Artificial Intelligence Guided *De Novo* Molecular Design targeting COVID-19

Srilok Srinivasan[1], Rohit Batra[1], Henry Chan[1,3], Ganesh Kamath[2], Mathew J. Cherukara[1], and Subramanian Sankaranarayanan[1,3]

[1]Center for Nanoscale Materials, Argonne National Laboratory, Lemont, Illinois 60439, United States
[2]Dalzielfiver LLC, 3500 Carlfield St., El Sobrante, CA 94803, United States
[3]Department of Mechanical and Industrial Engineering, University of Illinois,Chicago, Illinois 60607, United States

June 29, 2020

Abstract

An extensive search for active therapeutic agents against the SARS-CoV-2 is being conducted across the globe. Computational docking simulations have traditionally been used for *in silico* ligand design and remain popular method of choice for high-throughput screening of therapeutic agents in the fight against COVID-19. Despite the vast chemical space (millions to billions of biomolecules) that can be potentially explored as therapeutic agents, we remain severely limited in the search of candidate compounds owing to the high computational cost of these ensemble docking simulations employed in traditional *in silico* ligand design. Here, we present a *de novo* molecular design strategy that leverages artificial intelligence to discover new therapeutic biomolecules against SARS-CoV-2. A Monte Carlo Tree Search algorithm combined with a multi-task neural network (MTNN) surrogate model for expensive docking simulations and recurrent neural networks (RNN) for rollouts, is used to sample the exhaustive SMILES space of candidate biomolecules. Using Vina scores as target objective to measure binding of therapeutic molecules to either the isolated spike protein (S-protein) of SARS-CoV-2 at its host receptor region or to the S-protein:Angiotensin converting enzyme 2 (ACE2) receptor interface, we generate several (∼100's) new biomolecules that outperform FDA (∼1000's) and non-FDA biomolecules (∼million) from existing databases. A transfer learning strategy is deployed to retrain the MTNN surrogate as new candidate molecules are identified - this iterative search and retrain strategy is shown to accelerate the discovery of desired candidates. We perform detailed analysis using Lipinski's rules and also analyze the structural similarities between the various top performing candidates. We spilt the molecules using a molecular fragmenting algorithm and identify the common chemical fragments and patterns – such information is important to identify moieties that are responsible for improved performance. Although we focus on therapeutic biomolecules, our AI strategy is broadly applicable for accelerated design and discovery of any chemical molecules with user-desired functionality.

1 Introduction

In view of the unprecedented human and economic loss resulting from the COVID-19 pandemic, there is an extensive amount of research being conducted across the globe to design effective therapeutic agents and vaccines against the SARS-CoV-2. Computational docking simulations have traditionally been used for *in silico* ligand design and remain a popular method of choice for high-throughput screening of therapeutic agents in the fight against COVID-19. For instance, Smith et al. [29] recently conducted virtual high-throughput screening of nearly 9000 small-molecules that bind strongly to either the isolated spike protein (S-protein) of SARS-CoV-2 at its host receptor region (thus, hindering the viral recognition of the host cells) or to the S-protein:Angiotensin converting enzyme 2 (ACE2) interface thus reducing the host-virus

interactions. In their work, they have successfully identified 77 ligands (24 of which have regulatory approval from the Food and Drug Administration, FDA) that satisfied one of the above two criteria. Such ensemble docking studies have been traditionally employed to design active agents against viruses. However, given the relatively high computing cost of docking simulations, one can typically only screen a few 1000 molecules even with the use of leadership computing resources (For instance, Summit at ORNL was used by Smith et al. [29] for their ensemble docking calculations). Despite the vast chemical space [24] (millions to billions of biomolecules) that can be potentially explored as therapeutic agents, we remain severely limited in the search of candidate compounds owing to the high computational cost of the ensemble docking simulations employed in traditional *in silico* approaches [29, 22].

A crucial first step towards the development of new therapeutic ligands is the identification and characterization of new candidate compounds that are synthetically feasible and effective for the proposed application. Despite advances in high-throughput screening, only a small fraction of the enormous possibilities of organic compounds have been explored. *In-silico* approaches to designing compounds with desired properties has become an active field of research over the last two decades [25], and have the potential to drastically reduce the cost and time to design new therapeutic molecules. More recently, advances in artificial intelligence (AI) have accelerated this search by building cheaper surrogate models i.e. machine learning (ML) models (instead of expensive docking simulations) that are trained against a reasonably elaborate set of training dataset derived from ensemble docking simulations. A variety of AI methods have been applied to the problem of generating and evaluating new biomolecules including reinforcement learning [11, 31], Bayesian molecular design [14], generative models such as variation autoencoders[7, 12] and recurrent networks[11, 3, 26].

We have shown in a recent work [2] that a machine learnt model can be used to quickly and effectively screen active agents for their potential effectiveness against the COVID-19 virus. By using a computationally cheaper ($\sim$100-1000$\times$) ML model, we were able to rapidly screen millions of bio-molecules from existing databases and rank-order candidates in terms of their therapeutic prowess. We developed a random forest model using the training datasets derived from ensemble auto-docking simulations available in literature. Such ML models are computationally much cheaper than the auto-docking simulations and allows us to significantly expand the search space and screen millions of potential therapeutic agents against COVID-19. Our screening was based on the binding affinity to either the isolated SARS-CoV-2 S-protein at its host receptor region or to the S-protein:ACE2 interface complex, thereby potentially limiting and/or disrupting the host-virus interactions. The ligands screened from the CureFFI [8] and DrugCentral [34] datasets, using our ML model, were subsequently validated by auto-docking simulations. Other strategies for repurposing theurapeutics based on the principle of 'neighborhood behaviour' in the chemical space [6, 19, 9], docking simulations [36, 20], quantum mechanical scoring [5] and sequence comparison statistics [16] were also recently reported. While these examples are effective in screening and repurposing the molecules within an existing database, we are still limited within the known chemical space.

Here, we demonstrate that one can take full advantage of the recent advances in artificial intelligence and in particular, deep learning methods (multi-task and recurrent neural networks) to efficiently search the elaborate chemical phase space available to us and accelerate the discovery of candidate biomolecules that outperform pre-existing ones in their therapeutic prowess against COVID-19. We draw inspiration from the success of AI algorithms such as Monte-Carlo Tree Search (MCTS) which have proven to be extremely effective in computer games (Go and several others) [28] against human competitors. MCTS has also been shown to be an effective and much faster means of identifying retrosynthesis routes in organic chemistry[27, 37]. More recently, Kajita et al. [15] demonstrated the effectiveness of MCTS for autonomous molecular designs. A schematic of our workflow is shown in Figure 1. Our *de novo* molecular design is made possible by deploying a cheap surrogate multi-task neural network (MTNN) in place of docking simulations and searching the exhaustive SMILES chemical space of millions of prospective molecules using AI algorithms such as MCTS with rollout(s) using recurrent neural networks (RNN). We represent the SMILES string of a molecule as a tree with each character representing a node, so that, any path from the head node to the terminal node represents a unique SMILES string. The binding affinities for valid SMILES strings are computed using the MTNN model trained against data obtained from ensemble docking simulations available in literature [29]. Transfer learning strategies are employed to improve the accuracy of MTNN model, particularly for molecules with strong binding, to better the guide the MCTS search. We show that our AI based *in silico* design strategy allows us to rapidly screen through the chemical space and identify dozens of new candidates that are both physically viable and outperform existing FDA and

non-FDA biomolecules in terms of their Vina scores obtained from the docking simulations. Although our workflow here is demonstrated in the context of accelerated discovery of therapeutic agents against COVID-19, the approach is much broader and can be applied generally for inverse design of chemical molecules with user-desired properties.

2 Methods

Here we perform computational molecular design and discovery by searching the space of all possible SMILES strings (see Figure 1(b)) using Monte Carlo Tree Search (MCTS). The goal of the search is to discover new ligands that strongly bind to the S-protein of the SARS-CoV-2 virus and thereby shield it from the human receptors. The strength of S-protein binding is quantified by the binding Vina scores as defined by Smith et al. [29].

The SMILES string space is represented as a tree. Our MCTS algorithm uses recurrent neural networks (RNN) during rollouts to sample the chemical space of molecules. A multi-task neural network (MTNN) model, trained to predict the binding Vina scores for a given SMILES string, is used to evaluate the merit of the nodes. A schematic overview of the search algorithm is presented in Figure 1(a). The following subsections describes the components of our workflow in detail.

2.1 Monte Carlo Tree Search

We define our search domain as the space of all the possible combinations of "building blocks" of SMILES corresponding to the atoms, number of bonds, rings and branching. The "building blocks" are identified by gathering a list of unique and indivisible chemical units present within binding-DB [10] and ZINC [30] database. Our search space of SMILES strings contains 174 unique building blocks (see supporting information S4). We use the MCTS algorithm within the *ChemTS* [37] python library for molecular generation to sample from this search space. Each node of the MCTS tree contains a "building block" of SMILES string and a upper confidence bound (UCB) score defined as

$$UCB_i = \frac{w_i}{v_i} + C \times \sqrt{\frac{2 \ln v_{parent}}{v_i}} \tag{1}$$

where v_i is the number of visits to the node i. The constant C determines the balance between the exploration and exploitation of the tree. The merit of a node, w_i, is calculated as

$$w_i = \sum_{j \in \ child_i} \frac{0.8 \times Score_j}{1 + |0.8 \times Score_j|} \tag{2}$$

The head node and the terminal node contain dummy characters '&' and '$' respectively. Each branch of the tree corresponds to a unique SMILES string representation. The complete SMILES string is built by traversing the tree from the head node to the terminal nodes. Figure 1(a) illustrates the selection, expansion, playout and update scores steps of the MCTS. ChemTS [37] uses a RNN model during the playouts to sequentially determine the subsequent nodes until the terminal node is reached. For example, if the playout is performed at a node depth of d, the RNN uses a partial SMILES string $S_{partial} = s_1, ...s_d$ obtained by traversing the path from the head node to compute a probability distribution for the subsequent building block s_{d+1}. The SMILES string is then extended by sampling from the probability distribution and a new probability distribution for s_{d+2} is computed using the extended partial SMILES string $S_{partial} = s_1, ...s_d, s_{d+1}$. The complete SMILES string is constructed by iteratively extending until the terminal symbol is obtained. At the end of the playout, we obtain a list of complete SMILES string. We eliminate the ones which does not satisfy the SMILES grammar before computing the binding energies using the MTNN model. The node scores are computed as

$$Score_i = LP(S_i) + RP(S_i) + BVS(S_i) \tag{3}$$

where S_i is the complete SMILES string after playout, $BVS(S_i)$ is the S-protein binding Vina score predicted by the MTNN, $LP(S_i)$ is a penalty term for large $\log P$ values computed using Eq.4, $RP(S_i)$ is a penalty term for large number of rings computed using Eq.5.

