ZINC15 assay data was fine-tuned on our SARS-CoV-2 dataset. The model was implemented using TensorFlow + Keras.

SMILES - LSTM (trained from scratch):

For this approach several LSTM based Neural Network were trained with a Multi-Task objective on SMILES as input. Further, a target-specific model selection has been incorporated in order to identify models that performed best on a specific target which was then used for inference. This model was implemented in PyTorch.

Merging of model results:

Due to the large number of compounds we only considered the top permille with the highest prediction scores when merging the results of the different models. For every target we selected the compounds that were ranked high for the corresponding assays by multiple methods. The compounds in the final list were then ranked by the geometric mean of the predictions scores (the prediction scores for different models don't always resemble probabilities or are heavily skewed due to the unbalanced-ness of the training dataset, (Platt-) scaling of the raw predictions did not lead to satisfying results).

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.

As we perform ligand-based screening, we rely on assay measurements. Due to the novelty of SARS-CoV-2 no relevant bioassay measurements exist yet. Therefore, it was required to use assays from related viruses (e.g. SARS or MERS).

Target 1: 3CLpro
For this target we used data from PubChem assays AID1706, AID1879 that contain compounds screened against 3CLpro of the SARS virus.

Target 2: PLpro
For PLpro we relied on two other PubChem assays AID485353, AID652038 with data on SARS PLpro.

Target 3: RdRP
Due to the lack of RdRP assay data for SARS-CoV-2, SARS or MERS, we used RdRP assays for the Poliovirus as a proxy, namely PubChem Assay AID588519. This choice is relatively farfetched and predictions **should not be prioritized** for in-vitro testing. As a positive side effect, the inclusion of this assay and corresponding compounds greatly improved the compound-similarity based clustering of the folds (as discussed below) which also lead to better generalization estimates for the other assays.

The assay measurements were used to compile a new training dataset. As discussed by Mayr et al. [2], In order to obtain better estimates for the generalization error and to account for batch effects for compounds, as commonly encountered in High Throughput Screening, the compounds were clustered based on dice similarity for ECFP4 folded to 2048 bit. The clusters of compounds were then assigned to separate train, test and validation folds that were used for model training and evaluation.

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.

ZINC15:

We downloaded the "at least annotated" tranche of ZINC15 [3].

SWEETLEAD:

We downloaded SWEETLEAD from SimTK [4].

Preprocessing (identical for both libraries):

Compound SMILES strings were standardized using MOLVS . We then utilized RDKit [12] Release 2020.03.01 to compute several structural keys (MACCS Keys as well as "Tox Keys" based on SMARTS of known toxicophores). PubChem fingerprints were computed using a Python implementation available in the DeepPurpose [11] library. The feature types were selected based on initial performance results from models trained on our training dataset based on the PubChem assays mentioned in section 2 and using combinations of additional features (e.g. chemical descriptors computed using Mordred or RDKit Fingerprints). We were required to make a tradeoff between model accuracy against expected time to compute all the features for approximately 1.5 billion molecules which ultimately lead to the selection of MACCS Keys, Tox Keys and PubChem Fingerprints as features for the inference models.

Feature computation was parallelized using Apache Spark on a single Desktop with a Threadripper 3960x and required approximately 10 days for ZINC15.

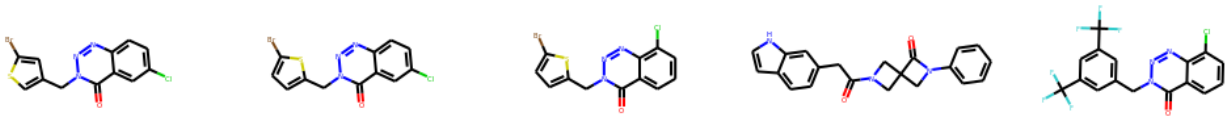
Section 4: results

Briefly describe you 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!

Manual inspection of the top-ranked compounds. We ran the Top20 Molecules for every target through the SwissADME [14] webservice and inspected the reports. For the Top20 Mpro compounds there are few alerts. For PLpro there are multiple alerts which might indicate problems when utilizing the compound as drug. Based on those reports, the original ranking by the ML methods (see above) has been retained.

