

De novo molecule design through molecular generative model conditioned by 3D information of protein binding sites

Mingyuan Xu¹, Ting Ran¹, Hongming Chen^{1,*}

¹*Bioland Laboratory (Guangzhou Regenerative Medicine and Health - Guangdong Laboratory),*

Guangzhou 510530, P. R. China

**Correspondence e-mail: chen_hongming@grmh-gdl.cn*

Abstract

De novo molecule design through molecular generative model is gaining increasing attention in recent years. Here a novel generative model was proposed by integrating the 3D structural information of the protein binding pocket into the conditional RNN (cRNN) model to control the generation of drug-like molecules. In this model, the composition of protein binding pocket is effectively characterized through a coarse-grain strategy and the three-dimensional information of the pocket can be represented by the sorted eigenvalues of the coulomb matrix (EGCM) of the coarse-grained atoms composing the binding pocket. In current work, we used our EGCM method and a previously reported binding pocket descriptor DeeplyTough to train cRNN models and compared their performance. It has been shown that the molecules generated with the control of protein environment information have a clear tendency on generating compounds with higher similarity to the original X-ray bound ligand than normal RNN model and also achieving better performance in terms of docking scores. Our results demonstrate the potential application of EGCM controlled generative model for the targeted molecule generation and guided exploration on the drug-like chemical space.

Introduction

In the past few decades, the cost of a new drug from research and development to market is estimated to be between 314 million and 2.8 billion US dollars, which takes more than 10 years on average.^{1,2} Computer-aided drug design (CADD) plays an important role in effectively reducing the cost of drug development and accelerating the research process.^{3,4} Application of CADD has been focused on efficiently identifying lead compounds and the follow-up lead optimization. Traditionally, there are two general strategies, i.e. structure-based drug design and ligand-based drug design,³ and are used in scenarios of utilizing the three-dimensional structure of the target and known ligand structures respectively to infer information about important protein-ligand interaction and the relationship between ligand physicochemical properties and compound bioactivity.^{5,6} For example, the protein X-ray structures or pharmacophore models based on known ligands were used to carry out virtual screening in vast chemical spaces to identify chemical starting points for drug discovery.

In recent years, machine learning methods especially deep learning methods have brought many new breakthroughs to the field of drug molecular design, for example machine learning accelerated ab-initio simulation⁷⁻¹⁰, deep learning based molecular properties prediction¹¹⁻¹³ and binding affinity prediction¹⁴⁻¹⁶ and so on. One particular interesting application of deep learning is the generative modelling for *de novo* molecule design. Molecular generative modelling provides an effective solution for generation of molecules targeting to specific proteins and represents a paradigm shift in the domain of structure generation. Different from the traditional drug design methods, deep generative model is completely data-driven and does not depend on any predefined rule. It utilizes neural networks to generate molecules by learning the underlying probability distribution of structure description from a large amount of molecular structure data.

Current generative modelling methodologies can be divided into two types depending if the molecule structure is described by string-based representation like SMILES or by molecule graph. The neural network architectures like RNN¹⁷, Autoencoder¹⁸, VAE¹⁹, GAN²⁰ have been extensively used for molecular generation tasks. These models can efficiently explore the chemical space and generation of valid molecule structures. However, generating random molecular structures is not what we need. What we need is to be able to design molecule structures that can bind to specific targets and fulfill certain physicochemical properties. In order to gain a better control of the structures generated by neural networks, Seglar *et al* employed transfer learning (TL) to generate molecules that are active to specific targets.²¹ Olivecrona *et al* proposed the REINVENT algorithm in combining RNN and reinforcement learning algorithm (RL) to optimize the score of the generated molecules by fine-tuning the model parameters of RNN, which achieves the purpose of controlling the molecular structure.²² Recently, Kotsias *et al* demonstrated that molecular property constraints can be integrated into

the RNN-based generative model as side information so that the generated molecules tend to meet the constraints of the input.²³ Fabritiis and co-workers proposed a 3D CNN model, LigVoxel, to learn latent vectors representing ligand shapes and use them for controlling the structure generation.²⁴ Although these ligand-based deep generation models have achieved great success, their application has been limited in using ligand properties as control and no protein structure information was used.

Recently, Aumentado-Armstrong have proposed a novel method to utilize protein information for doing targeted design of small molecules, in which a signature of the protein binding site is extracted with a graph convolutional network and combined with ligand latent vector to optimize compound's binding affinity.²⁵ Fabritiis *et al* further developed LiGANN model based on BicycleGAN model. In this model, the structure of the protein pocket was mapped into the shape of the ligand through the BicycleGAN, and then the shape of the ligand was decoded into SMILES through the captioning network.²⁶

In our work, we proposed a novel descriptor for characterizing the three-dimensional structure information of protein binding pocket and integrate it into the conditional RNN model (cRNN) to generate structures tends to bind with specific binding pocket. A coarse-grained strategy was employed to describe the composition of the binding pocket and the sorted eigenvalues of the coulomb matrix between coarse-grained atoms was generated as the descriptor (called as EGCM descriptor) of binding pocket. The EGCM descriptor was then combined with molecule structures to train a cRNN model for structure generation. Our results demonstrated that the molecules generated with the EGCM constrained model, in general, have higher similarity to the X-ray bound ligand than using unconstrained model and also have better docking scores.

Methods and materials

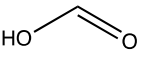

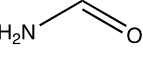

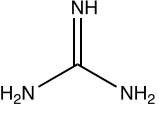
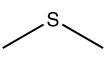
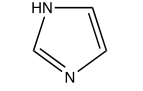

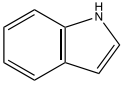

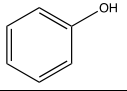
Representation of protein binding pockets

Efforts on construction of descriptor of protein binding pocket have been reported before²⁷⁻³¹. The main purpose of these efforts is to develop methodologies to accurately and effectively represent the complex chemical composition and structure information of the pocket, so that it can be used for investigating similarity between different proteins or for property predicting. Among the existing descriptors, shape-based description methods³² often ignore important interaction information in the protein environment, while energy-based description methods are often implemented by describing the electrostatic potential and van der Waals interaction on the surface of the pocket.³³⁻³⁶ Grid-based methods and graph-based methods are two popular structure-based descriptors in machine learning models. For the former method, the main drawback is that translational and rotational independency is not satisfied. DeeplyTough method is a recent example on generating pocket descriptor based on grid.³¹ For the graph

model, the three-dimensional structure is compressed into a two-dimensional representation, and three-dimensional information is lost at some extent.^{37, 38}

Eigenvalues of Coulomb matrix, as a global 3D representation of molecular structure, has been widely used in the prediction of atomic energies, prioritize geometry searches and so on.³⁹ Once the eigenvalues are sorted, the vector composed by eigenvalues can satisfy the translation, rotation and exchange symmetry.⁴⁰ In current study, we extend the sorted eigenvalues of coulomb matrix (EGCM) concept to represent the protein structure. Given the large number of atoms composing the binding pocket, a coarse-grained strategy was used to simplify the composition of protein binding pocket. Firstly, the ligand was taken as the basis and delineated the residues of protein binding pocket within the radius of 6.5 angstroms around ligand atoms; Secondly, 11 molecular fragments from 20 standard amino acids was defined as key elements of residue and the atoms not included in 11 molecular fragments were ignored. The smiles representation of 11 fragments and its coarse-grained atom type number is as shown in Table 1; Lastly, a dummy coarse-grained atom, whose coordinates are calculated as that of the mass center position of the fragment, was generated to represent each fragment, and a series of ghost atoms located at infinity were introduced to ensure the same size of Coulomb matrix for different pockets. By doing in this way, the protein binding pocket can be represented by a list of coarse-grained dummy atoms and ghost atoms.

Table 1. SMILES representation of 11 molecular fragments and the corresponding type numbers of coarse-grained atoms.

Graph Representation	RDKit Smiles	Atom type index	Max Number	Graph Representation	RDKit Smiles	Atom type index	Max Number
	<chem>C(O)=O</chem>	1	30		<chem>C1=CC=CC=C1</chem>	7	20
	<chem>O=CN</chem>	2	65		<chem>CN</chem>	8	40
	<chem>NC(N)=N</chem>	3	10		<chem>CSC</chem>	9	15
	<chem>C1=CN=CN1</chem>	4	10		<chem>CS</chem>	10	15
	<chem>C1=CNC2=C1C=CC=C2</chem>	5	10		<chem>CO</chem>	11	20
	<chem>C1=CC=C(O)C=C1</chem>	6	20				

The Coulomb matrix of coarse-grained virtual atoms is then constructed, where the element of the matrix is defined as in Eq (1):

$$C_{ij} = \begin{cases} 0.5Z_i^{2.4} & \forall i = j \\ \frac{Z_i Z_j}{|R_i - R_j|} & \forall i \neq j \text{ and } i \notin \text{ghost atoms} \\ 0 & \forall i \neq j \text{ and } i \in \text{ghost atoms} \end{cases} \quad (1)$$

where Z_i is the atom type index of coarse-grained atom i , R_i represents the position of coarse-grained atom i . The eigenvalue of the Coulomb matrix is obtained by solving the eigen equation with Linalg module of Numpy package^{41, 42} and the eigen values are sorted to construct the eigen vector. The sorted eigen vector is the ECGM descriptor for the protein pocket.