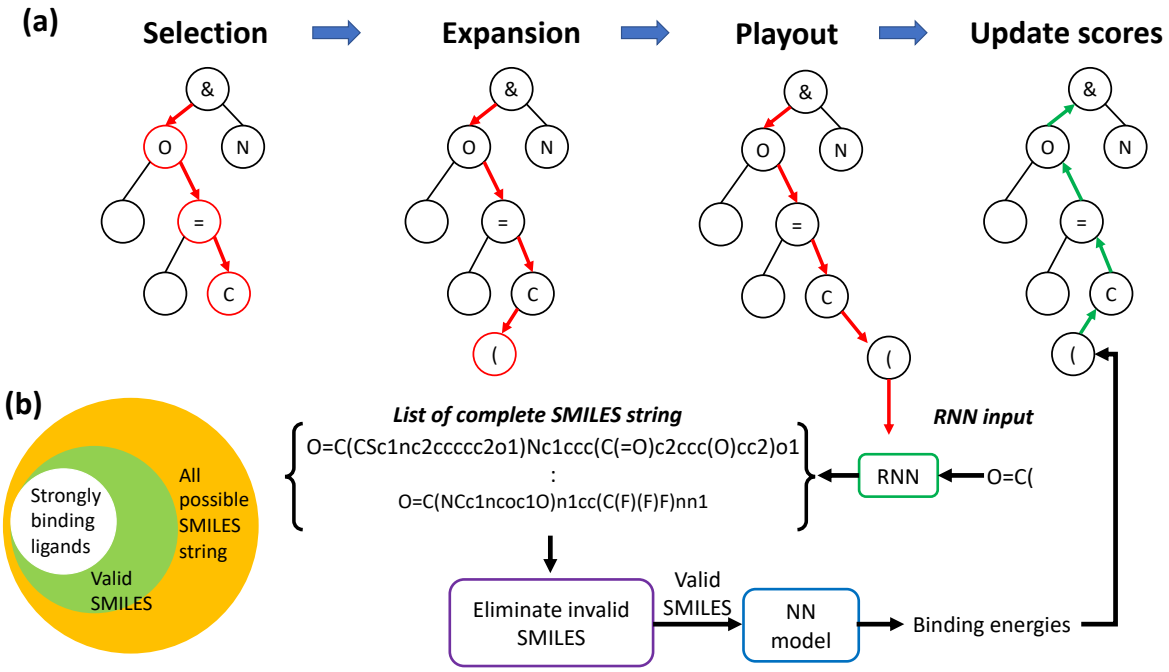

Figure 1: (a) Schematic illustration of the MCTS algorithm. The head node contains a dummy character &. A recurrent neural network model is used to complete the partial SMILES string, obtained by traversing the tree from the head node to the leaf node selected. S-protein binding Vina scores for the valid SMILES strings are predicted using a multi-task neural network (MTNN) model described in Figure 2, (b) A Venn diagram representation of phase space of all the possible SMILES, valid SMILES and the ligands with high binding energy

$$LP(S_i) = \begin{cases} \frac{-\log P(S_i)+4}{<\log P>}, & \text{if } \log P(S_i) \geq 4 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

$$RP(S_i) = \frac{NR - NR_{avg}}{NR_{std}}. \tag{5}$$

Here, $< logP >$ and NR_{avg} are the mean $\log P$ and the number of rings of the molecules in the bindingDB [10] and ZINC database [30]. NR_{std} is the standard deviation of the the number of rings.

The final step of the MCTS cycle involves updating the UCB scores of all the nodes in the tree. The MCTS cycle of selection, expansion, playouts and updating the scores is continued until a desired stopping criteria is met.

2.2 Recurrent Neural network for SMILES

The RNN maps an input string $s_1, ...s_d$ to a probability distribution $P(y_{d+1} = j) = g_j(h_d)$, where $h_{d+1} = f(h_d, x_d)$, g_j is a softmax activation function and x_d is a one-hot coded vector of length 175 representing the input s_d. The architecture of the RNN is same as the one used in the original ChemTS [37] framework where two gated recurrent units (GRU) with a 256 dimensional hidden state are stacked on top of each other and represents the function f. .

The training dataset consist of the 250k molecules from ZINC database [30] used in original ChemTS *and* an additional 800k molecules from bindingDB database [10]. The training of the RNN is performed by minimizing the following loss function

$$\min_{\theta} \sum_{i=1}^{N} \sum_{d=1}^{L} D(x_d^i, P(y_d)) \tag{6}$$

where D is the relative entropy, x^i is the ith SMILES string in the training data and θ represents the network parameters. The training is performed using ADAM[18] optimizer with a batch size of 512.

2.3 Neural network model for binding energy

A multi-task neural network (MTNN) was used as a surrogate to quickly predict Vina scores of the sampled candidate SMILES strings during the MCTS search. The architecture of the MTNN model is shown in Figure 2(a). The input layer consists of 358-dimensional molecular descriptor, followed by a series of dense layers (with/without dropout), diverging into two branches, one for the isolated S-protein and the other for the interface Vina score predictions. The molecular descriptor was obtained based on our past experience on fingerprinting organic materials, including polymers [17]. Our fingerprinting scheme uses the SMILES representation of the molecule to capture its chemical information at multiple length-scales, i.e., atomic, block-level and at a slightly higher morphological level. More details on the descriptor definition can be obtained elsewhere [17]. Superior performance of multi-task networks over their single-task counterparts for several different classes of supervised learning problems motivates the use of such network architecture here [4]. The MTNN model, as implemented in Tensorflow [1], was trained using the dataset from Smith et. al [29], which consists of Vina scores for the S-protein and the S-protein:ACE2 interface systems for nearly 8000 molecules computed using docking simulations. The loss function consisted of the sum of the mean absolute errors of the S-protein and the interface scores, while the root mean square error (RMSE) and correlation coefficient (R^2) were computed to measure model performance. To avoid overfitting, the MTNN model used dropout layers. Further, the number of training epochs was determined by tracking the loss function on the validation set (20 % of the Smith dataset). The ADAM optimizer [18] was used to train the model.

We note that the Vina scores for many molecules within the Smith dataset were reported to be extremely high (as much as 1,000,000 kcal/mol), while those with favorable binding energetics have Vina scores roughly between -7 to 0 kcal/mol. In order to remove such skewness in the data and train the MTNN model geared towards identifying favorable molecules with lower Vina scores, data points with scores greater than 5 kcal/mol where excluded during the training (by masking their contribution to the cost function). Further for a few cases, the SMILES representation could not be resolved correctly and were filtered out. Overall, the MTNN model when trained on 80 % of the Smith dataset, resulted in MAE of 0.21 and 0.81 kcal/mol in Vina scores of the S-protein and the Interface systems. Parity plots demonstrating the learning trends for both the training and the validation set are included in the Supporting Information (Figure S1).

2.4 Transfer learning to improve network performance

While good overall accuracy was achieved using the MTNN model when trained on the Smith dataset, the model performance for extreme cases that are more relevant to this work, i.e., ligands with very low S-protein or Interface Vina scores, was low. This is a known issue with surrogate models, whose loss function is based on the overall data distribution, rather than a specific/extreme part of it. In order to improve the MTNN performance, particularly to reliably identify cases with extremely low Vina scores, we retrained our model using a transfer learning approach. During the first round of MCTS search we identified 170 molecules predicted to have low Vina scores. Vina scores from the docking simulations for these 170 screened candidates were combined with 510 randomly selected cases from the Smith dataset to obtain the 'retraining' dataset. With the MTNN model weights initialized from the previous training (using Smith dataset), the model was retrained to reduce error on the new retraining dataset. Again, error on the validation set (80 % of the retraining dataset) was used to track the number of epochs, with the results for the retrained MTNN model on the training and the validation set shown in Figure 2(b). An overall MAE of 0.70 and 0.27 kcal/mol respectively for the Interface and the S-protein vina scores on the validation set denotes the good accuracy achieved by the MTNN model, which is comparable to the 1 kcal/mol accuracy expected from the reference docking simulations. The MTNN architecture was kept fixed during the retraining step.

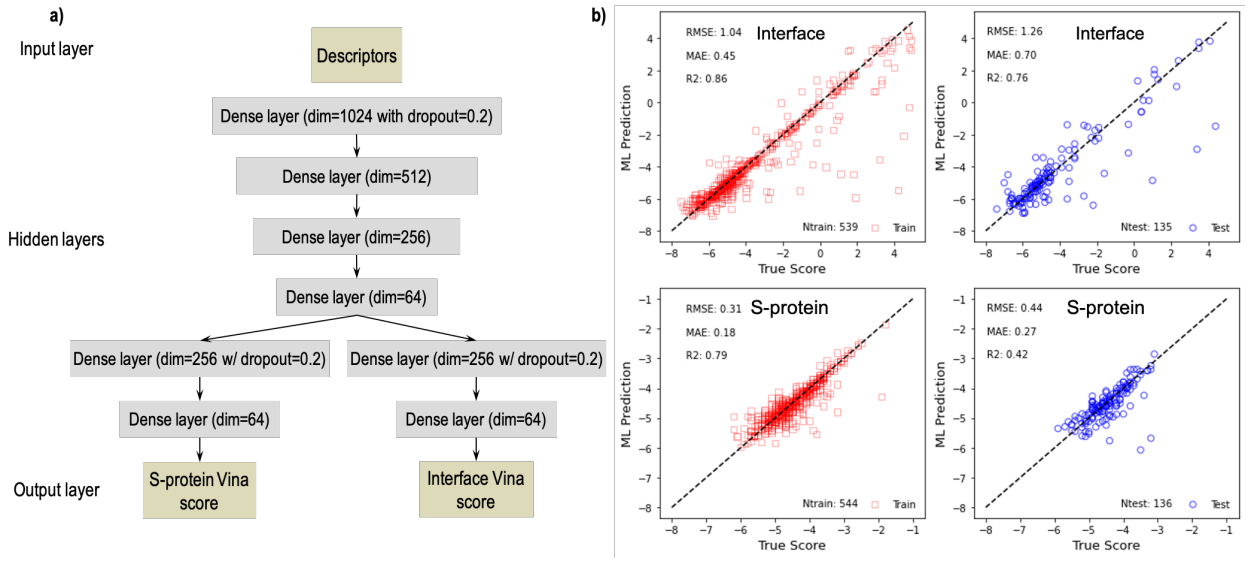

Figure 2: (a) Multi-task NN architecture to predict Vina scores of different molecules. (b) Parity plot of the S-protein and interface ML models for the training and the test set, demonstrating the good prediction accuracy achieved by both the ML models. RMSE: root mean square error, MAE: mean absolute error, R2: correlation coefficient.

2.5 Docking simulations

To validate the candidates obtained from our ML models, we perform ∼600 docking calculations, using the Autodock Vina software [33], for a diverse range of molecules with large spread in the predicted binding affinity values. Similar to the work of Smith et al. [29], we estimate the binding affinities of molecular ligands in the binding pocket region of SARS-CoV-2 S-protein:ACE2 complex (Figure 6(b)) based on the Vina score, a hybrid empirical and knowledge-based scoring function that ranks molecular conformations and estimates the free energy of binding based on inter-molecular contributions, *e.g.*, steric, hydrophobic, and hydrogen bonding, etc. [33]. As such, the Vina score can be used as a proxy that correlates to the binding affinity between molecule ligands and the SARS-CoV-2 S-protein:ACE2 receptor.

Structure of the ACE2 receptor has a protein data bank ID PDB:2AJF whereas structure of the SARS-CoV-2 S-protein has a NCBI Reference Sequence YP_009724390.1, which has the necessary mutations from its predecessor SARS variety SARS-CoV, namely at L(455), F(486), Q(493), S (494), and N(501), respectively. A total of twelve docking receptors that include six conformations of SARS-CoV-2 S-protein:ACE2 interface and the corresponding isolated SARS-CoV-2 S-protein receptors, *i.e.*, with the ACE2 receptor removed, are used in our docking calculations. These docking receptor conformations are obtained from Smith et al. [29], which were originally sampled using root mean squared displacement (RMSD) based clustering from 1.3 microsecond long all-atom Temperature Replica Exchange GROMACS simulations of the SARS-CoV-2 S-protein:ACE2 complex in water. Details regarding the construction and modeling of the complex are described here [29].

Structure of the docking ligands is converted from SMILES string using the Open Babel software [21]. As in the work of Smith et al. [29], we define a 1.2 nm × 1.2 nm × 1.2 nm search space in our docking calculation setup, which encompasses the binding pocket located at the S-protein:ACE2 interface. The same search space is explored for the isolated SARS-CoV-2 S-protein receptor cases. In our docking procedure, we find the top 10 optimized docking configurations for each molecule candidate and pick the best-scoring configuration as the predicted Vina score. Furthermore, we rank-order top candidates from our MTNN model predictions using the best docking scores of each molecule candidate, *i.e.*, the best interface score among the six interface receptor conformations and the best S-protein score among the six isolated S-protein receptor conformations.