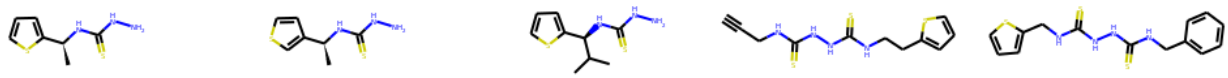
All compounds are available in ZINC15 or SWEETLEAD and should be easily obtainable.

Top 5 Molecules for Mpro



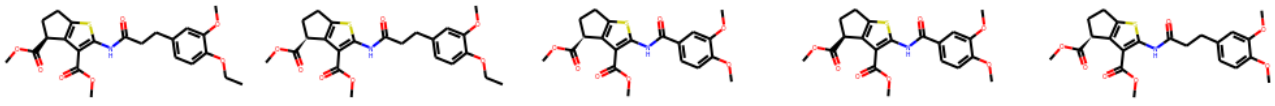
ZINC001656660237 ZINC001559428262 ZINC001614548380 ZINC001181808260 ZINC001644025384

Top 5 Molecules for PLpro



ZINC000035737173 ZINC000061364822 ZINC000049558756 ZINC000758193974 ZINC000047582111

Top 5 Molecules for RdRP



ZINC000040999979 ZINC000040999980 ZINC000000622590 ZINC000000622591 ZINC000040994873

References:

[1] Hofmarcher, Markus, Andreas Mayr, Elisabeth Rumetshofer, Peter Ruch, Philipp Renz, Johannes Schimunek, Philipp Seidl et al. "Large-scale ligand-based virtual screening for SARS-CoV-2 inhibitors using deep neural networks." *Available at SSRN 3561442* (2020).

[2] Mayr, Andreas, Günter Klambauer, Thomas Unterthiner, Marvin Steijaert, Jörg K. Wegner, Hugo Ceulemans, Djork-Arné Clevert, and Sepp Hochreiter. "Large-scale comparison of machine learning methods for drug target prediction on ChEMBL." *Chemical science* 9, no. 24 (2018): 5441-5451.

[3] Sterling, Teague, and John J. Irwin. "ZINC 15–ligand discovery for everyone." *Journal of chemical information and modeling* 55, no. 11 (2015): 2324-2337.

[4] Novick, Paul A., Oscar F. Ortiz, Jared Poelman, Amir Y. Abdulhay, and Vijay S. Pande. "SWEETLEAD: an in silico database of approved drugs, regulated chemicals, and herbal isolates for computer-aided drug discovery." *PLoS One* 8, no. 11 (2013): e79568.

[5] Klambauer, Günter, Thomas Unterthiner, Andreas Mayr, and Sepp Hochreiter. "Self-normalizing neural networks." In *Advances in neural information processing systems*, pp. 971-980. 2017.

[6] Vinayak, Rashmi Korlakai, and Ran Gilad-Bachrach. "Dart: Dropouts meet multiple additive regression trees." In *Artificial Intelligence and Statistics*, pp. 489-497. 2015.

[7] Ke, Guolin, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. "Lightgbm: A highly efficient gradient boosting decision tree." In *Advances in neural information processing systems*, pp. 3146-3154. 2017.

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

[9] Bergstra, James, Daniel Yamins, and David Cox. "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures." In *International conference on machine learning*, pp. 115-123. 2013.

[10] Liaw, Richard, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E. Gonzalez, and Ion Stoica. "Tune: A research platform for distributed model selection and training." *arXiv preprint arXiv:1807.05118* (2018).

[11] Huang, Kexin, Tianfan Fu, Cao Xiao, Lucas Glass, and Jimeng Sun. "DeepPurpose: a Deep Learning Based Drug Repurposing Toolkit." *arXiv preprint arXiv:2004.08919* (2020).

[12] Landrum, Greg. "RDKit: Open-source cheminformatics." (2006): 2012.

[13] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9, no. 8 (1997): 1735-1780.

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