Supplementary matrial for VISAR: a useful strategy in dissecting chemical features learned by neural networks for quantitative QSAR modeling

Qingyang, Ding; Songpeng, Zu; Siyu Hou; Yonghui Zhang; Shao Li

September 3, 2019

1 Supplementary Methods

1.1 Datasets

The compound-protein interaction (CPIs) were extracted from the ChEMBL database Version 23 [1]. ChEMBL is one of the biggest open database of drug-like bioactive compounds, with the data manually or automatically annotated from a large number of scientific literature, presenting in a unified format and confidence scores of the data quality. We filtered the dataset for target type as SINGLE PROTEIN, organism as Homo sapiens, assay type as B (for binding), standard type as Ki or IC50, and confidence score larger than 7 (with 9 the highest, 7 means the data is reliable).

The CPI datasets of family A GPCRs and protein kinases were used in this study. Both of them are classic drug targets with many previous pharmacological studies, sufficiently large CPI recording entries with targets testing against structural diverse compounds, making it possible for us to evaluate and interpret the trained models by referring to related chemoinformatics, medicinal chemistry and structural biology works. The protein classification table of ChEMBL was used for the selection of the single protein targets belonging to "small molecule receptor (family A GPCR)" and "Protein Kinases" on level 3, corresponding to the subclass of family A GPCR and kinases respectively. Dataset of size less than 100 were dropped, leaving 54 GPCRs with more than 70000 CPIs and 29 kinases with more than 7000 CPIs as the training and validation dataset used in this study (See supplementary for the detailed summary of datasets).

In the case of CPIs with multiple recordings (same target-ligand pair tested in different experiment batches), a further filtering were applied – if the standard deviation of the affinities in multiple recording was bigger than the cutoff (in this study we set it to 0.5), these recordings were considered inconsistent and discarded, while for the consistent ones, the mean of them was used to represent the CPI strength.

Before training QSAR models, the binding affinities of CPIs in ChEMBL were transformed. The negative logarithm of the original values (unit nM) times $1e^{-9}$ were lower clip to 4 and then shifted to $-\epsilon$.

To further validate the performance of our model, external datasets from DUDE [2] was also used in our study. DUDE is a famous database of decoys for the benchmark of virtual screening studies, and in this study our trained models were tested by 4 datasets for GPCRs and 11 datasets for kinases extracted from DUDE, with half of compounds in testing sets as decoys.

1.2 Compounds and Descriptors

In an effort to obtain valid and clean compound data for model training, compounds with molecular weight larger than 1000 were filtered out; salt removal and charge neutralise were also carried out for each compound. Deepchem package [3], an open-source toolchain for deep-learning in drug discovery, was adopted for dataset mangement, feature extraction. The featurizer 'circular 2048' and 'circular 1024' of deepchem were used as the descriptors of compounds for both baseline models and neural network models. The open-source software RDKit, one of the chemoinformatics libraries and toolkits [4], was then used for the interpretation of circular fingerprints (built by applying Morgan algorithm thus was called Morgan fingerprints [5] in RDKit). Circular 2048 and Circular 1024 corresponds to radius 2 and 1 for Morgan fingerprint settings respectively. RDKit.Chem.Draw functions were also applied for the compound visualization. PandaTools along with descriptor mapping function of RDKit was applied to calculate physiological properties of the compounds.

1.3 Model

Suppose we have J proteins and for each protein j, $j = 1, 2, \ldots, J$, we have the data $D_j = \{\mathbf{X}_j, \mathbf{y}_j\}$, $\mathbf{X}_j \in \mathcal{R}^{n_j \times p}$, and $\mathbf{y}_j \in \mathcal{R}^{n_j \times 1}$. Each row of $\mathbf{X}_j$ represents one compound with the chemical fingerprint features of p dimensions. y_{ji} records transformed value of the binding affinity (unit as nM) of the compound i against the protein j, $i = 1, 2, \ldots, n_j$.