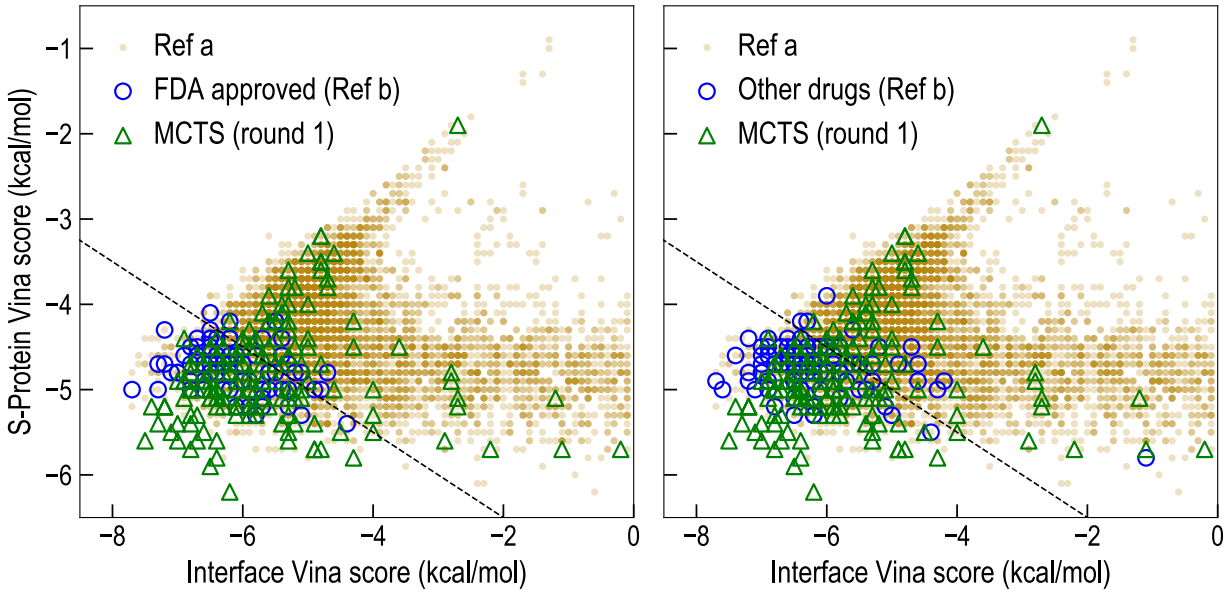

Figure 3: Vina scores for the isolated S-protein and S-protein:ACE2 interface using docking calculations. The 170 candidates selected after first round of RNN+NN models are shown in green, with a few superior candidates displaying low Vina scores. For comparison, previously considered candidates from past works (Ref a [29], Ref b [2]) are also included.

3 Results

Our goal is to search the vast chemical space of therapeutic agents and identify biomolecules that have high binding affinities to the isolated S-protein or the interface between ACE2 receptor and S-protein. To achieve this, we search for the region in the SMILES string space with low S-protein binding energies (Figure 1(b)) using the MCTS algorithm and a surrogate MTNN model as described in Section 2. Our workflow discovered many new molecules, not present in the existing databases and have a high affinity towards the S-protein of the virus. In total, we identified 97,973 unique molecules and their performance, as predicted by the surrogate MTNN model, are shown in the supporting information (Figure S4). A large fraction of these newly discovered molecules perform better i.e. have better vina scores compared to top FDA molecules sampled in Ref [2].

In order to validate the MTNN predictions on the sampled space of SMILES strings, we perform docking simulations on the molecules whose S-Protein and Interface vina scores are both less than -5.8 kcal/mol. We chose this criteria so that the number of more expensive docking simulations are kept in the order of few hundred while the molecules with high binding affinities, as predicted by MTNN, are also included in the docking simulation set. The Vina scores computed using docking simulations for the 177 molecules that satisfies the above criteria are compared against the FDA approved drugs and bindingDB dataset in Figure 3. Our strategy of using a surrogate MTNN model to quickly screen, and steer the MCTS search algorithm towards a promising region in the SMILES string space has successfully identified molecules with better binding affinity than the existing ones (Figure 3). However, we note that the mean absolute error between the MTNN model predictions and the docking simulations are 1.19 kcal/mole and 2.38 kcal/mol for the S-protein and interface vina scores respectively (see supporting information, Figure S2). Such high errors in the MTNN predictions are expected since the sampled molecules are not representative of the data distribution that was used to train the model, but instead lie in a region with extreme Vina scores. In other words, the MTNN model is being perhaps utilized in the extrapolative regime for these new molecules, and its accuracy can be improved using active learning (or retraining).

To improve the MTNN performance and thereby the efficiency of MCTS search, we retrain the surrogate MTNN model by including the newly discovered molecules in the training data. We use a transfer learning

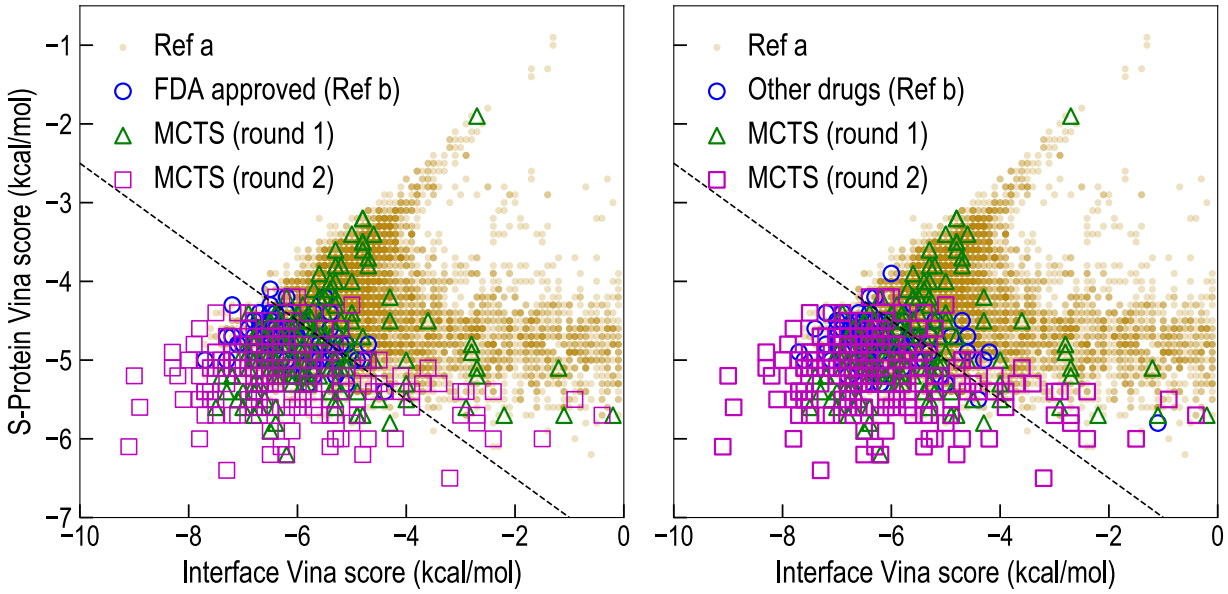

Figure 4: Vina scores for the isolated S-protein and S-protein:ACE2 interface using docking calculations. The 200 candidates selected after second round of RNN+MTNN models are shown in green, with a few superior candidates displaying low Vina scores. For comparison, previously considered candidates from past works (Ref a [Smith work], Ref b [2]) are also included.

approach as detailed in section 2.4. Extending the training data beyond the distribution of initial data set allows the MTNN model to learn correlation between the SMILES strings and Vina scores over a larger domain of SMILES space. We next repeat the MCTS search with the improved MTNN model and now identify a significantly larger number of unique molecules ($\sim$ 300,000) compared to the previous search. The distribution of the MTNN vina scores for these molecules are shown in supporting information (Figure S5). The best molecules were selected by sorting the points by its distance from the line $\mathrm{S{-}Protein}_{vina} + \mathrm{Interface}_{vina} \leq -7.5$ kcal/mol, which is the same screening criteria used in [2], and selecting the 200 farthest molecules. The Vina scores computed from the docking simulation for the 200 best candidates sampled using the retrained MTNN model are compared with the results of the sampling from the first MCTS-MTNN run as well as FDA approved drugs from the CureFFI database [8] and a dataset of common active ingredients from DrugCentral [35] (see Figure 4). The mean absolute error in the MTNN predictions are 0.38 kcal/mol and 1.28 kcal/mol for the S-protein and interface vina scores respectively, which is less than the original MTNN model. Sampling with an improved MTNN model which can correctly bias MCTS algorithm to search the promising regions of the SMILES string space thus enables the design of better performing ligands (purple markers in Figure 4).

We note that the MTNN model has learnt better the correlation between the SMILES string and the binding energies by extending the dataset to include new on-the-fly sampled data and performing the transfer learning. In principle, we can iteratively improve MTNN model by including the diverse molecules sampled by MCTS during transfer learning. The advantage of iterative sampling during any given search cycle is twofold: (i) the MCTS designed molecules perform better with each iteration; (ii) the surrogate MTNN model to predict the binding energies get progressively better with each iteration since it learns the correlation over a larger region in the search space. Our work here demonstrates the potential for iterative learning and exploration strategy to design new molecules as well as concurrently improve the accuracy of the machine learning models. In principle, this search and improve strategy can be broadly applied to discover and inverse design molecules for any user-desired target properties.

Although the goal in this work is to discover molecules with high binding affinities towards the S-protein of the SARS-Cov-2 virus, one can also utilize other metrics related to the druglikeness of the compounds to further determine the viability of drug. Here, we evaluate the druglikeness of the identified molecules using

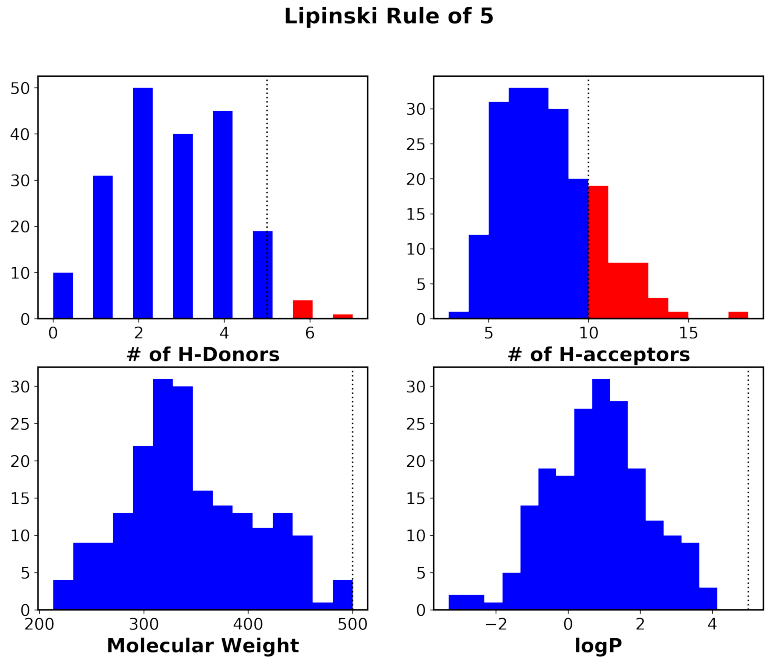

Figure 5: Histogram of the Lipinski attributes for the top 200 molecules discovered by the MCTS algorithm. 134 out of the 200 molecules satisfy all the 4 criteria

the Lipinski's rule of 5 which requires the following criteria to be met:

- Number of H-bonds < 5
- Number of H-acceptors < 10
- Molecular weight < 500 daltons
- Octanol-water partition coefficient $(\log P) < 5$

Using these above metrics, we further screen the candidates identified by our MCTS-NN search. The distribution of the Lipinski attributes for the top 200 molecules are shown in Figure 5. The S-protein Vina scores of the 200 molecules are provided in Table S1 in supporting information. Furthermore, the table of the top candidates and their complete Lipinski metrics are provided in the supplementary files. Out of the 200 best molecules, 134 molecules satisfy the rule of 5 criteria. Some of the top ranking candidate molecules from this list are depicted in Fig. 6.

4 Discussion

In this work, the therapeutic prowess of the molecules is determined by their vina scores which is a measure of the binding energies. The binding energies of the molecules are dictated by the strength of the chemical interaction at the S-protein of the virus as well as the interface formed by the S-protein and human receptors. We aim to identify the common chemical fragments and patterns within set of strongly binding molecules to gain insights into the chemical features which favors binding. Knowledge of such chemical fragments which make-up the top candidates will allow accelerated design of therapeutic agents against COVID-19.