Construction of cRNN generative model

The basic work flow of training generative model is as shown in Figure 1. Two types of binding pocket descriptors were employed in our study. One is the recently reported DeeplyTough descriptor generated via convolutional neural network and the other is our ECGM descriptor. The cRNN model developed by Kotsias was used for combining protein descriptor and SMILES input.²³

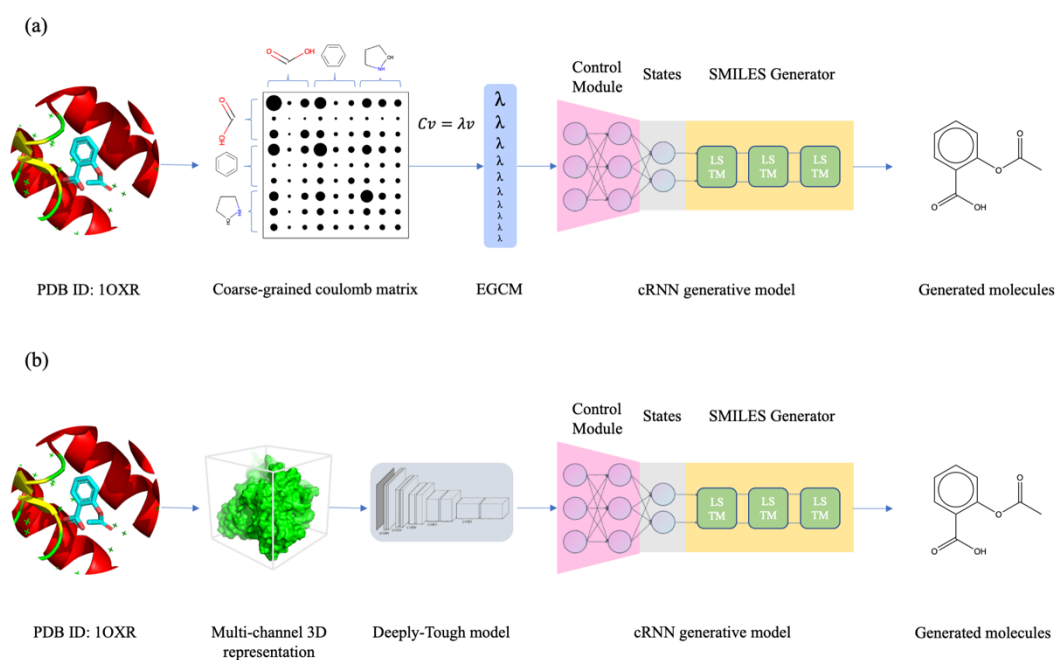


Figure 1. Basic workflow of cRNN molecular generative model with the control of ECGM (a) and DeeplyTough descriptors (b).

Details of Kotsias' model can be referred to the original paper²³, here we only briefly describe the architecture. As shown in Figure 1(a), a three dense layer feed-forward neural network was used as a control module which received the ECGM descriptor as input. A decoder

network containing two LSTM layers and a feed forward layer was employed as the SMILES generator. The output of control module was used to set either the cell state or the hidden state of each of the LSTM layers in the network. Here cRNN model was trained with same structure parameter for hidden layers on different datasets, which the size of two hidden layers of dense neural network is [128,256] and each LSTM layers contains 256 neurons.

During the training process, cRNN generative model follows the ‘teacher’s forcing’ method, and randomized SMILES strings are used to enhance the versatility of generative model. At each step, the model uses the ground truth as prior knowledge instead of the character previously predicted by the network. A batch size of 128 sequences was used along with the Adam optimizer with default parameters and an initial learning rate of 10^{-3} in the training process. A custom learning rate schedule was used, where the learning rate was kept constant for the first 400 epochs and 100 epochs for sc-PDB and eModel-BDB datasets (explained in following section) respectively and then decayed exponentially at each epoch, down to a value of 10^{-6} at the final epoch. The early stop strategy with patience of 100 epoch was adopted to avoid over-fit.

During the molecule generation stage, the trained model was sampled to generate SMILES under the control of pocket structure information. The output vector of each cell of the last LSTM layer was set to a vector representing the possibility distribution of SMILES tokens. During the SMILES generation, a single token per cell was sampled out of this vector using multinomial sampling and a SMILES string was jointly formed in an iterative process until the terminator token were sampled.

As a comparison, we also test the effect of Deeply-Tough descriptor on the control of SMILES generation. The workflow of this scheme is as shown in Figure 1 (b) where the Deeply-Tough descriptor is generated using the codes from Meyers et al³¹, which takes the 3D grid data of the pockets as input and encode them with convolution neural network into a vector space where the proximity of a pair of vector indicates the structural similarity of a pair of pockets. For the sake of comparison, we also trained a RNN based REINVENT model without running reinforcement learning as the baseline generative model without using protein constraints.

Datasets

The datasets used in this work come from two publicly available sources: sc-PDB dataset⁴³ and eModel-BDB dataset⁴⁴. The sc-PDB dataset includes binding pockets for 17499 crystal structures of PDB Bank which corresponds to 5307 UniProt IDs and 7315 unique ligand structures. In order to effectively evaluate the performance of the generative model on the unknown pockets, we divided the training set and test set according to UniProt ID and structures respectively. Under different scenarios, the composition of the dataset is as shown in Table 2.

Table 2. The composition of training set and test set in sc-PDB datasets

Dataset	Split datasets with UniProt ID		Split dataset with structures
	UniProt ID Number	Structure Number	Structure Number
Training set	4748	14414	13960
Test set	559	3035	3489
Total	5307	17449	17449
Training/Total	0.89	0.82	0.8

Given the limited number of binding pockets and ligand in the sc-PDB dataset, a much larger dataset eModel-BDB was used to evaluate the generative model. The details for the eModel-BDB dataset can be found in literature.⁴⁴ eModel-BDB contains around 200,000 protein/ligand complex structures constructed by homology modelling, which corresponds to 108,363 unique drug-like compounds and 2791 proteins in BindingDB dataset. The complexes in eModel-BDB are constructed using template-based approach, in which eThread-template is used for homology modelling of the entire protein and HoloPDB template is used to optimize the binding pocket. There are in total 1357 eThread and 8521 HoloPDB templates for building the homology structures for 2732 protein sequences. We further divided the training set and test set according to protein sequence, eThread-template PDB ID, HoloPDB template PDB ID, and actual protein structures to test the performance of our model. The composition of the divided dataset is shown in Table 3. During the training process, 10% of training set randomly selected were used as validation set to detect the performance of model on the fly.

Table 3. The composition of training set and test set in eModel-BDB datasets in four different schemes.

Split dataset with protein sequence	Dataset	Sequence Number	Structure Number
	training set	2513	172005
	test set	219	21514
	total	2732	193519
	ratio	0.92	0.89
Split dataset with eThread template PDB ID	Dataset	eThread-template PDB ID Number	Structure Number
	training set	1204	168213
	test set	153	25306
	total	1357	193519
	ratio	0.89	0.87
Split dataset with HoloPDB template PDB ID	Dataset	HoloPDB Template PDB ID Number	Structure Number
	training set	7669	173307
	test set	852	20212
	total	8521	193519
	ratio	0.90	0.90

Split dataset with actual protein structures	Dataset	Structure Number
	training set	174167
	test set	19352
	total	193519
	ratio	0.90

Evaluation of model performance

Once the generative models were built, binding pocket descriptors were used to sample the models for structure generation. 20 structures were generated for each pocket in the test set and the structural similarity between the sample set and the ground truth ligand were calculated and compared. Molecule similarity was obtained using ECFP4 fingerprint and RDKit package was used for all the calculation.⁴⁵ Docking study is another way for evaluating the model performance. Various generative models built on eModel-BDB dataset were used to generate structures corresponding to binding pockets in the test set. The sampled structures were docked into protein models and their docking scores were compared with those of random compounds selected from ChEMBL database⁴⁶, compounds sampled from unconstrained RNN model and ground truth compound set. The Vina docking program⁴⁷ was used for the docking study.