First, we analyze the frequency of occurrence of chemical features by splitting each molecule into smaller fragments and counting the number of unique fragments. We use the molecular fragmenting algorithm within RDKit[23] to break the molecules across the single bonds and building a catalog of unique fragments. A fragment can be a part of a bigger fragment which can contribute to further bigger ones. We represent

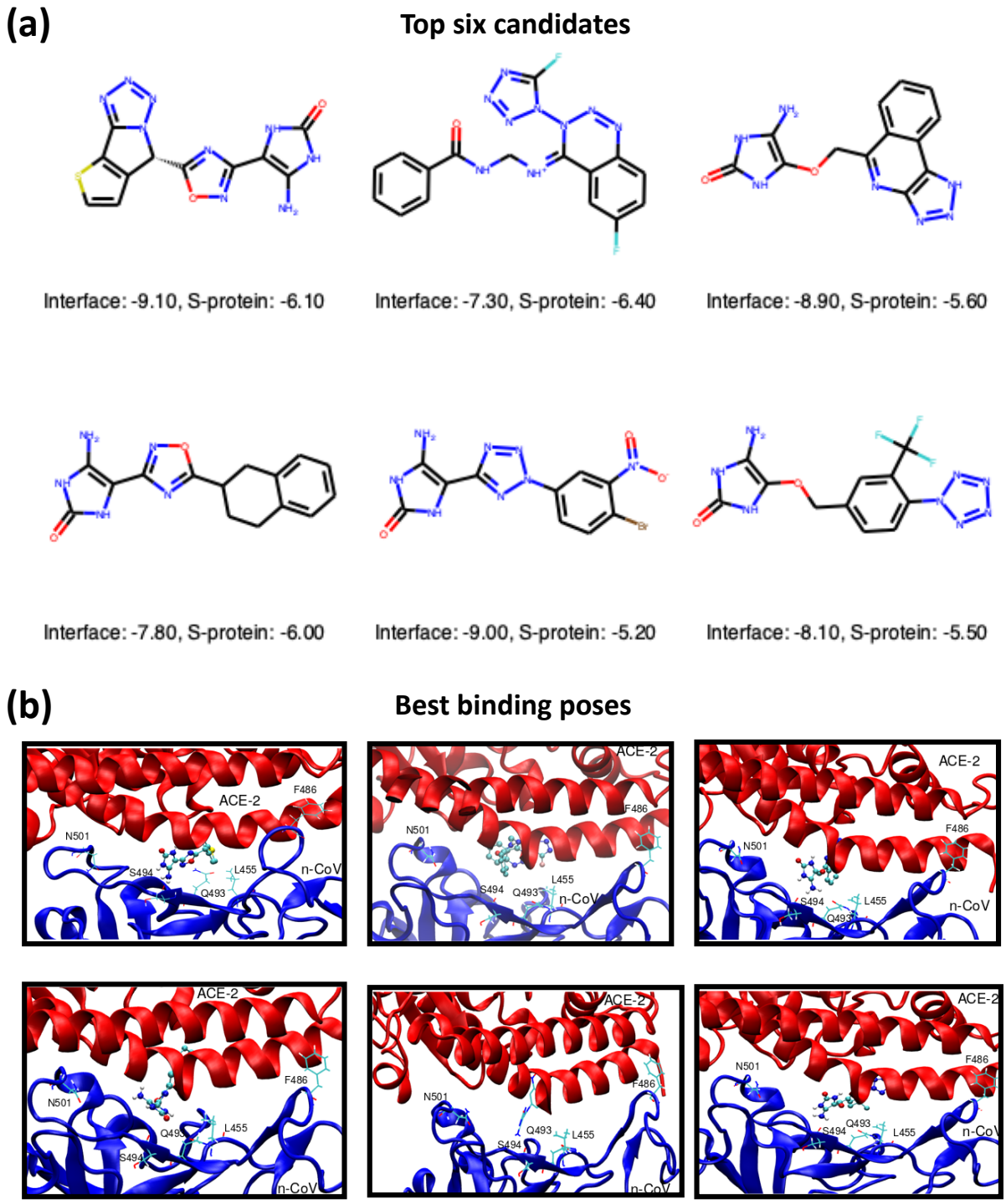

Figure 6: Top candidates identified from this work along with their Vina scores for the S-protein:ACE2 interface (labeled, interface) and the S-protein systems using the ensemble docking simulations. The bottom panel shows the best binding pose for the corresponding molecules at the interface of nCov and ACE2 receptor.

this hierarchy as a directed graph (see supporting information S6) and pick only the terminal nodes for our analysis. The frequency of the top 15% fragments within the best molecules sampled by MCTS and the best molecules identified by [2] are shown in Figure 7. The strings within the angular brackets describe the functional group attached to the fragment. The fragments within the MCTS sampled molecules are predominantly different from the molecules identified from the FDA and Drug Central data base in Ref [2]. Hence, the superior performance of the molecules identified in this work can be partly attributed to the presence of unique fragments favoring strong binding.

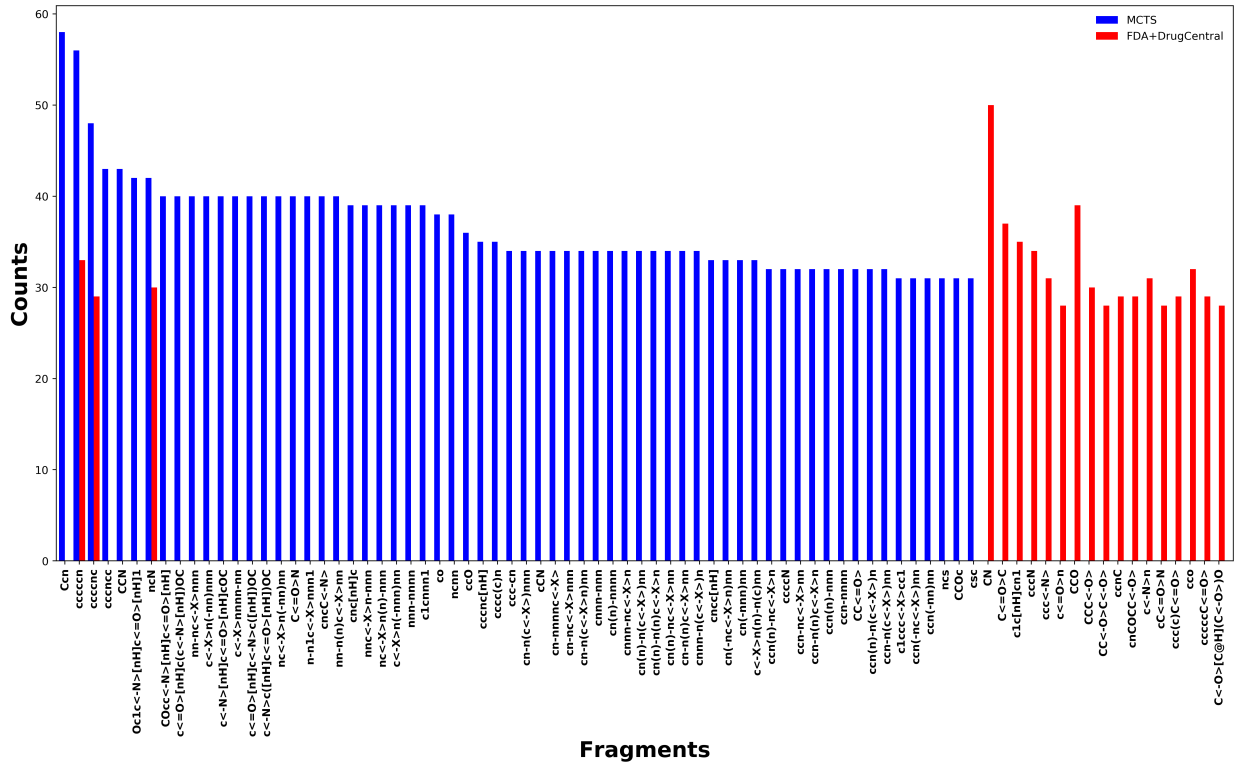

Figure 7: Counts for a given fragment within the 200 best performing molecules sampled by MCTS and the 200 best performing molecules identified from the FDA approved drugs and the DrugCentral database. Only the fragments which are present in more than 15% of the molecules are shown here. The functional group contained within the angular brackets are attached to the fragments

Based on the fragment distribution, we find that the top candidates from our search tend to include fluoro-alkyl or fluoro-aromatic groups in their molecular makeup. Fluorinated derivatives and substitution are known to play a unique role in medicinal chemistry. The tailored incorporation of a fluoro-alkyl or fluoro-aromatic group (Enamine also having cyclic amines) into bioactive compounds influences the binding affinity, stability, lipophilicity, membrane permeability and bioactivity [13]. Examples including some of our top compounds such as ligand 2 and ligand 6 consists of fluoromethyl group (CF3) which are known to improve adsorption digestion metabolism and extraction (ADME) and pharmaco-kinetic properties [13, 32], see Figure 6a. It is worth noting that heterocycles and fused rings constitute of about 70 % of the drug market with one of the reasons being the limited entropic effects showcased by their molecular rigidity. Nitrogen forms a key atom in such fused rings and heterocycles with a large number of ligands consisting of the azole moiety. Analysis of some of the top fragments for the 200 best performing molecules indicate the presence of unsaturated amines (eg. ccccn ccccnc) as seen from Figure 7. This suggests the ubiquity of enamine and azole chemistry in drug design - such moieties have had considerable success as anti-fungal agents. To gain more insights into the performance of the top ranked candidates, we closely analyze our autodocking simulations and evaluate the binding pose of these candidates on the interface of ACE-2 receptor and the virus. We observe that the exocylic nitrogen (NH2) amino group of the cyclic lactum forms hydrogen bond

interactions with the serine (S494) amino acid of the n-CoV. In the case of another top-ranked molecule, the benzene moiety interacts with the oxygen of the serine amounting to about -4 to -5 kcal/mol energetic interaction. Based on these observations, we postulate the role of amino-azole group with the serine residue (S494 one of the five mutating sites from the SARS-CoV 2002 virus), see Figure 6b, to be an influential factor in determining the efficacy of the ligands.

5 Conclusion

We present an AI search strategy (decision trees such as Monte Carlo Tree Search) to discover and design of therapeutic agents against COVID-19 virus. We replace the expensive docking simulations with a surrogate multi-task-neural-network (MTNN) model to accelerate the search of the vast chemical space represented by SMILES strings. The MTNN model is trained against an extensive dataset of therapeutic agents and their binding affinities obtained from docking simulations from available literature. The MCTS search includes an RNN model that is used during the playouts to sequentially determine the subsequent nodes until the terminal node is reached. We hypothesize that the therapeutic prowess against SARS-CoV-2 virus is measured by the binding affinity (Vina scores) of the biomolecules to either the isolated S-protein of SARS-CoV-2 at its host receptor region or to the S-protein:ACE2 receptor interface. These Vina scores, evaluated using a MTNN surrogate model, are used as target objectives in the MCTS-RNN search of the chemical space. We perform transfer learning to retrain the MTNN surrogate model as the AI search identifies new molecules that are beyond those included in the original training dataset. This iterative search and retrain strategy is shown to significantly accelerate the discovery- we predict over a 100 new biomolecules that outperform existing FDA (several 100's) and non-FDA (million) molecules in their therapeutic prowess measured via Vina scores. We perform detailed analysis using other metrics related to the druglikeness of the compounds to further determine the viability of these agents. Finally, we also analyze structural features and attempt to identify structural similarities between top performing candidates. We note that our molecular discovery approach using MCTS and RNN is a more general technique which can be applied to design chemical molecules with desired property as long we have a reasonably fast and accurate surrogate model to screen the SMILES string space and guide the search algorithm to towards promising region. In addition, the strategy to iteratively explore the design space and concurrently improve the accuracy of the model, can open a range of possibilities with regard to molecular design and discovery of therapeutic agents.

Acknowledgements

We acknowledge funding from BES Award DE-SC0020201 by DOE to support this research. The use of the Center for Nanoscale Materials, an Office of Science user facility, was supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Contract No. DE-AC02- 06CH11357. This research used resources of the National Energy Research Scientific Computing Center, which was supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. An award of computer time was provided by the Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program of the Argonne Leadership Computing Facility at the Argonne National Laboratory, which was supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-06CH11357. SKRS acknowledges UIC start-up funds for supporting this research.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

[2] R. Batra, H. Chan, G. Kamath, R. Ramprasad, M. J. Cherukara, and S. Sankaranarayanan. Screening of Therapeutic Agents for COVID-19 using Machine Learning and Ensemble Docking Simulations. *arXiv e-prints*, art. arXiv:2004.03766, Apr. 2020.

[3] N. Bung, S. R. Krishnan, G. Bulusu, and A. Roy. De novo design of new chemical entities (nces) for sars-cov-2 using artificial intelligence, Mar 2020. URL https://chemrxiv.org/articles/De_Novo_Design_of_New_Chemical_Entities_NCEs_for_SARS-CoV-2_Using_Artificial_Intelligence/11998347/1.

[4] R. Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

[5] C. Cavasotto and J. D. Filippo. In silico drug repurposing for covid-19: Targeting sars-cov-2 proteins through docking and quantum mechanical scoring, Apr 2020. URL https://chemrxiv.org/articles/In_silico_Drug_Repurposing_for_COVID-19_Targeting_SARS-CoV-2_Proteins_through_Docking_and_Quantum_Mechanical_Scoring/12110199/2.

[6] S. Chakraborti, S. Bheemireddy, and N. Srinivasan. Repurposing drugs against main protease of sars-cov-2: mechanism based insights supported by available laboratory and clinical data, Apr 2020. URL https://chemrxiv.org/articles/Drug_Repurposing_Approach_Targeted_Against_Main_Protease_of_SARS-CoV-2_Exploiting_Neighbourhood_Behaviour_in_3D_Protein_Structural_Space_and_2D_Chemical_Space_of_Small_Molecules/12057846/2.

[7] V. Chenthamarakshan, P. Das, I. Padhi, H. Strobelt, K. W. Lim, B. Hoover, S. C. Hoffman, and A. Mojsilovic. Target-Specific and Selective Drug Design for COVID-19 Using Deep Generative Models. *arXiv e-prints*, art. arXiv:2004.01215, Apr. 2020.

[8] CureFFI. https://www.cureffi.org/2013/10/04/list-of-fda-approved-drugs-and-cns-drugs-with-smiles/.

[9] A. D. Elmezayen, A. Al-Obaidi, A. T. Şahin, and K. Yelekçi. Drug repurposing for coronavirus (covid-19): in silico screening of known drugs against coronavirus 3cl hydrolase and protease enzymes. *Journal of Biomolecular Structure and Dynamics*, 0(0):1–13, 2020. doi: 10.1080/07391102.2020.1758791. URL https://doi.org/10.1080/07391102.2020.1758791. PMID: 32306862.

[10] M. K. Gilson, T. Liu, M. Baitaluk, G. Nicola, L. Hwang, and J. Chong. BindingDB in 2015: A public database for medicinal chemistry, computational chemistry and systems pharmacology. *Nucleic Acids Research*, 44(D1):D1045–D1053, 10 2015. ISSN 0305-1048. doi: 10.1093/nar/gkv1072. URL https://doi.org/10.1093/nar/gkv1072.

[11] A. Gupta, A. T. Müller, B. J. H. Huisman, J. A. Fuchs, P. Schneider, and G. Schneider. Generative recurrent networks for de novo drug design. *Molecular Informatics*, 37(1-2):1700111, 2018. doi: 10.1002/minf.201700111. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/minf.201700111.

[12] R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 4(2):268–276, 2018. doi: 10.1021/acscentsci.7b00572. URL https://doi.org/10.1021/acscentsci.7b00572. PMID: 29532027.

[13] K. C. Howard, E. K. Dennis, D. S. Watt, and S. Garneau-Tsodikova. A comprehensive overview of the medicinal chemistry of antifungal drugs: perspectives and promise. *Chem. Soc. Rev.*, 49:2426–2480, 2020. doi: 10.1039/C9CS00556K. URL http://dx.doi.org/10.1039/C9CS00556K.

[14] H. Ikebata, K. Hongo, T. Isomura, R. Maezono, and R. Yoshida. Bayesian molecular design with a chemical language model. *Journal of Computer-Aided Molecular Design*, 31(4):379–391, 2017. doi: 10.1007/s10822-016-0008-z. URL https://doi.org/10.1007/s10822-016-0008-z.

[15] S. Kajita, T. Kinjo, and T. Nishi. Autonomous molecular design by monte-carlo tree search and rapid evaluations using molecular dynamics simulations. *Communications Physics*, 3(1):77, 2020. doi: 10.1038/s42005-020-0338-y. URL https://doi.org/10.1038/s42005-020-0338-y.

[16] M. Kandeel and M. Al-Nazawi. Virtual screening and repurposing of fda approved drugs against covid-19 main protease. *Life Sciences*, 251:117627, 2020. ISSN 0024-3205. doi: https://doi.org/10.1016/j.lfs.2020.117627. URL http://www.sciencedirect.com/science/article/pii/S0024320520303751.

[17] C. Kim, A. Chandrasekaran, T. D. Huan, D. Das, and R. Ramprasad. Polymer genome: a data-powered polymer informatics platform for property predictions. *J. Phys. Chem. C*, 122(31):17575–17585, 2018.

[18] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, art. arXiv:1412.6980, Dec. 2014.

[19] S. Mahapatra, P. Nath, M. Chatterjee, N. Das, D. Kalita, P. Roy, and S. Satapathi. Repurposing therapeutics for covid-19: Rapid prediction of commercially available drugs through machine learning and docking. *medRxiv*, 2020. doi: 10.1101/2020.04.05.20054254. URL https://www.medrxiv.org/content/early/2020/04/23/2020.04.05.20054254.

[20] S. Mahdian, A. Ebrahim-Habibi, and M. Zarrabi. Drug repurposing using computational methods to identify therapeutic options for covid-19. *Journal of Diabetes & Metabolic Disorders*, 2020. doi: 10.1007/s40200-020-00546-9. URL https://doi.org/10.1007/s40200-020-00546-9.

[21] N. M. O'Boyle, M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison. Open Babel: An open chemical toolbox. *Journal of cheminformatics*, 3(1):33, 2011.

[22] J. M. Parks and J. C. Smith. How to discover antiviral drugs quickly. *New England Journal of Medicine*, 0(0):null, 0. doi: 10.1056/NEJMcibr2007042. URL https://doi.org/10.1056/NEJMcibr2007042.

[23] RDKit, online. RDKit: Open-source cheminformatics. http://www.rdkit.org. [Online; accessed 11-April-2013].

[24] J.-L. Reymond. The chemical space project. *Accounts of Chemical Research*, 48(3):722–730, 2015. doi: 10.1021/ar500432k. URL https://doi.org/10.1021/ar500432k. PMID: 25687211.

[25] G. Schneider and U. Fechner. Computer-based de novo design of drug-like molecules. *Nature Reviews Drug Discovery*, 4(8):649–663, 2005. doi: 10.1038/nrd1799. URL https://doi.org/10.1038/nrd1799.

[26] M. H. S. Segler, T. Kogej, C. Tyrchan, and M. P. Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4(1):120–131, 2018. doi: 10.1021/acscentsci.7b00512. URL https://doi.org/10.1021/acscentsci.7b00512. PMID: 29392184.

[27] M. H. S. Segler, M. Preuss, and M. P. Waller. Planning chemical syntheses with deep neural networks and symbolic ai. *Nature*, 555(7698):604–610, 2018. doi: 10.1038/nature25978. URL https://doi.org/10.1038/nature25978.

[28] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018. ISSN 0036-8075. doi: 10.1126/science.aar6404. URL https://science.sciencemag.org/content/362/6419/1140.

[29] M. Smith and J. C. Smith. Repurposing therapeutics for COVID-19: Supercomputer-based docking to the SARS-CoV-2 viral spike protein and viral spike protein-human ACE2 interface. 2020.

[30] T. Sterling and J. J. Irwin. Zinc 15 – ligand discovery for everyone. *Journal of Chemical Information and Modeling*, 55(11):2324–2337, 2015. doi: 10.1021/acs.jcim.5b00559. URL https://doi.org/10.1021/acs.jcim.5b00559. PMID: 26479676.

[31] N. Ståhl, G. Falkman, A. Karlsson, G. Mathiason, and J. Boström. Deep reinforcement learning for multiparameter optimization in de novo drug design. *Journal of Chemical Information and Modeling*, 59(7):3166–3176, 2019. doi: 10.1021/acs.jcim.9b00325. URL https://doi.org/10.1021/acs.jcim.9b00325. PMID: 31273995.

[32] N. Thamban Chandrika, S. K. Shrestha, H. X. Ngo, O. V. Tsodikov, K. C. Howard, and S. Garneau-Tsodikova. Alkylated piperazines and piperazine-azole hybrids as antifungal agents. *Journal of Medicinal Chemistry*, 61(1):158–173, 2018. doi: 10.1021/acs.jmedchem.7b01138. URL https://doi.org/10.1021/acs.jmedchem.7b01138. PMID: 29256601.

[33] O. Trott and A. J. Olson. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2): 455–461, 2010.

[34] O. Ursu, J. Holmes, J. Knockel, C. G. Bologa, J. J. Yang, S. L. Mathias, S. J. Nelson, and T. I. Oprea. DrugCentral: online drug compendium. *Nucleic Acids Research*, 45(D1):D932–D939, 10 2016. ISSN 0305-1048. doi: 10.1093/nar/gkw993. URL https://doi.org/10.1093/nar/gkw993.

[35] O. Ursu, J. Holmes, J. Knockel, C. G. Bologa, J. J. Yang, S. L. Mathias, S. J. Nelson, and T. I. Oprea. DrugCentral: online drug compendium. *Nucleic acids research*, page gkw993, 2016. http://drugcentral.org.

[36] J. Wang. Fast identification of possible drug treatment of coronavirus disease-19 (covid-19) through computational drug repurposing study. *Journal of Chemical Information and Modeling*, 60(6):3277–3286, 2020. doi: 10.1021/acs.jcim.0c00179. URL https://doi.org/10.1021/acs.jcim.0c00179. PMID: 32315171.

[37] X. Yang, J. Zhang, K. Yoshizoe, K. Terayama, and K. Tsuda. Chemts: an efficient python library for de novo molecular generation. *Science and Technology of Advanced Materials*, 18(1):972–976, 2017. doi: 10.1080/14686996.2017.1401424. URL https://doi.org/10.1080/14686996.2017.1401424. PMID: 29435094.

Supporting Information for Artificial Intelligence Guided *De Novo* Molecular Design targeting COVID-19

Srilok Srinivasan[1], Rohit Batra[1], Henry Chan[1,3], Ganesh Kamath[2], Mathew J. Cherukara[1], and Subramanian Sankaranarayanan[1,3]

[1]Center for Nanoscale Materials, Argonne National Laboratory, Lemont, Illinois 60439, United States
[2]Dalzielfiver LLC, 3500 Carlfield St., El Sobrante, CA 94803, United States
[3]Department of Mechanical and Industrial Engineering, University of Illinois,Chicago, Illinois 60607, United States

June 29, 2020

1 Training and Validation of Neural Network Models

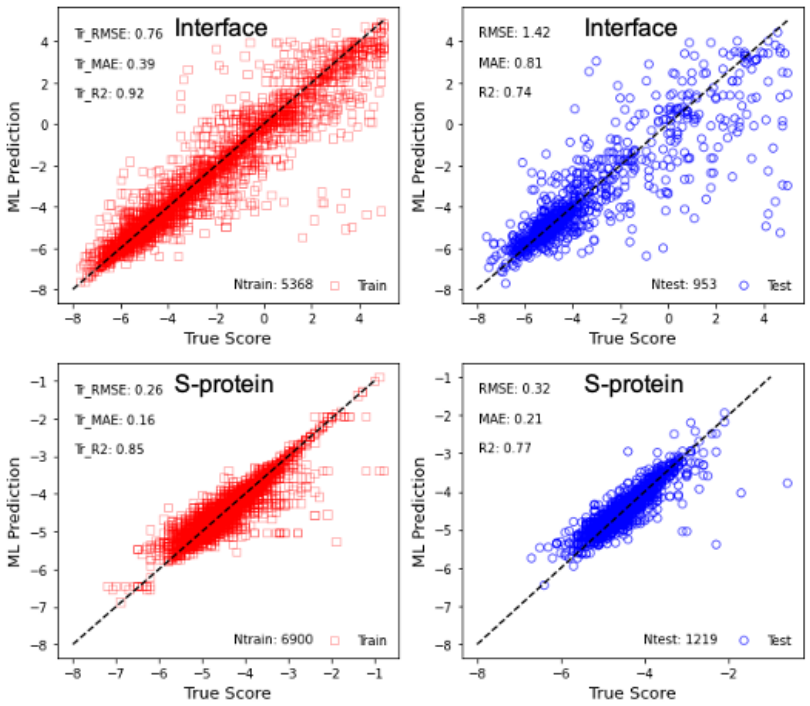

Figure 1: Parity plot of the MTNN Vina score predictions for the S-protein:ACE2 interface (top panel) and the isolated S-protein (bottom panel) systems against those obtained from the Smith dataset. Results for both the training (left panels) and the validation (right panel) set are presented.