Results and discussion

Similarity analysis of binding pockets in training and test set

To evaluate the pocket similarity between test set and training set, the Euclidean distances of EGCM descriptor were calculated and the results is shown in Figure 2. For sc-PDB datasets, the ratio of pockets with distance less than 10 is 6.75% in the test set split with UniProt ID, while the corresponding ratio is 19.65% for test set split with structure. It means that the training and test set, in case of splitting with structure, are more similar than splitting with protein UniProt ID. This is due to the fact that some structures in sc-PDB correspond to the same protein (albeit bound with different ligands), so it can happen that pockets with same UniProt ID can exist in both training and test set in case of splitting with structure. This is avoided when splitting by protein sequence. For eModel-BDB set, there are 0.33% of pockets in the test set with distance less than 5 splitting with eThread-template, 0.06% for splitting with sequence, 1.75% for splitting with HoloPDB ID, and 6.22% for splitting with structures. The impacts of pocket similarity between the training set and test set on the molecular generation will be discussed in the following up section.

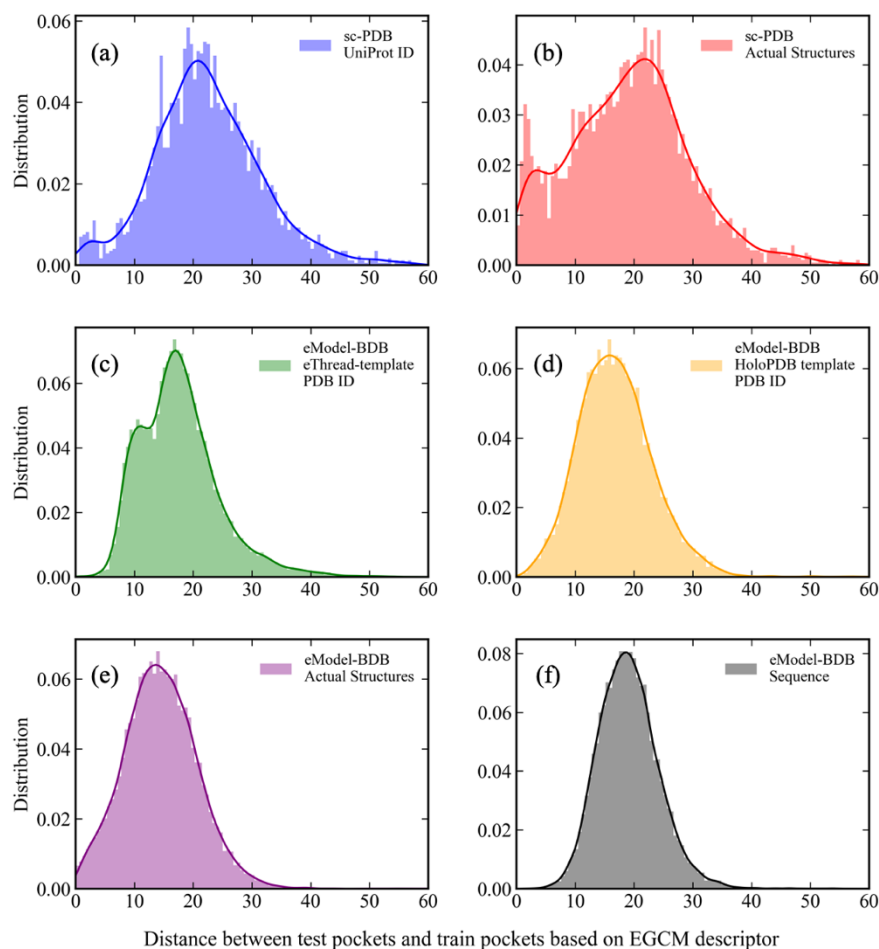


Figure 2. The distribution of distance between test set and training pockets based on EGCM descriptor. Test set of sc-PDB split with UniProt ID (a) and actual structures (b). Test set of eModel-BDB split with eThread-template PDB ID (c), HoloPDB template PDB ID (d), actual structures (e) and sequences (f).

Training results

Firstly, 6 cRNN-based molecule generative models controlled with EGCM and Deeply-Tough descriptors were trained under different divisions for two datasets and the training result is as shown in Table 4. In sc-PDB dataset, it can be seen that the training loss and validation loss of two scheme are almost the same in two different test sets. But in eModel-BDB dataset, the validation loss of Deeply-Tough controlled model is clearly larger than EGCM controlled model.

Table 4. Training results of EGCM cRNN and Deeply-Tough cRNN on sc-PDB and eModel-BDB datasets with different divisions.

Dataset	Methods	Model	Training Loss	Validation Loss	Training Epoch Number
---------	---------	-------	---------------	-----------------	-----------------------

sc-PDB	Split dataset with UniProt IDs	EGCM cRNN	0.0996	0.134	2000
		Deeply-Tough cRNN	0.1065	0.1238	1990
	Split dataset with actual structures	EGCM cRNN	0.0888	0.122	2000
		Deeply-Tough cRNN	0.0623	0.1344	1950
eModel-BDB	Split dataset with sequences	EGCM cRNN	0.1219	0.1636	385
		Deeply-Tough cRNN	0.1185	0.1568	350
	Split dataset with eThread-template PDB IDs	EGCM cRNN	0.0821	0.0955	700
		Deeply-Tough cRNN	0.1256	0.1383	750
	Split dataset with HoloPDB template PDB IDs	EGCM cRNN	0.0818	0.0888	270
		Deeply-Tough cRNN	0.1248	0.1365	290
	Split dataset with actual structures	EGCM cRNN	0.082	0.091	230
		Deeply-Tough cRNN	0.1264	0.1353	585

Similarity analysis on the model generated structures

To verify the control effect of protein binding pocket information on molecular generation, we test the similarity between the generated molecules of controlled model and the known ligands bound in the protein pockets in the test set of both sc-PDB and eModel-BDB datasets. Here, we calculated the dice similarity based on Morgan fingerprint in RDkit package to evaluate the similarity between different molecules. For each pocket in the test set, we generated 20 ligands to calculate the highest similarity to the ground truth ligands. At the same time, the highest similarity for 20 randomly selected drug-like molecules from ChEMBL25, 20 molecules generated by uncontrolled RNN model were also calculated for comparison.

The similarity results on the sc-PDB set are shown in Figure 3 and Table 5 respectively. It seems that the similarity between the molecules generated under the control of both EGCM and Deeply-Tough descriptors and the ground truth ligands is significantly better than those of the random set and ones generated from the uncontrolled model. For sc-PDB test set split by UniProt ID, the percentage of compound whose similarity with the ground truth is higher than 0.5 is 38.9% and 32.5% for EGCM and Deeply-Tough models respectively while the random set and uncontrolled set are 1.7% and 3.8% only. When the test set is split with structure, EGCM and Deeply-tough model can achieve 59.6% and 53.6% respectively and the random set and uncontrolled set are only 1.4% and 3.8%. This indicates that the pockets environmental information can have a significant control effect on molecule generation. It is noticed that the results on the test set split by structures is also better than on the test set split by protein sequence. As we mentioned above, this is due to the fact that splitting data on structure can lead to pockets of same sequence being distributed into both training and test set. Another observation is that the results of the model using EGCM descriptor is slightly better than using Deeply-Tough descriptor. The training of Deeply-Tough model is to make sure that a pair of similar protein are mapped to as close as possible in the latent space, while dissimilar protein pairs are mapped as far away as possible. This may make the Deeply-Tough descriptor is not directly related to

3D information of binding pocket, while EGCM descriptor is rather sensitive to the 3D structure information of the pocket per definition.

Table 5. Ratio of pockets with greatest similarity larger than 50% between molecules generated by different methods with pocket-bound ligands on sc-PDB dataset.

Dataset	Methods	Model	Ratio of similarities greater than 0.5
sc-PDB Dataset	Split dataset with UniProt ID	EGCM cRNN	38.90%
		Deeply-Tough cRNN	32.50%
		Random	1.70%
		Uncontrolled RNN	3.80%
	Split dataset with structure	EGCM cRNN	59.60%
		Deeply-Tough cRNN	53.60%
		Random	1.40%
		Uncontrolled RNN	3.80%

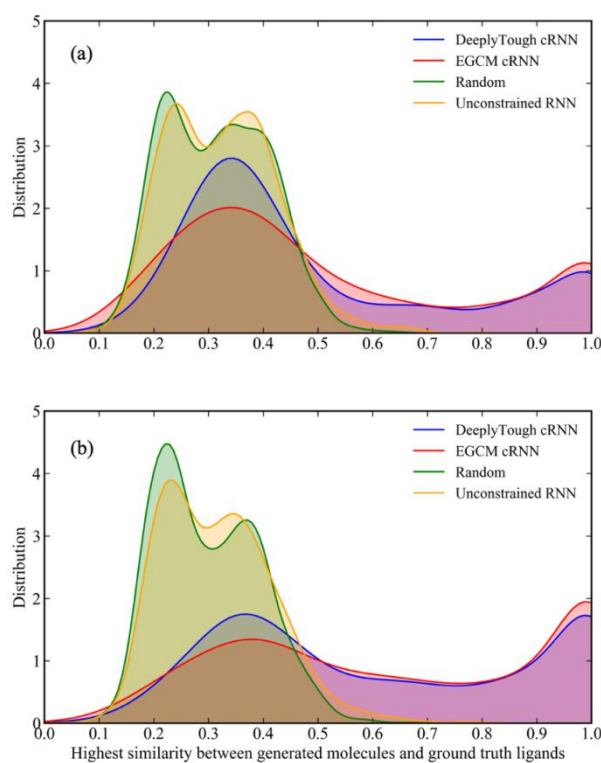


Figure 3. The similarity distribution between generated molecules and ground truth ligands on (a) sc-PDB test set split by UniProt IDs and (b) sc-PDB test set split by structures. the red, blue, green and orange curve represents the result of EGCM, Deeply-Tough, random selection and molecule set generated from uncontrolled model respectively.