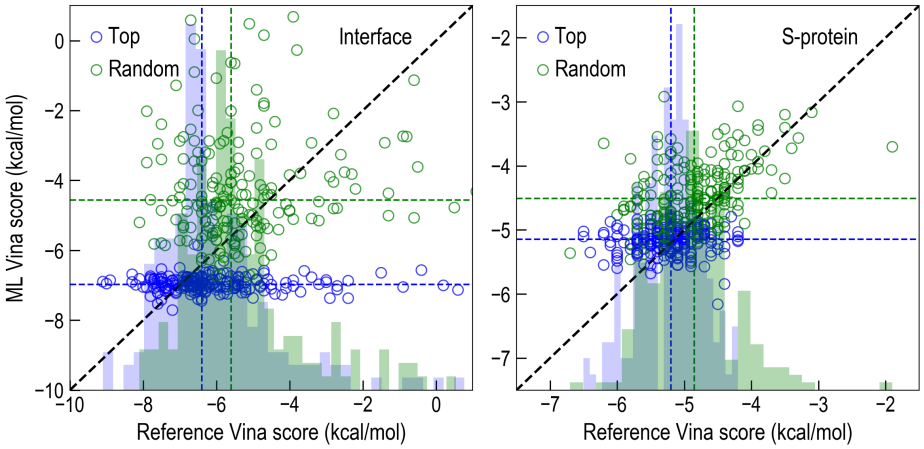

Figure 2: Parity plot of the MTNN Vina score predictions for the S-protein:ACE2 interface (left panel) and the isolated S-protein (right panel) systems against those obtained from the autodock simulations for the identified top (blue) and randomly (green) chosen candidate from the molecules sampled with MCTS using the retrained MTNN (round 2). In each panel, histograms for the top (blue) and randomly (green) chosen candidate molecules illustrate that MTNN models indeed help the MCTS to sample candidates with low Vina scores. This is further supported by the dashed horizontal and vertical lines, which denote the median MTNN prediction and Autodocking score, respectively. We note that 3 of the selected top 200 candidates had greater than 0 vina score (i.e., no binding affinity) for the S-protein:ACE2 interface complex, and are not included in this plot for better readability. These cases highlight limitations of the MTNN model employed and are available in the Supporting files.

2 Training of RNN

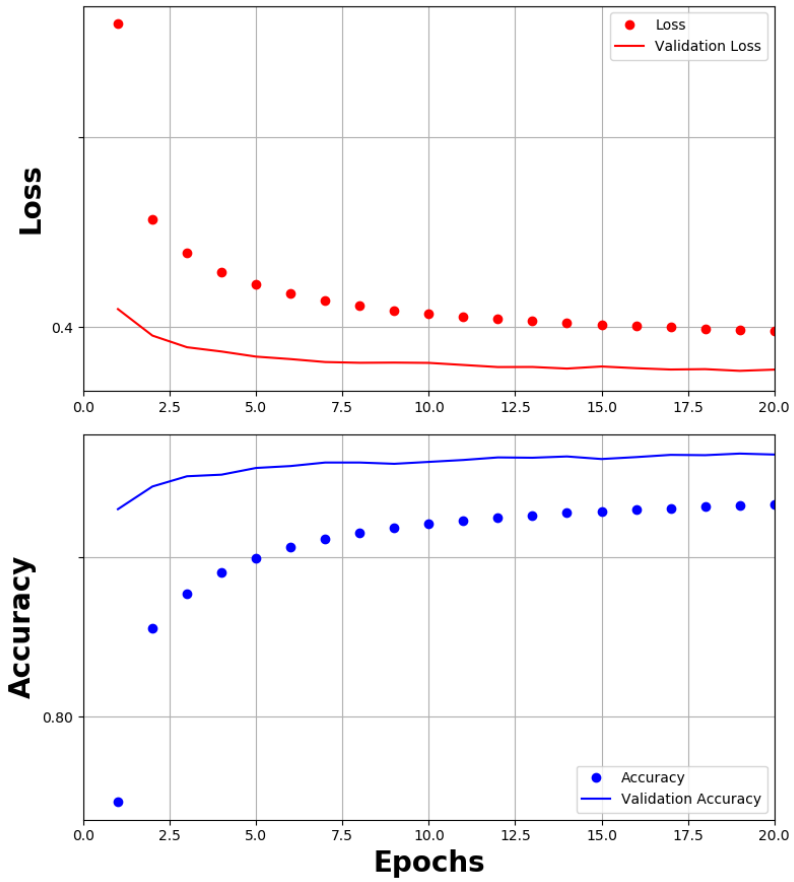

Figure 3: Loss and the accuracy of RNN during the training. The RNN was trained on 945512 SMILES string and the validation set consist of 105057 SMILES string

3 Distribution of binding Vina scores as predicted by MTNN

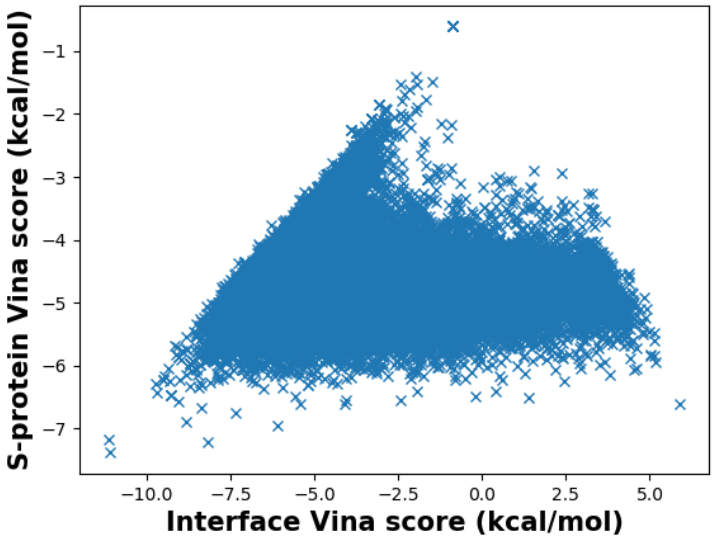

Figure 4: Distribution of the MTNN vina scores sampled by MCTS with the original MTNN (round 1)

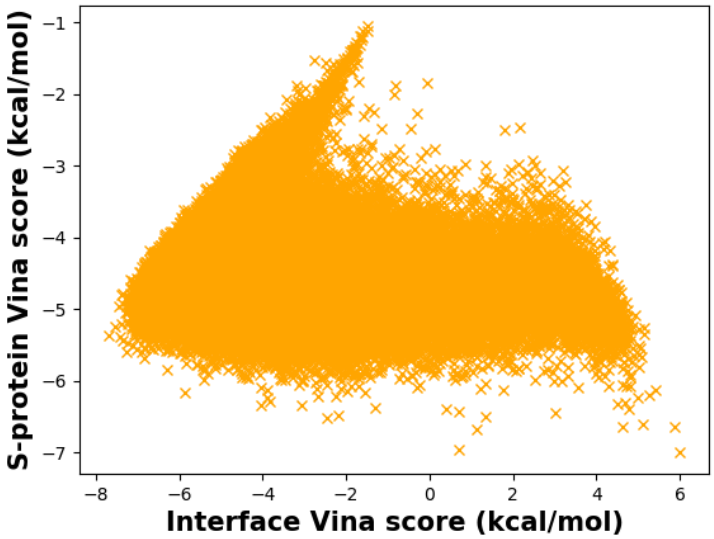

Figure 5: Distribution of the MTNN vina scores sampled by MCTS with improved MTNN (round 2)

4 List of SMILES building block for MCTS sampling

- &
- C
- O
- c
- 1
- (
-)
- 2
- [nH]
- N
- =
- 3
- -
- n
- [C@H]
- S
- [C@@H]
- 4
- Cl
- o
- F
- 5
- Br
- 6
- s
- #
- [C-]
- [N+]
- /
- \
- [Si]
- 7
- [O-]
- [C@]
- I
- [C@@]
- P
- [n+]
- .
- [S@]
- 8
- [N-]
- [NH3+]
- [NH+]
- [S+]
- [nH+]
- [SeH]
- [Br-]
- [Na+]
- [n-]
- [P@]
- [O]
- [K+]
- [Na]
- [Li+]
- [Li]
- B
- [B-]
- [OH+]
- [c+]
- [Se]
- [I-]
- [P+]
- [CH-]
- [NH-]
- [Re]
- [PH]
- [P@@]
- [S@@]
- [S@@+]
- [S@+]
- [N@+]
- [N@@+]
- [V]
- [NH2+]
- [CH+]
- [c-]
- [SH+]
- [BH2-]
- [S]
- [Zn+]
- [se]
- [S-]
- [Au]
- [F-]
- [Cl-]
- [SH-]
- [Sn]
- [Te]
- [Os]
- [Ru-]
- [Cl+3]
- [As]
- 9
- %
- 0
- [C+]
- [O+]
- [o+]
- [N@H+]
- [N@@H+]
- [s+]
- [Hg]
- [BiH3]
- [Re-]
- [SH]
- [CH]
- [cH-]
- [I]
- [SiH2]
- [C]
- *
- [2H]
- [SiH]
- [K]
- [OH-]
- [Ca]
- [Cu]
- [Mn]
- [BH-]
- [BH3-]
- [Ca+2]
- [W]
- [Se+]
- [N@]
- [N@@]
- [Se-]
- [Zn]
- [11CH3]
- [Fe+2]
- [Pd]
- [PH+]
- [Fe-2]
- [I+]
- [Ru]
- [Zn+2]
- [P-]
- [NH4+]
- [Nb-2]
- [Pt-2]
- [Fe-3]
- [Al-3]
- [Cu-]
- [Ag-]
- [Au-]
- [As-]
- [Pd-2]
- [Sb]
- [CH2-]
- [Pt]
- [Mo]
- [I+3]
- [Br+2]
- [I+2]
- [Cl+2]
- [CH2+]
- [Fe]
- [3H]
- [Co]
- [Sr+2]
- [125I]
- [te]
- [18F]
- [cH+]
- [123I]
- [Ni+2]
- [Tc]
- [Ni]
- [CH2]
- [Gd-4]
- [Gd-5]
- [SiH3]

5 Rank ordering of Lipinski metrics for top 200 MCTS sampled molecules

Table 1: Top 200 candidates sampled my MCTS with improved MTNN model and their S-protein Vina scores