To show more intuitively the similarity between the molecule generated with constrained model and original X-ray bound ligand, the top 9 most similar generated molecules of two

example (PDB ID: 1G1D and 5IL1) are as shown in Figure 4. It is clear that the structure of molecules generated with constrained model is closer to the ground-truth.

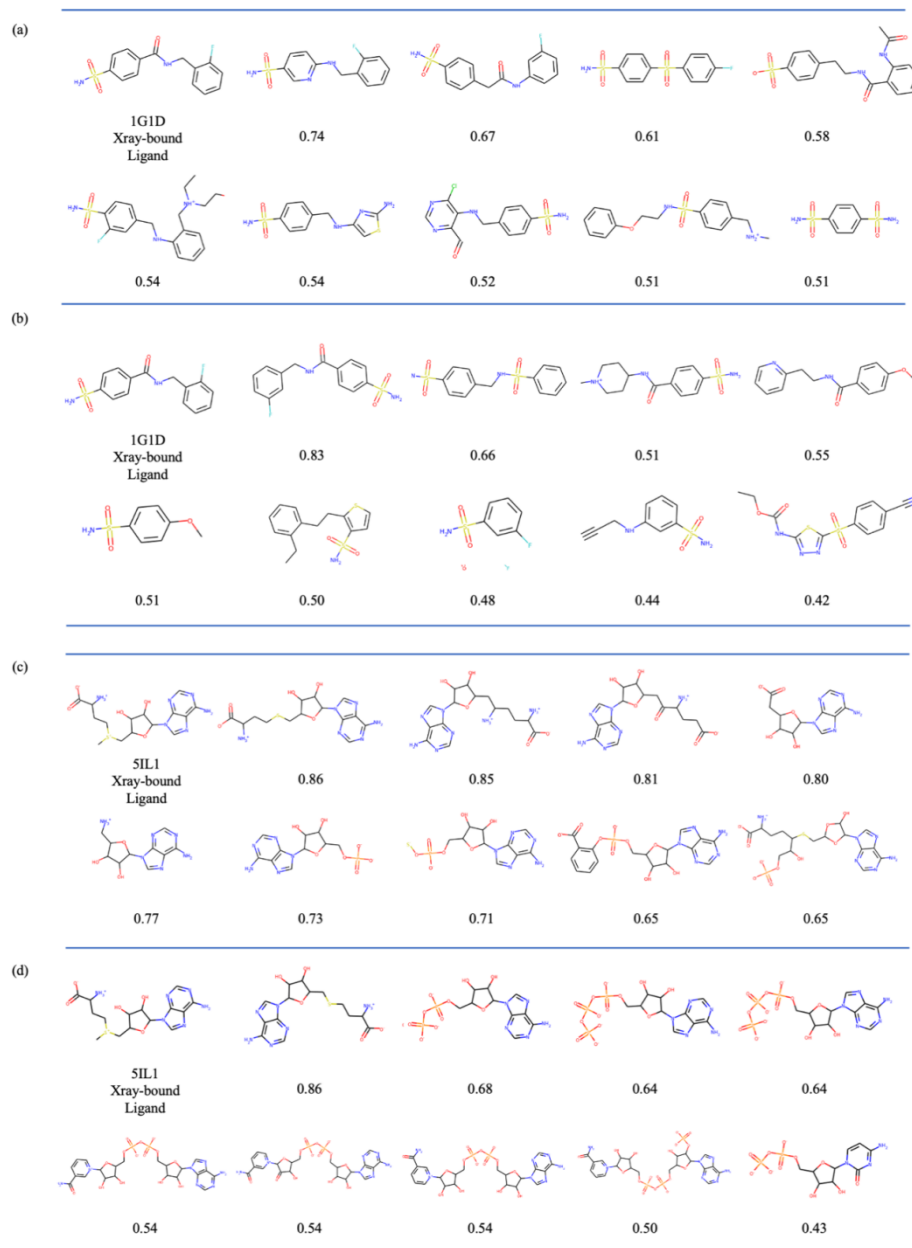


Figure 4. Original Xray bound ligand and top 9 most similar molecules generated with two constrain methods: (a) EGCM (PDB ID: 1G1D), (b) Deeply-Tough (PDB ID: 1G1D), (c) EGCM (PDB ID: 5IL1), (d) Deeply-Tough (PDB ID: 5IL1). The similarity between Xray-bound ligand and generated molecules are list under the molecule graph.

To analyze the influence of the pocket similarity between test and training pockets on the control effect, we further divide the test set into different intervals according to their distance to the training set, and the similarity distribution for compounds generated from pockets in each interval is displayed in Figure 5. It seems that, for both EGCM and Deeply-Tough descriptors,

the median similarity between generated molecules and ground truth ligands increase as the descriptor distance decreases. The same trend is observed on the sc-PDB test set split with structures as shown in Figure S1. This result suggests that in order to use model for generating structures for a novel target, it'd better to have similar binding pockets included in the training set.

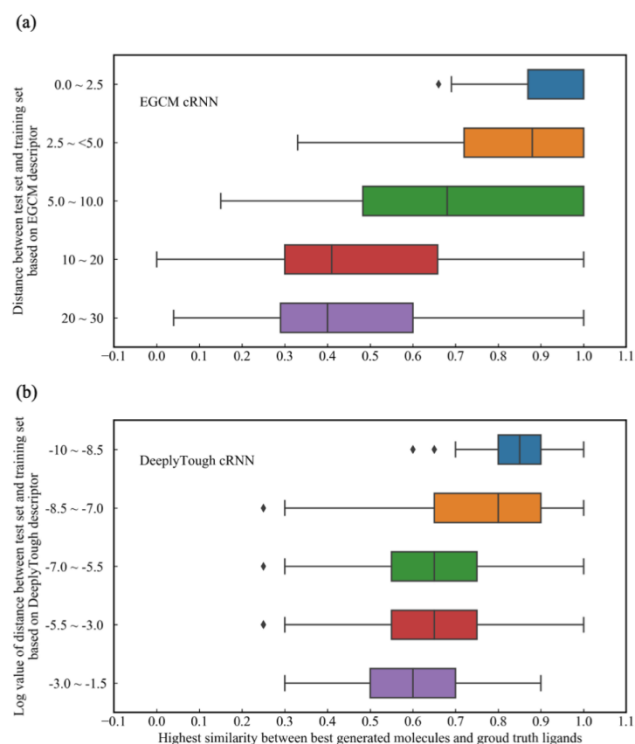


Figure 5. Similarity distributions of the intervals divided by the descriptor distance to the training set for (a) EGCM model and (b) Deeply-Tough model. The result is for sc-PDB test set split with UniProt IDs.

We further test the performance of pocket environment controlled molecular generative model on the eModel-BDB dataset. Four different kinds of splitting were applied on the dataset to obtain various training/test sets (as described in the previous section). The distribution of pockets according to their highest similarity between generated molecules and ground truth ligands in the different test sets are shown in Figure 6 and particularly the ratio of pockets with highest similarity larger than 0.5 between molecules generated by different models and ground truth ligands are shown in Table 6. It seems that the percentage of pocket with high similarity is lower than that of sc-PDB dataset. We speculate that there are two possible reasons: Firstly, the quality of the eModel-BDB is not as good as the sc-PDB set, since all the protein structures in eModel-BDB were constructed via homology modelling, not the experimental X-ray structure, this will largely bring in noise to the data set; Secondly, in eModel-BDB, the diversity of ligands belonging to each protein structure is in general much larger than that in the sc-PDB set. The similar control inputs during the teaching force training process can cause a one-to-

many mapping which is expected to increase the diversity of generated structures and may lead to deviation to the ground truth structures for each pocket. Nevertheless, it can be seen from Table 6 that, comparing with random set and structures generated by uncontrolled model, there is still large improvement in terms of the ratio of pockets with ligand similarity larger than 0.5 in structure generation under the control of EGCM as well as Deeply-Tough in all splitting scenarios. It is also observed that EGCM models perform better than Deeply-Tough models, which is consistent to the results of sc-PDB dataset. For EGCM model, the dataset split by PDB file achieves best result, which is also consistent to the conclusion draw on the sc-PDB set. As discussed in the previous section, this is again due to the similarity between test set and training set.

Table 6. Ratio of pockets with highest similarity larger than 0.5 between molecules generated by different methods and ground truth ligands on eModel-BDB dataset.

Dataset	Methods	Model	Ratio of similarities greater than 50%
eModel-BDB Dataset	Split Dataset with Sequence	EGCM cRNNs	20.60%
		Deeply-Tough cRNNs	13.50%
		Random	3.33%
		Uncontrolled RNN	3.81%
	Split Dataset with eThread-template	EGCM cRNNs	16.24%
		Deeply-Tough cRNNs	12.03%
		Random	2.24%
		Uncontrolled RNN	4.50%
	Split Dataset with HoloPDB ID	EGCM cRNNs	23.65%
		Deeply-Tough cRNNs	11.86%
		Random	3.33%
		Uncontrolled RNN	3.90%
	Split Dataset with PDB File	EGCM cRNNs	43.37%
		Deeply-Tough cRNNs	16.74%
		Random	2.95%
		Uncontrolled RNN	2.66%

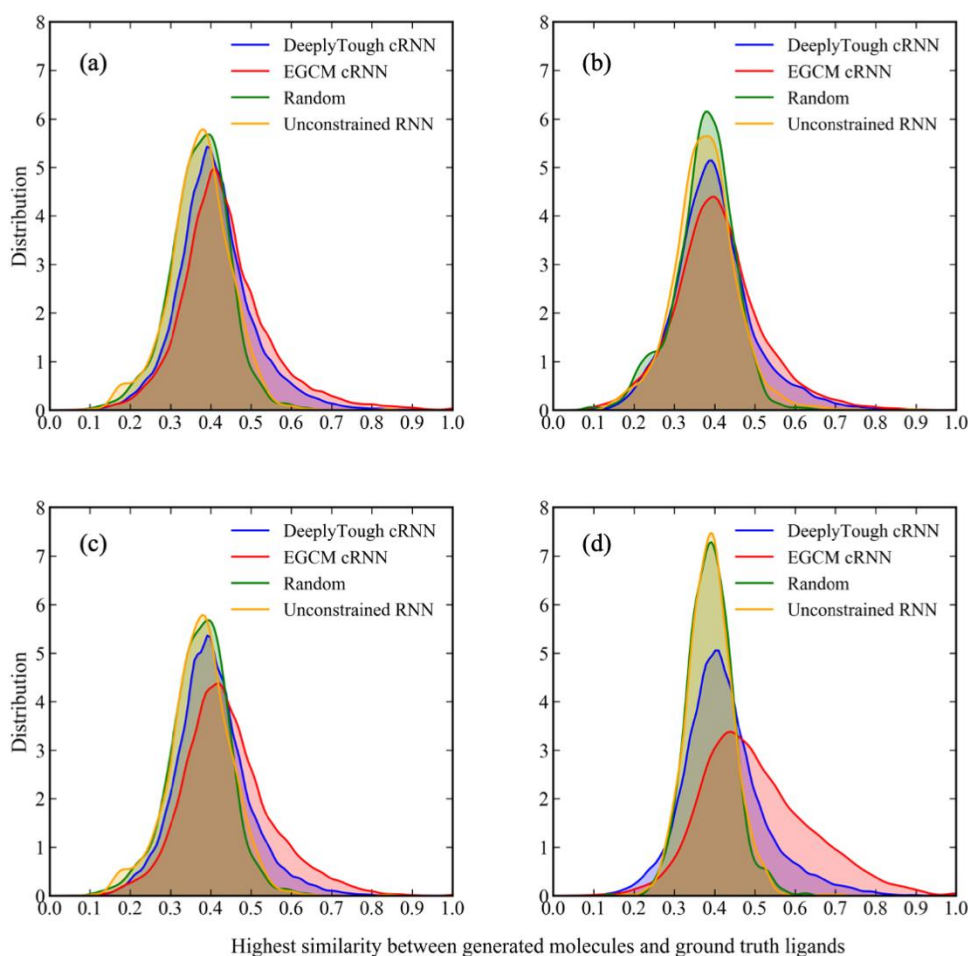


Figure 6. The pocket distribution according to the highest similarity between generated molecules and ground truth ligands on eModel-BDB dataset split by sequence (a), eThread-template (b), HoloPDB ID (c) and structures (d). The red, blue, green and orange curves represent the result of EGCM, DeeplyTough, random selection and unconstrained RNN model respectively.

Structure validity

The validity of generated molecules is also an important metric for evaluating generative models. The distribution of validity of generated molecule under the control of pocket environmental information on sc-PDB test set is shown in Figure 7. It is clear that the validity performance of EGCM model is worse than the uncontrolled model. The validity of generated molecules for less than 10% pockets in test set are less than 50%. We speculate that the possible reasons are: Firstly, the sc-PDB training set is still too small and SMILES grammar rules are not fully learned by the molecular generative models; Secondly, the ground truth ligands for the pockets with poor molecular validity may have low similarity to the ones in the training set.

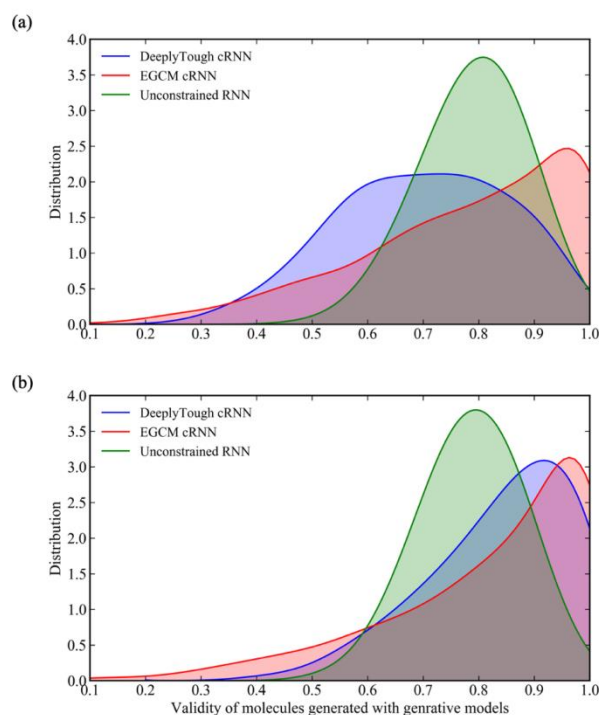


Figure 7. The validity distribution of generated molecules under the control of pocket environmental information for the sc-PDB test set split with (a) UniProt IDs and (b) structures.

To figure out the reason of low molecule validity for some pockets, the relationship between molecular validity for test set pockets and ground truth ligand similarity between the test set and training set was investigated. The density distribution of pockets with less than 50% validity along the validity and ligand similarity to their nearest neighbors in the training set is shown in Figure 8. It is clear that, for those low validity test pockets, the highest similarity between their ground truth ligands and the ones in training set is in general quite low. This implies that these ground truth ligands are dissimilar to the ones in the training set and given the limited number of compounds in the training set, the generative models are not trained well enough. Therefore, we expect that increasing the diversity of molecules in the training set should improve the validity of the generative model.

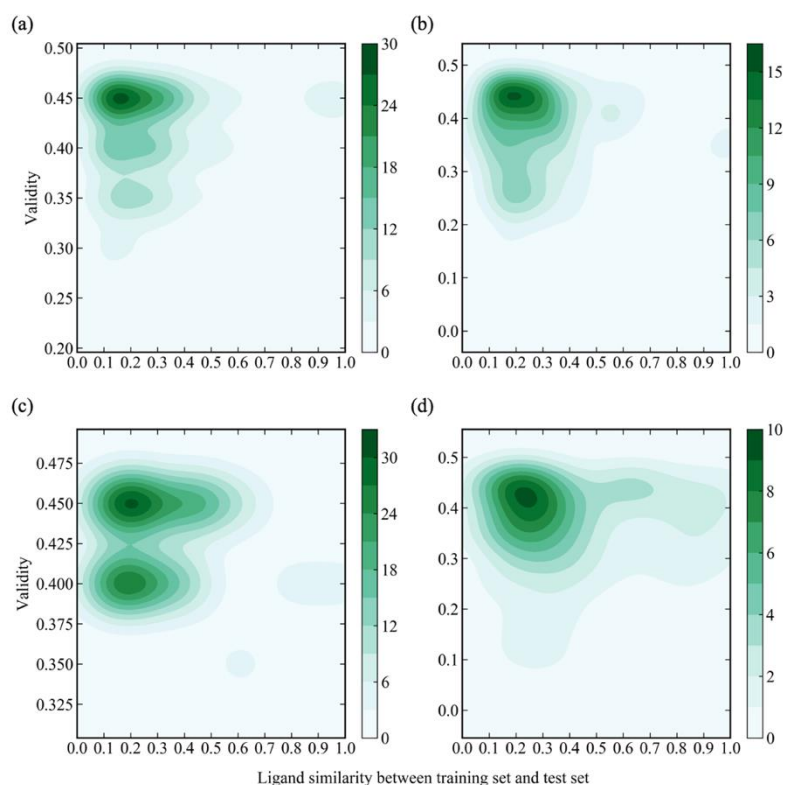


Figure 8. Two-dimensional density distribution between molecular validity and the ground truth ligand similarity between the test pockets and pockets in training set. X-axis is the ligand similarity between test and training set. Y-axis is the pocket validity of test pockets. The darker color corresponds to higher pocket density in the region.