SMILES	SP-Vina(kcal/mol)
N#Cc1ccc(Cl)c(ONc2cnnn2-n2nnnc2F)c1	-5.700
Nc1[nH]c(=O)[nH]c1OCc1nsc(Cn2nnc(-c3ccccc3)c2-c2ccco2)n1	-6.000
O=C(Nc1ccc2c(c1)CC(Oc1cnnn1-n1nnnc1F)C2)c1nc2ccccc2[nH]1	-5.400
Nc1[nH]c(=O)[nH]c1-c1nsc(C[C@@H]2C(=O)NCC[C@@H]2O)n1	-4.900
Nc1[nH]c(=O)[nH]c1-c1noc(NC2=N[C@H](S(=O)(=O)Cc3ccccc3)C2)n1	-5.500
Nc1[nH]c(=O)[nH]c1OCc1nc(-c2cnc(C3(C(=O)O)CCC3)c(Cl)c2)no1	-5.300
[NH2+][C@H]([NH+][C@H](c1ccco1)[C@@H](CC(=O)O)C(F)(F)F)c1ncno1	-4.900
Nc1[nH]c(=O)[nH]c1OC[C@H](CCC(=O)O)NC(=O)Nc1cc(Cl)cc(Cl)c1	-4.900
[NH2+][C@H](N[C@@H](Cc1ccncc1)c1ccccc1)c1nc2ncccc2o1	-5.000
Nc1[nH]c(=O)[nH]c1OCc1nnn(C[C@H](CC(=O)O)c2ccccc2Cl)n1	-5.500
Fc1nnnn1-n1nnc2cc(NCc3nnc[n-]3)ccc21	-5.400
C[C@@]1(c2ccco2)O[C@@H]2S[C@H](Oc3[nH]c(=O)[nH]c3N)C(=O)N21	-5.100
O=C(NCc1ncoc1O)n1cc(C(F)(F)F)nn1	-5.100
O=C(CSc1nc2ccccc2o1)Nc1ccc(C(=O)c2ccc(O)cc2)o1	-5.000
Nc1[nH]nnc1CNC(=O)C(Nc1ccccc1)c1ccccc1	-5.700
Nc1[nH]c2nc(-c3nc(O)cc(=O)o3)nnc2c1-c1ccc(NC(=O)C2CCOC2)cc1	-4.200
[NH2+][C@H](Cn1nnn(N(CCF)Cc2ccccc2)c1=O)c1ncc[nH]1	-4.900
Nc1[nH]c(=O)[nH]c1OCc1nc2nn[nH]c2c2ccccc12	-5.600
Nc1[nH]c(=O)[nH]c1-c1nsnc1OCC/C=C/CCC(=O)NO	-5.100
Nc1[nH]c(=O)[nH]c1OCC(=O)COC(=O)c1ccccc1F	-5.100
O[C@H]1C[C@](O)([C@H](c2ccsc2)n2nnnc2-n2nnnc2F)C1	-5.200
O=C(Nc1ccnc(Oc2ccnc3nnn(-n4nnnc4F)c23)c1C(F)(F)F)c1cccs1	-5.500
Nc1[nH]c(=O)[nH]c1OCc1nsc2sc3ccccc3c12	-5.200
Cn1nnn(-n2nnnc2F)c1=NNC(=O)[C@H](Cc1ccccc1)NC(=O)c1ccccc1	-5.800
O=C(NC[NH+]=c1c2cc(F)ccc2nnn1-n1nnnc1F)c1ccccc1	-6.400
[NH2+][C@H]([NH+][C@@H]1OCCCCc2c1ccn2-c1ccccc1)c1ncno1	-5.300
N#C[C@@H](COc1cnon1)NC1=C(C(=O)O)C[C@H]2CCC[C@H]12	-5.100
Nc1[nH]c(=O)[nH]c1OCc1n[nH]cc1-n1nc2ccccc2c1N1CCC[C@H](C(=O)O)C1	-5.600
Nc1[nH]c(=O)[nH]c1OC[C@H](N)CNC(=O)c1ccc(-c2ccccc2)cc1	-5.400
Fc1nnnn1-n1nnc2cc(NCc3cc(-c4ccccc4)[nH]c3-c3ccccc3)ccc21	-6.000
[NH2+][C@@H](c1ncns1)c1cc2occn2c1C[NH+]1Cc2ccccc2C1=O	-5.000
[NH2+]C(N[C@@H](CO)C(=O)O)c1ccc(F)cc1	-4.200
Nc1[nH]c(=O)[nH]c1OC[C@H]([O-])[C@@H](N)[C@@H](O)C(F)(F)F	-4.700
Nc1[nH]c(=O)[nH]c1OC1c2ccccc2C[SH]1C1=NS(=O)(=O)N1C1CC1	-5.400
Nc1[nH]nnc1-c1cn(C2(C(=O)O)CCCCCCC(=O)O2)s1	-5.000
Oc1ccc(Cc2noc([C@@H](O)CNCc3cnnn3-n3nnnc3F)n2)c(C2CC2)c1	-5.100
Nc1[nH]c(=O)[nH]c1O[CH]c1ccccc1CNS(=O)(=O)C1CC1	-4.900
[NH2+][C@H](C[C@H]1COc2ccccc2C1)c1ncn2ncnc2n1	-5.100
[NH2+][C@@H](c1ncc[nH]1)[C@H]1c2[nH]nnc2C[C@@H]2c3ccccc3C[C@@H]21	-5.000
O=C(Nc1ccc2c(c1)nnn2-c1cnnn1-n1nnnc1F)c1cccs1	-5.700
Nc1c([C@H]([NH2+])C2CCC[C@@H]2C(=O)[O-])ncn1Cl	-4.200
O=C(Nc1ccc(C(Oc2nnn[nH]2)n2nnnc2F)cc1)c1ccccc1Cl	-6.200
[NH2+][C@@H](c1ncc[nH]1)[C@@H]1C(=O)N2C(=O)N=C2Sc2nn[nH]c21	-4.900
CP(=O)([O-])OCc1c([C@H]([NH2+])c2ccccc2)nc2ncccn12	-4.800
Nc1[nH]c(=O)[nH]c1O[C@H]1SN[C@@H]2c3ccc(Cl)cc3[C@H]12	-4.900
Nc1[nH]c(=O)[nH]c1-c1nc2c(s1)CCCc1ccc(S(=O)(=O)Nc3nnn[nH]3)cc1-2	-6.000

[NH2+]Cc1cnnc(Oc2c(O)ccc3cccc(O)c23)c1	-5.100
Fc1nnnn1-n1nncc1-n1nnc(N=[NH+]Cc2ccc(OCc3cccs3)cc2)n1	-5.700
Nc1[nH]c(=O)[nH]c1C=C1N[N-]N(c2c(F)cccc2C(F)(F)F)C1=O	-5.200
[NH2+][C@@H](c1nc2ncccc2o1)c1nncs1	-4.700
Nc1[nH]c(=O)[nH]c1-c1csc2nnnc(NSc3nccs3)c12	-5.100
NCc1cc(-c2nnn([C@@H](CCC(=O)O)C(=O)CO)n2)c[nH]c1=O	-4.600
[NH2+][C@@H](c1nc2ccsc2[nH]1)[C@@H]1Oc2c([nH]c3ccccc23)[C@H](O)[C@@H]1O	-5.000
[NH2+][C@@H](c1nc2ncncc2s1)n1[nH+]c2[nH]c3ncccc3nc1-2	-5.100
[NH2+][C@@H](c1nc2cn[nH]c2[nH]1)C1[C@H]2CCC[C@H]1CS(=O)(=O)C2	-4.600
Fc1nnnn1-n1nncc1N[C@H]1[CH]CNC[C@@H]1Cc1ccc(Cl)cc1	-5.300
Nc1[nH]c(=O)[nH]c1-c1n[nH]cc1NS(=O)(=O)C1CCOCC1	-5.300
Fc1nnnn1-n1nncc1OCCc1noc2cc(OCc3c(Cl)cccc3Cl)ccc12	-6.000
Nc1[nH]nc([C@@H]2C[C@@H]3CC[C@H]3N2)c1-c1nnc[nH]1	-4.600
[NH2+][C@@H](c1nc(Br)no1)[C@H](c1ccccc1)c1c[nH]c2ccccc12	-5.000
Fc1ccc(-n2nncc2CO[C@@H](Nc2cnnn2-n2nnnc2F)c2ccccc2)cc1	-5.300
[NH2+][C@@H](c1nc2ncccc2[nH]1)[C@H](Oc1ccc2ccccc2c1)c1ccccc1	-5.700
[NH2+][C@@H](c1ncco1)C(NC1COC1)c1ccccc1-c1ccccc1	-4.300
[NH2+][C@@H]1C=Nc2nnc(-c3ccc(Oc4ccc(F)cc4)nc3)n21	-4.800
Nc1[nH]nnc1-c1cccnc1N[C@@H]1[C@H]2CC[C@@H]1[C@@H]2C(=O)O	-4.800
[NH2+][C@@H](c1ncns1)C1Cc2ncncc2NC1=O	-4.400
[NH2+][C@H](Cn1cc[nH+]c1Nc1ccccc1)c1nc2nccc[c-]2n1	-5.200
Nc1[nH]nnc1C=NNC(=O)[C@@H]1CCCO1	-4.400
Nc1[nH]c(=O)[nH]c1O[CH]c1ccc(N=Nc2ncccc2O)cc1	-4.700
Fc1cc(Oc2ccccc2)cc2c1nnn2-n1nnnc1F	-5.500
Nc1[nH]c(=O)[nH]c1-c1noc([C@@H]2c3ccsc3-c3nnnn32)n1	-6.100
Fc1nnnn1-n1nnc(COc2cnc(OCc3ccc4ccccc4n3)nc2)n1	-6.200
Nc1[nH]nnc1CCCOC(=O)c1c(Cl)cccc1Cl	-4.700
NC(=O)c1ccn2c(COc3[nH]c(=O)[nH]c3N)ncc2c1	-5.100
O=c1[nH]c2ccccc2nc1Oc1cccc(Oc2c(O)noc2O)c1	-5.600
O=S(=O)(CCOCc1ccccc1)Nc1cnnn1-n1nnnc1F	-5.300
O=c1cc(NCc2ccccc2)n(-n2nnnc2F)nn1	-5.500
N#C[C@H](CNC(=O)C1=CC=CCC1)Nc1cnnn1-n1nnnc1F	-5.400
Nc1[nH]c(=O)[nH]c1OCCc1cccc2c1OCCO2	-5.000
Nc1[nH]c(=O)[nH]c1-c1noc(-c2cc3cc(Cl)ccc3[nH]c2=O)n1	-6.200
Nc1[nH]c(=O)[nH]c1O[C@H]1C[C@@H]2OC[C@H]2O1	-4.400
Fc1nnnn1-n1nnc2nc(COc3ccccc3)ccc21	-5.600
Fc1nnnn1-c1nnc(NCc2ccccc2)s1	-5.200
Nc1[nH]c(=O)[nH]c1-c1c[nH]c2c(Cl)ncc(O[C@@H]3CCNC3)c12	-5.300
Nc1[nH]c(=O)[nH]c1O[C@H]1C[C@@H]2OC[C@@H]1[C@@H]2CSc1nc(-c2ccccc2)no1	-5.400
Fc1nnnn1-n1nnc(SONc2nsnc2NCCCn2cnnc2)n1	-5.300
Nc1[nH]c(=O)[nH]c1OC[C@H](CO)NC(=O)COc1cccnc1	-4.700
Nc1[nH]c(=O)[nH]c1-c1noc(C2CCc3ccccc3C2)n1	-6.000
Fc1nnnn1-n1nncc1-n1nnnc1CCc1ccsc1	-5.400
[NH2+][C@@H](c1ncns1)C1CSc2ncccc2O1	-4.400
Nc1[nH]c(=O)[nH]c1-c1ncnc2[nH]c3c(c12)CCOC3(O)CC(=O)O	-5.400
Nc1nnc(Oc2c(N)[nH]c3ncccc23)s1	-4.600
Nc1[nH]c(=O)[nH]c1-c1nn(-c2nc[nH]c2-c2ccccc2)c2c1C(=O)CC(CC(=O)O)OC2=O	-6.000
Nc1[nH]c2ccccc2c1COc1ccc2c(c1)nnn2-n1nnnc1F	-5.600
Nc1[nH]c(=O)[nH]c1O[C@@H]1NC(=O)c2cccc(Cl)c2S1	-5.100
CCn1nnnc1C[C@@H]([NH2+])c1nc2nccc[c-]2n1	-5.000
Fc1nnnn1-n1nnc(C2(N3CC(c4ccccc4)CN3)CC2)n1	-6.100
O=c1c(-n2nnnc2F)cc(Nc2nnnn2Cc2ccccc2)c2cnc(Cl)cn12	-6.500
Nc1[nH]c(=O)[nH]c1OC[C@@H]1CCO[C@H](O)[C@H](O)[C@H]1O	-4.400
[NH2+][C@@H](c1ncccn1)c1cnc(Oc2ccccc2F)s1	-4.600