To verify this, a similar validity analysis on the eModel-BDB dataset, which contains more than 100,000 different ligands in the training set, was carried out. The validity of molecule generated for pockets in the various eModel-BDB test sets is shown in Figure 9. It is clear that, in all circumstances, the validity distribution of molecules generated from pockets in eModel-BDB is much better than the one in sc-PDB dataset, which confirms our conjecture.

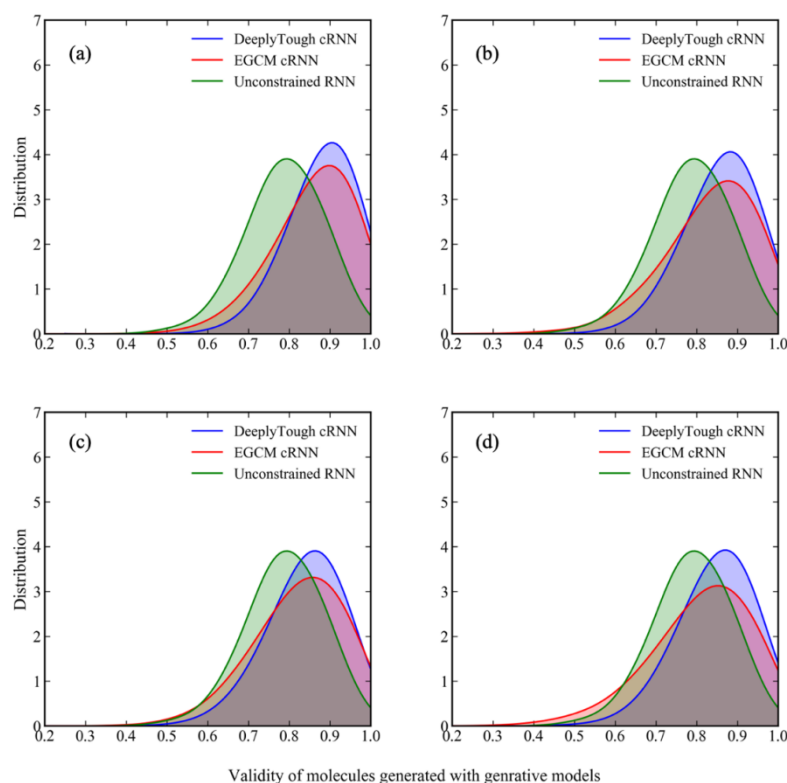


Figure 9. The distribution of validity of molecules generated under the control of pocket environmental information on (a) the eModel-BDB test set split with sequence; (b) split with eThread-template; (c) split with HoloPDB ID; and (d) split with structures.

Docking scores

Another way to evaluate the targeted generative model is to check how well those generated compounds dock into the target protein structures. Here, we use Vina docking package⁴⁷ to perform protein ligand docking on different test set of eModel-BDB dataset as an indication of the potential binding affinity of molecules generated through pocket environmental information control. Given the results of ligand similarity analysis, two generative models which used the way of splitting dataset by structure and by HoloPDB were used in the docking evaluation study.

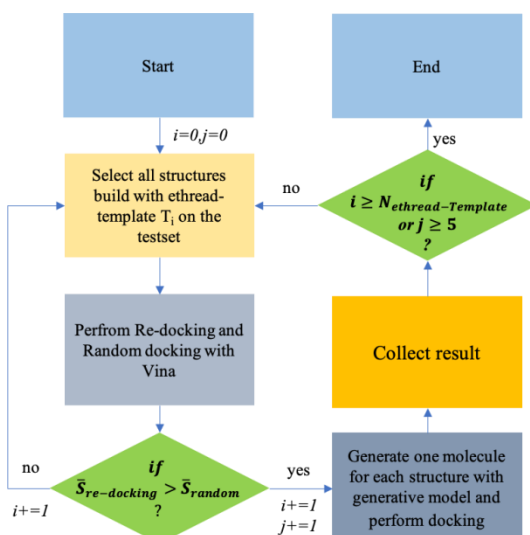


Figure 10. The docking test workflow on the test set of eModel-BDB split with actual structures.

For the test set picked up by structures, a simple workflow was designed as shown in Figure 10. The pockets were first clustered by their eThread-templates, and the ground truth ligands for the proteins in each cluster were collected and re-docked to their belonging proteins for obtaining their re-dock scores. On the other hand, same amount of randomly selected molecules of ChEMBL database were also docked to the same proteins in the cluster and obtain their docking score. For the clusters whose median re-dock scores are better than that of the random set, five example clusters (as shown in Table 7) were selected for comparing docking scores of compounds selected via difference sources. In this case, same amounts of molecules (the number is the same to the re-docked ground truth ligands and the random set) were generated from EGCM, Deeply-Tough and uncontrolled models and were docked to their individual belonging proteins of the cluster. The distribution of docking scores for each cluster are shown in Figure 11 and Table 7. In all five cases, the ground truth ligands always obtained best docking score and EGCM and Deeply-Tough models got better score than that of the random sets and the uncontrolled models. The T-test results show that the the median scores between cRNN models and random set/uncontrolled model are significantly different. These results demonstrate that the pocket environmental information clearly show control effect on structure generation. Again, the EGCM models seems perform slightly better than that of the Deeply-Tough models. Similar procedure was applied on the test set picked up by HoloPDB ID. Four clusters were selected for comparing the docking scores of compounds generated from different sources. The distribution of docking score for these four examples are shown in Figure 11 and Table 7. The same trend, ground truth ligands > EGCM models > Deeply tough > Random set ~ Uncontrolled model, was observed among the examples. For illustration purpose, top 9 generated molecules (both EGCM and DeeplyTough models) with highest docking scores for Mitogen-activated protein kinase 14 (MAPK14, UniProt ID: Q16539) built with eThread-

template of 2ewaA were exhibited in Figure 12 together with a known MAPK14 ligand structure. It is interesting to see that some generated compounds show certain extent similarity to the known MAPK14 ligands, while some are quite different. All the displayed structures have good docking score though.

Kotsias *et al* reported that combining molecule descriptor such as physicochemical descriptor or structural fingerprint with the cRNN model can steering the structure generation along certain criteria. As an extension, our results demonstrate that using pocket environmental information as control signal, the cRNN based generative models can generate better compounds, comparing with the random set and uncontrolled model, in terms of the similarity to the ground truth ligands for the protein and also the docking score. The EGCM descriptor of binding pocket seems better catch up the pocket environmental information than the DeeplyTough descriptor. The same trend was found for other splitting scheme and the corresponding result are as shown in Figure S2~S4 and Table S1~S3 We believe this type generative model can be useful for generating structures for proteins which doesn't have too many structure activity data.

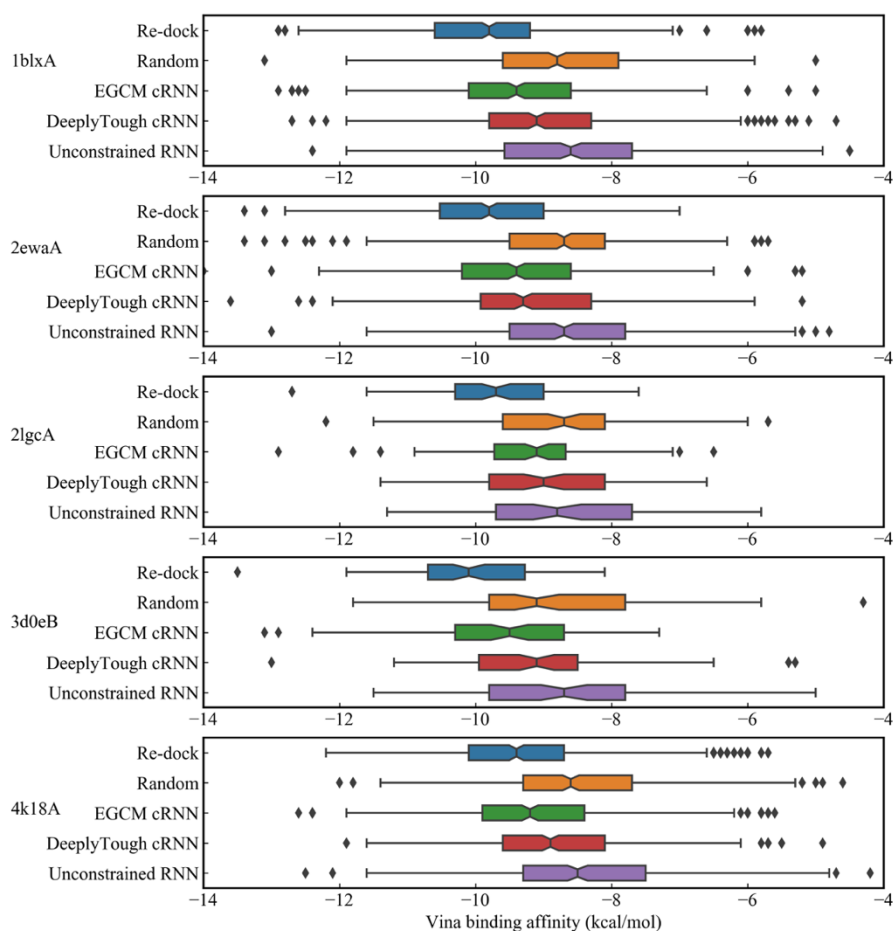


Figure 11. Distribution of Vina binding affinity of generated molecules with the control of Sorted-EGCM (green) and Deeply-Tough descriptors (red) and randomly selected (orange) or generated (purple) molecules and redocked pocket-bound ligands (blue) on eModel-BDB test set split with structures.

Table 7. The average and median Vina binding affinity of generated molecules with different methods on the eModel-BDB test set split with structures.

Template PDB ID	Ligand number	Methods	Average Vina Binding affinity (kcal/mol)	Median Vina Binding affinity (kcal/mol)	P value in T test
1blxA	450	Redock	-9.85	-9.80	3.4836e-35
		Random	-8.79	-8.80	1.0000
		Sorted-EGCM	-9.33	-9.40	3.4851e-10
		Deeply Tough	-9.02	-9.10	0.0128
		Uncontrolled RNN	-8.64	-8.60	0.1205
2ewaA	451	Redock	-9.87	-9.80	6.9130e-38
		Random	-8.80	-8.70	1.0000
		Sorted-EGCM	-9.36	-9.40	1.1116e-11
		Deeply Tough	-9.22	-9.30	9.5719e-07
		Uncontrolled RNN	-8.63	-8.70	0.0642
2lgcA	98	Redock	-9.68	-9.70	1.9697e-07
		Random	-8.81	-8.70	1.0000
		Sorted-EGCM	-9.18	-9.10	0.0281
		Deeply Tough	-9.00	-9.00	0.3056
		Uncontrolled RNN	-8.66	-8.80	0.4474
3d0eB	97	Redock	-10.03	-10.10	1.1164e-09
		Random	-8.81	-9.10	1.0000
		Sorted-EGCM	-9.57	-9.50	0.0002
		Deeply Tough	-9.17	-9.10	0.1004
		Uncontrolled RNN	-8.78	-8.70	0.8734
4k18A	451	Redock	-9.26	-9.40	3.4165e-19
		Random	-8.50	-8.60	1.00
		Sorted-EGCM	-9.07	-9.20	9.8226e-11
		Deeply Tough	-8.86	-8.90	3.2560e-05
		Uncontrolled RNN	-8.44	-8.50	0.5204

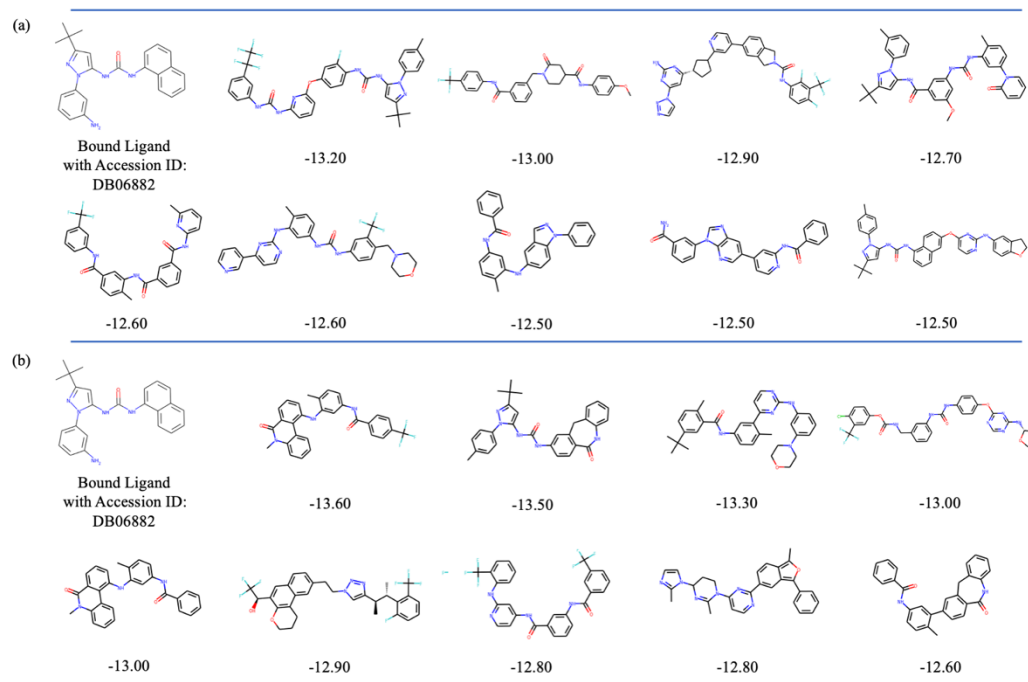


Figure 12. Top 9 molecules with highest docking scores generated with two constrain methods: (a) EGCM, (b) Deeply-Tough for homology structures of Mitogen-activated protein kinase 14 (UniProt ID: Q16539) built with eThread-template of 2ewaA and one of the bound ligands in Drug Bank (Accession ID: DB06882). The Vina docking score of generated molecules are list under the molecule graph.

Conclusion and outlook

Here, the control effect of using protein pockets environmental information as part of input to an existing SMILES generator architecture based on RNNs has been investigated. It has been shown that the molecules generated with the control of protein environment information have a clear tendency on generating compounds with higher similarity to the original X-ray bound ligand than normal RNN model. And it is more obvious when the target is similar to the learned pockets. Additionally, the molecules generated with pocket environmental control have a better docking performance compared with the base line of molecules randomly generated by unconstrained model. It indicates the cRNN based structure generator can sample the drug-like chemical space more efficiently, which suggests the structure based generative model could play an important role in virtual screening and it is worth to be further explored in the future.

In this work, we have shown the sorted eigenvalues of coulomb matrix of coarse-grained atoms is a useful way to represent the structure and composition of protein binding pockets on the task of controlling molecule generation. Compared with Deeply-Tough descriptor, it satisfies the symmetry of translation and rotation exchange and can better distinguish similar pocket structures which make it more suitable to guide the molecule generative model to explore the drug-like chemical spaces. Our results have shown the importance of high-precision

and high-diversity protein-ligand complex datasets for the development of structure-based molecular generation models and also the potential application of EGCM controlled generative model for the targeted molecule generation and guided exploration on the drug-like chemical space.