Fc1nnnn1-n1nncc1Nc1ccc2onc3c2c1CCC3	-6.100
[NH2+][C@@H](c1ncc[nH]1)c1nc2ccccc2c(=O)n1-c1ccccc1	-5.000
[NH2+][C@H](C1=CCCc2ccccc21)c1nccc2[nH]nnc12	-5.200
Nc1[nH]c(=O)[nH]c1-c1csc(Nc2ccc(F)cc2)n1	-5.100
O=C(NN[C@H](Cc1ccccc1)[C@@H](O)Nc1cnnn1-n1nnnc1F)c1ccnc2ccccc12	-6.000
[NH2+][C@H](NCc1ccc2c(c1)OCC2)c1nc(-c2ccc(Br)cc2)no1	-5.200
NC(=O)SCN(Nc1cnnn1-n1nnnc1F)c1ccccc1	-5.000
Nc1[nH]nnc1C=Nc1nc(F)cc(Cc2ccccc2)n1	-5.500
Nc1[nH]c(=O)[nH]c1O[CH]c1ccc(-n2nnnn2)c(C(F)(F)F)c1	-5.500
Fc1nnnn1-c1nn[n-]c1OCSCOCc1csc(-c2ccc(Cl)cc2)n1	-5.300
Fc1ncccc1-c1nc([CH]c2nnccc2-n2nnnc2F)ns1	-5.700
O=C(Oc1ncsc1/[O+]=c1/oc(=O)[nH]s1)c1ccccc1	-5.200
[NH2+][C@@H](c1nc2ncccn2c1F)c1nnnn1Cc1cccs1	-5.300
Nc1[nH]c(=O)[nH]c1O[CH]c1ccc(NC(=O)NCc2cnc[nH]2)cc1	-5.200
Nc1[nH]c(=O)[nH]c1-c1c[nH]c2c(C3CC=NC3=O)cccc12	-5.200
Nc1[nH]c(=O)[nH]c1O[CH]c1ccccc1NS(=O)(=O)C1CC1	-5.100
Nc1[nH]c(=O)[nH]c1OCCC1(S(=O)(=O)c2ccc(Cl)cc2)CCCCC1	-4.900
Nc1[nH]c(=O)[nH]c1-c1nsc(S(=O)(=O)NCc2ccc(Br)cc2)n1	-5.400
[NH2+][C@H](Oc1ccccc1)c1nc2ncccc2n1-c1noc(=O)[nH]1	-5.100
O=C1Nc2cc(Cl)ccc2C1[NH+]Cc1cnnn1-n1nnnc1F	-5.900
Nc1[nH]c(=O)[nH]c1-c1csc(Nc2noc3c2CCCC3)c1	-5.000
O=C(NCCn1c(=O)nnn1-n1nnnc1F)Nc1ccccc1	-5.700
O=c1ccn(Cc2ccccc2)c(Oc2c(O)noc2O)c1	-4.900
Nc1[nH]c(=O)[nH]c1-c1csc(Nc2nn(C3CCCCC3)c3cc[nH]c23)n1	-5.400
O=C(NCc1ccccc1)N(c1nnn(-n2nnnc2F)n1)C1CCNCc2ccccc21	-6.500
O=C(Cl)ONc1cnnn1-n1nnnc1F	-4.800
Oc1oncc1N[C@@H](c1nnc(-c2ccccc2)o1)C1CC1	-5.300
Nc1[nH]c(=O)[nH]c1O[C@H]1SCC2(NC1=O)C(O)[C@@H](O)[C@@H]2O	-5.100
Nc1[nH]c(=O)[nH]c1O[C@H]1C[C@@H]2OC[C@@]21CNCCCC(=O)O	-4.500
Nc1[nH]c(=O)[nH]c1OC(CNC(=O)c1ccc(Cl)c([N+](=O)[O-])c1)Cc1ccccc1	-5.100
[NH2+]C([NH+]=CC(=O)O)C(=O)c1ccc(O)cc1	-4.200
Nc1[nH]c(=O)[nH]c1OCC(=O)c1sc(Nc2ccccc2)nc1-c1ccccn1	-5.400
NNc1nnnn1C(C(=O)O)c1ccc(F)cc1	-4.700
O=C(c1ccco1)[C@@H](COc1c(O)noc1O)Nc1ccccc1	-5.100
O=C1NS/C1=[NH+]/c1[nH]nnc1-c1ccccc1	-5.000
[NH2+][C@@H](c1ncn[nH]1)[C@H](O)[C@@H]1Oc2ccccc2C[C@@H]1C(=O)[O-]	-4.700
O=C(N[C@@H](Cc1c[nH]c2ccccc12)[NH2+]Cc1cnnn1-n1nnnc1F)c1nc(Cl)ccc1F	-6.000
Nc1[nH]c(=O)[nH]c1-c1c[nH]c(=O)n(Cc2ccccc2)c1=O	-5.100
Nc1[nH]c(=O)[nH]c1OC1=Nc2n[nH]c(Cl)c2C1=O	-5.000
N#Cc1cccc(NCc2nc(-c3[nH]c(=O)[nH]c3N)no2)c1	-5.400
Nc1[nH]c(=O)[nH]c1-c1csc(Nc2nncn2-c2cccc(C(F)(F)F)c2)n1	-5.900
Nc1[nH]c(=O)[nH]c1OCc1ncc2cc(F)cc(F)c2n1	-5.300
[NH2+][C@@H](c1ncnc(Br)n1)[C@@H]1c2ccccc2-n2ccnc21	-5.000
Fc1nnnn1-n1nnc(N(Cc2cccnc2)NCc2ncccn2)n1	-5.400
O=C1NN(n2nnnc2F)N=NN1[C@H]1C[CH][CH]C[NH+]1Cc1ccc(F)cc1	-5.300
Oc1noc(O)c1OCNc1n[n-][n+](CCCc2ccccc2)c1O	-5.500
O=c1nnn(-n2nnnc2F)c2c(-c3ccccc3F)oc(C3CC3)c12	-5.400
[NH2+][C@@H](c1ncns1)C(Oc1nnn[nH]1)C1CCCCC1	-4.700
Fc1nnnn1-n1nnc(COCc2ccno2)c1OCc1ccccc1	-5.300
Nc1[nH]c(=O)[nH]c1OC(COC=O)NC(=O)c1ccccc1	-4.900
Nc1[nH]c2ncccc2c1N=Nc1nccs1	-4.800
Nc1[nH]c(=O)[nH]c1OC1(c2n[nH]c3c(-c4nc5ccccc5[nH]4)cccc23)COC1	-5.700
OC1=NN=NC(Oc2noc(O)c2O)C1Cc1ccccc1	-5.200
Nc1[nH]c(=O)[nH]c1-c1noc(CNCCO)n1	-4.500

Nc1[nH]c(=O)[nH]c1OC(=S)Cc1nnnn1Cc1cccnc1	-5.100
[NH2+][C@H](c1nnn(-c2ccccc2)n1)c1nc2cccnc2s1	-5.200
Nc1[nH]c(=O)[nH]c1OC[C@H]([O-])[C@H](Cc1ccccc1)NC(=O)NCc1ccccc1	-5.300
Nc1[nH]c(=O)[nH]c1OC(=S)CNC1=NC(=O)CC1	-4.800
Nc1[nH]c(=O)[nH]c1OC[C@H](N)COC[C@@H]1CC[C@H]2CC21	-4.800
Fc1nnnn1-n1nncc1-n1nnc(N=[NH+]Cc2ccc(Br)cc2)n1	-5.800
O=C(O)CC(C(=O)c1cccc([N+](=O)[O-])c1)N(c1cccc(O)c1)P(=O)([O-])[O-]	-4.800
[NH2+][C@@H](c1nc2nccc[c-]2n1)c1cc(Cl)ccc1-c1ccccc1	-4.800
[NH2+][C@@H](c1ncc[nH]1)[C@@H]1c2cc(NC(=O)c3ccccc3)ccc2-c2nn[nH]c21	-5.400
Fc1nnnn1-n1nnc(CCn2ncnn2)c1CCc1ccccc1	-5.600
[NH2+][C@@H](c1ncno1)[NH+]1CC1c1ccccc1C(F)(F)F	-4.700
[NH2+][C@@H](c1nccs1)[C@@H]1OS(=O)(=O)[C@]12Cc1c(Br)cncc1O2	-4.600
Nc1[nH]c(=O)[nH]c1-c1c(-c2cccs2)nnn1-c1ccccc1	-5.100
O=C1/C(=N/S(=O)(=O)n2cc(O)c(O)c2O)NC12CCCC2	-5.100
Nc1[nH]c(=O)[nH]c1O[C@@H]1CCc2c(cc(Cl)nc2NCc2ccccc2)S1	-5.200
Nc1nc(NC[C@@H]2CCCO2)c2cc(-c3cscc3-c3[nH]c(=O)[nH]c3N)cnc2n1	-5.500
Nc1[nH]c(=O)[nH]c1O[C@H]1SN([C@@H](c2ccccc2)c2c(Cl)cccc2Cl)C1O	-5.500
[NH2+][C@@H](c1ncno1)[C@H](Cc1c[nH]c2ccccc12)c1nc2ccc([N+](=O)[O-])cc2[nH]c1=O	-5.300
O=C(CC[C@@H]1CNC(c2cnnn2-n2nnnc2F)N1)c1ccccc1Cl	-5.600
O=C(O[C@@H]1[C@H]2OC[C@H](Oc3nc(Cl)nn3CCO)[C@H]2[C@H]1O)c1ccco1	-4.700
O=C1c2ccccc2C1C(Oc1c(O)noc1O)c1nsc2ncccc12	-5.300
Nc1[nH]nnc1CNC(=O)c1ccc(Cl)cc1	-5.000
Cc1[nH]c2ccccc2c1C(=O)ON=CNC(N)NC(=O)c1ccccc1	-5.700
Nc1[nH]c(=O)[nH]c1-c1ncnc2[nH]c(C(N)c3ccccc3)cc12	-5.500
Nc1[nH]c(=O)[nH]c1OC(Cc1ccccc1)(Cc1ncn[nH]1)C(=O)N1CCCC1	-5.200
O=C(Nc1ccc(Oc2ccnn2-n2nnnc2F)cc1)Nc1cccc2ncncc12	-5.100
Nc1[nH]c(=O)[nH]c1-c1nnn(-c2ccc(Br)c([N+](=O)[O-])c2)n1	-5.200
N[C@H](CCC(=O)[O-])c1[nH]sc1[C@@H]([NH2+])c1ncc[nH]1	-4.500
Nc1[nH]c(=O)[nH]c1-c1csc(-c2ccnc(N=Nc3nc4ccccc4o3)n2)n1	-5.700
[NH2+][C@@H](c1nc2ncccc2[nH]1)c1c[nH]nc1-c1ccccc1	-4.800
[NH2+][C@@H](c1nc2nccc[c-]2n1)[C@H]1CC(=O)OC1O	-4.600
Fc1nnnn1-n1nncc1N[C@H](C[NH2+]c1ccccn1)c1ccccc1	-5.600
Fc1nnnn1-c1nnn(-c2nsc(NC3CCCCCC3)n2)n1	-5.700
Nc1[nH]c(=O)[nH]c1C1=S(C2CC2)C=C1Nc1nc2scnc2nc1-c1ccccc1	-5.300
O=C1Nc2ccccc2C1C(O)c1cncc2[nH]nnc12	-5.200
Nc1[nH]c(=O)[nH]c1OC1=CNc2c(F)cccc2N1	-4.900
Nc1[nH]nnc1C1C=C2CCN(S(=O)(=O)C3(COc4ccccc4)CCCC3)CC2N1	-5.200
[NH2+][C@H](Cc1nnc(N[C@H]2CCOc3ccc(F)cc32)s1)c1ncco1	-5.500
[NH2+][C@H](c1ccc(F)cc1)c1nc(-n2ccnn2)oc1N[C@H](CCO)c1ccncc1	-4.900
N[C@@H](OCS(N)(=O)=O)c1cccc(-c2nnc(Oc3ccnc4ccccc34)s2)c1	-5.400
O=Nc1nnn(-n2nnnc2F)c1NCc1cccc2cnccc12	-5.700
[NH2+][C@H](NC[C@H]1OC[C@@H]1Cc1ccccc1)c1ncno1	-4.800
[NH2+][C@H](C[C@@H]1NC(=O)c2nc(-c3ccccc3)[nH]c21)C1=Nc2ccncc21	-5.500
Nc1[nH]c(=O)[nH]c1OCCn1cc(-c2ccccc2Cl)nn1	-5.700
[NH2+][C@@H](c1nc2ncccn2c1-n1nnc(-c2ccco2)n1)C1CC1	-5.300
Nc1[nH]c(=O)[nH]c1OC[C@H]1COc2cc(Br)ccc2N1	-5.000

6 Graph representation of the hierarchy of chemical fragments

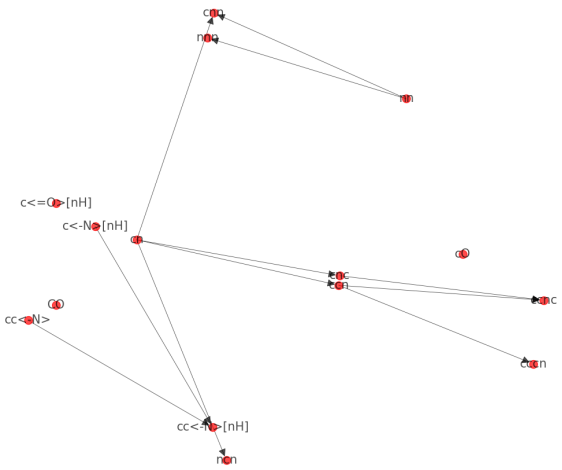

Figure 6: The chemical fragments identified by the molecular fragmenting algorithm in RDKit is represented as a directed graph. The directed edges one fragment to another indicate hierarchy. The terminal nodes then correspond to largest unique fragments. Only the most commonly occuring fragments within the top 200 candidates are shown in this figure for clarity