Acknowledgment

M. Xu would like to acknowledge the funding of Xtal Pi Inc. for his postdoc. research.

Reference

1. Wouters, O. J.; McKee, M.; Luyten, J., Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018. *JAMA* **2020**, 323 (9), 844-853.
2. DiMasi, J. A.; Grabowski, H. G.; Hansen, R. W., Innovation in the pharmaceutical industry: New estimates of R&D costs. *J. Health Econ.* **2016**, 47, 20-33.
3. Yu, W.; MacKerell, A. D., Jr., Computer-Aided Drug Design Methods. *Methods Mol. Biol.* **2017**, 1520, 85-106.
4. Drews, J., Drug discovery: a historical perspective. *Science* **2000**, 287 (5460), 1960-4.
5. Acharya, C.; Coop, A.; Polli, J. E.; Mackerell, A. D., Jr., Recent advances in ligand-based drug design: relevance and utility of the conformationally sampled pharmacophore approach. *Curr. Comput.-Aided Drug Des.* **2011**, 7 (1), 10-22.
6. Ferreira, L. L. G.; Andricopulo, A. D., Editorial: Chemoinformatics Approaches to Structure- and Ligand-Based Drug Design. *Front. Pharmacol.* **2018**, 9, 1416.
7. Behler, J.; Lorenz, S.; Reuter, K., Representing molecule-surface interactions with symmetry-adapted neural networks. *J. Chem. Phys.* **2007**, 127 (1).
8. Zhang, L. F.; Han, J. Q.; Wang, H.; Car, R.; Weinan, E., Deep Potential Molecular Dynamics: A Scalable Model with the Accuracy of Quantum Mechanics. *Phys. Rev. Lett.* **2018**, 120 (14).
9. Casalino, L.; Dommer, A.; Gaieb, Z.; Barros, E. P.; Sztain, T.; Ahn, S.-H.; Trifan, A.; Brace, A.; Bogetti, A.; Ma, H.; Lee, H.; Turilli, M.; Khalid, S.; Chong, L.; Simmerling, C.; Hardy, D. J.; Maia, J. D. C.; Phillips, J. C.; Kurth, T.; Stern, A.; Huang, L.; McCalpin, J.; Tatineni, M.; Gibbs, T.; Stone, J. E.; Jha, S.; Ramanathan, A.; Amaro, R. E., AI-Driven Multiscale Simulations Illuminate Mechanisms of SARS-CoV-2 Spike Dynamics. *bioRxiv* **2020**, 2020.11.19.390187.
10. Lu, D.; Wang, H.; Chen, M.; Lin, L.; Car, R.; E, W.; Jia, W.; Zhang, L., 86 PFLOPS Deep Potential Molecular Dynamics simulation of 100 million atoms with ab initio accuracy. *Comput. Phys. Commun.* **2021**, 259, 107624.
11. Bennett, W. F. D.; He, S.; Bilodeau, C. L.; Jones, D.; Sun, D.; Kim, H.; Allen, J. E.; Lightstone, F. C.; Ingólfsson, H. I., Predicting Small Molecule Transfer Free Energies by Combining Molecular Dynamics Simulations and Deep Learning. *J. Chem. Inf. Model* **2020**, 60 (11), 5375-5381.
12. Hansen, K.; Biegler, F.; Ramakrishnan, R.; Pronobis, W.; von Lilienfeld, O. A.; Müller, K.-R.; Tkatchenko, A., Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *The Journal of Physical Chemistry Letters* **2015**, 6 (12), 2326-2331.
13. Jiang, S.; Balaprakash, P., Graph Neural Network Architecture Search for Molecular Property Prediction. *arXiv e-prints* **2020**, arXiv:2008.12187.
14. Zhu, F.; Zhang, X.; Allen, J. E.; Jones, D.; Lightstone, F. C., Binding Affinity Prediction by Pairwise Function Based on Neural Network. *J. Chem. Inf. Model* **2020**, 60 (6), 2766-2772.
15. Li, S.; Wan, F.; Shu, H.; Jiang, T.; Zhao, D.; Zeng, J., MONN: A Multi-objective Neural Network for Predicting Compound-Protein Interactions and Affinities. *Cell Systems* **2020**, 10 (4), 308-322.e11.
16. Hassan-Harrirou, H.; Zhang, C.; Lemmin, T., RosENet: Improving binding affinity prediction by leveraging molecular mechanics energies with a 3D Convolutional Neural Network. *bioRxiv* **2020**, 2020.05.12.090191.
17. Medsker, L. R.; Jain, L., Recurrent neural networks. *Design and Applications* **2001**, 5.
18. Bank, D.; Koenigstein, N.; Gyrnes, R. Autoencoders 2020, p. arXiv:2003.05991.

- <https://ui.adsabs.harvard.edu/abs/2020arXiv200305991B> (accessed March 01, 2020).
19. Doersch, C. Tutorial on Variational Autoencoders 2016, p. arXiv:1606.05908. <https://ui.adsabs.harvard.edu/abs/2016arXiv160605908D> (accessed June 01, 2016).
 20. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A. A., Generative Adversarial Networks: An Overview. *IEEE Signal Process. Mag.* **2018**, *35* (1), 53-65.
 21. Segler, M. H.; Kogej, T.; Tyrchan, C.; Waller, M. P., Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS central science* **2018**, *4* (1), 120-131.
 22. Olivecrona, M.; Blaschke, T.; Engkvist, O.; Chen, H. M., Molecular de-novo design through deep reinforcement learning. *J Cheminformatics* **2017**, *9*.
 23. Kotsias, P.-C.; Arús-Pous, J.; Chen, H.; Engkvist, O.; Tyrchan, C.; Bjerrum, E. J., Direct steering of de novo molecular generation with descriptor conditional recurrent neural networks. *Nature Machine Intelligence* **2020**, *2* (5), 254-265.
 24. Skalic, M.; Varela-Rial, A.; Jimenez, J.; Martinez-Rosell, G.; De Fabritiis, G., LigVoxel: inpainting binding pockets using 3D-convolutional neural networks. *Bioinformatics* **2019**, *35* (2), 243-250.
 25. Aumentado-Armstrong, T., Latent Molecular Optimization for Targeted Therapeutic Design. *arXiv e-prints* **2018**, arXiv:1809.02032.
 26. Skalic, M.; Sabbadin, D.; Sattarov, B.; Sciabola, S.; De Fabritiis, G., From Target to Drug: Generative Modeling for the Multimodal Structure-Based Ligand Design. *Molecular Pharmaceutics* **2019**, *16* (10), 4282-4291.
 27. Levitt, D. G.; Banaszak, L. J., POCKET: a computer graphics method for identifying and displaying protein cavities and their surrounding amino acids. *Journal of molecular graphics* **1992**, *10* (4), 229-234.
 28. Jiang, M.; Li, Z.; Bian, Y.; Wei, Z., A novel protein descriptor for the prediction of drug binding sites. *BMC bioinformatics* **2019**, *20* (1), 1-13.
 29. Morris, R. J.; Najmanovich, R. J.; Kahraman, A.; Thornton, J. M., Real spherical harmonic expansion coefficients as 3D shape descriptors for protein binding pocket and ligand comparisons. *Bioinformatics* **2005**, *21* (10), 2347-2355.
 30. Borrel, A.; Regad, L.; Xhaard, H.; Petitjean, M.; Camproux, A.-C., PockDrug: A model for predicting pocket druggability that overcomes pocket estimation uncertainties. *J. Chem. Inf. Model* **2015**, *55* (4), 882-895.
 31. Simonovsky, M.; Meyers, J., DeeplyTough: Learning Structural Comparison of Protein Binding Sites. *J. Chem. Inf. Model* **2020**, *60* (4), 2356-2366.
 32. Coleman, R. G.; Sharp, K. A., Protein pockets: inventory, shape, and comparison. *J. Chem. Inf. Model* **2010**, *50* (4), 589-603.
 33. Goodford, P. J., A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. *J. Med. Chem.* **1985**, *28* (7), 849-857.
 34. Laurie, A. T.; Jackson, R. M., Q-SiteFinder: an energy-based method for the prediction of protein-ligand binding sites. *Bioinformatics* **2005**, *21* (9), 1908-1916.
 35. An, J.; Totrov, M.; Abagyan, R., Pocketome via comprehensive identification and classification of ligand binding envelopes. *Mol. Cell. Proteomics* **2005**, *4* (6), 752-761.
 36. Bitencourt-Ferreira, G.; de Azevedo, W. F., Development of a machine-learning model to predict Gibbs free energy of binding for protein-ligand complexes. *Biophys. Chem.* **2018**, *240*, 63-69.
 37. Samanta, B.; De, A.; Jana, G.; Chattaraj, P. K.; Ganguly, N.; Gomez-Rodriguez, M., NeVAE: A Deep Generative Model for Molecular Graphs. *arXiv e-prints* **2018**, arXiv:1802.05283.
 38. Jin, W.; Barzilay, R.; Jaakkola, T., Junction Tree Variational Autoencoder for Molecular Graph Generation. *arXiv e-prints* **2018**, arXiv:1802.04364.
 39. Simonovsky, M.; Komodakis, N. In *GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders*, Cham, Springer International Publishing: Cham, 2018; pp 412-422.
 40. Hansen, K.; Montavon, G.; Biegler, F.; Fazli, S.; Rupp, M.; Scheffler, M.; von Lilienfeld, O. A.; Tkatchenko, A.; Muller, K. R., Assessment and Validation of Machine Learning Methods for Predicting Molecular Atomization Energies. *J. Chem. Theory Comput.* **2013**, *9* (8), 3404-3419.
 41. Strang, G., *Linear Algebra and Its Applications*. Thomson, Brooks/Cole: 2006.
 42. Harris, C. R.; Millman, K. J.; van der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; Kern, R.; Picus, M.; Hoyer, S.; van Kerkwijk, M. H.; Brett, M.; Haldane, A.; del Río, J. F.; Wiebe, M.; Peterson, P.; Gérard-Marchant, P.; Sheppard, K.; Reddy, T.; Weckesser, W.; Abbasi, H.; Gohlke, C.; Oliphant, T. E., Array programming with NumPy. *Nature* **2020**, *585* (7825), 357-362.
 43. Desaphy, J.; Bret, G.; Rognan, D.; Kellenberger, E., sc-PDB: a 3D-database of ligandable binding sites-10 years on. *Nucleic Acids Res.* **2015**, *43* (D1), D399-D404.

44. Naderi, M.; Govindaraj, R. G.; Brylinski, M., eModel-BDB: a database of comparative structure models of drug-target interactions from the Binding Database. *Gigascience* **2018**, 7 (8).
45. Landrum, G., RDKit: Open-source cheminformatics. **2006**.
46. Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B., ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res.* **2012**, 40 (D1), D1100-D1107.
47. Trott, O.; Olson, A. J., AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *J. Comput. Chem.* **2010**, 31 (2), 455-461.