1.3.1 Ridge regression model optimized by cross-validation

For each protein j, the ridge model coefficients minimize a penalized residual sum of squares, and leave-one-out cross-validation was applied to determine the alpha for each model:

$$\min \|X_j\omega_j - y\|^2 + \alpha_j\|\omega_j\|^2 \tag{1}$$

In this work, we applied RidgeCV from sklearn [6] linear model package, with 20 α to try, starting from -1 and ending at 2.

1.3.2 Support vector regression

Linear support vector regression(SVR) was also used to train quantitative predictive models for each protein j respectively. Given $X_j \in \mathcal{R}^{N_j \times p}$ as fingerprint of ligands that tested against the protein j, and y_j as the corresponding $log(IC_{50})$ value, the training data $(X_j^1, y_j^1), ..., (X_j^{N_j}, y_j^{N_j}) \subset \chi \times \mathcal{R}$, where χ denotes the space of the input space patterns. we have linear function in the form of:

$$y_j = X_j\omega_j + b \tag{2}$$

with $\omega \in \chi, b \in \mathcal{R}$, flatness means a small w is required. Thus it equals to the convex optimization problem:

$$\text{minimize } \frac{1}{2}\|\omega\|^2 + \mathbf{C}\sum_{i=1}^{N_j}(\xi_i + \xi_i^*) \tag{3}$$

$$\text{subject to } \begin{cases} y_j^i - X_j^i\omega_j - b \leq \epsilon + \xi_i \\ X_j^i\omega_j + b - y_j^i \leq \epsilon\xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \tag{4}$$

In this work, we applied LinearSVR from sklearn.svm package, with $\mathbf{C} = 1.0, \epsilon = 0.2$.

1.3.3 Single task neural network

In this study we use the feedforward (artificial) neural network (ANN) as the basis of our approach. For input descriptors $X_j = [x_1, \ldots, x_{N_j}]^T$, they are fed to neurons associated with weights $W^{[1]}$ of layer 1, a bias term b and activation function $f(z)$. After goes through a neuron, the output is:

$$A_j^{[1]} = f(W^{[1]} X_j + b^{[1]}) \tag{5}$$

Then for subsequent layers:

$$A_j^{[l]} = f(W^{[l]} A_j^{[l-1]} + b^{[l]}) \tag{6}$$

The rectified linear unit (ReLU) function was used as the activation function in this study. The typical cost function of root mean squared error (RMSE) was the training object. A series of hyperparameters regarding the network architecture and training strategies were tested, including:

- the number of hidden layers
- the number of neurons in each hidden layer
- the percentage of neurons to drop-out during training

Each dataset for a specific target was trained separately using backpropagation. The model was implemented by Tensowflow [7], Keras [8] and DeepChem package. The Adam optimizer was applied with standard settings, the default learning rate was set to 0.001, and the default size of each batch was 128. HyperparamOpt function of DeepChem package was used for hyperparameter screening.

1.3.4 Multitask neural network

Compared to single neural network, multitask models are trained for multiple objects simutaneously. The architecture of the multitask model could be diverse [9], and the objects selections could also be delicate in order to balance the data sparsity and useful information for sharing.

Here we adopted the a straight forward implemmentation of multitask hard neural network, attaching the logP value and a measurement for compound structure complexity (the descriptor BertzCT in RDKit) to the last layer as additional output nodes, and keeping the rest of the model architecture the same. For this model architecture, there's no missing value for all objects in dataset. This implementation was only meant for illustration of chemical landscape analysis, rather than optimizing predictive performance.

Additionally, the hybrid bypass architecture proposed by Ransundar et al. [9] was also adopted, referred to as the 'RobustMT' training mode.

1.4 Model evaluation

When evaluating the model performance, we randomly partition the data into 100 samples validation set and the rest as training set. For quantitative prediction, we applied the root-mean-square error (RMSE) to compare the predicted results to real data.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y_i}^2)}{n}} \tag{7}$$

In which, n was the test sample number, y_i was the true value of the sample i, and $\hat{y_i}$ was the predicted value. We also use Pearson correlation as evaluation endpoint:

$$\rho_{Y,\hat{Y}} = \frac{\mathbf{cov}(Y, \hat{Y})}{\sigma_Y \sigma_{\hat{Y}}} \tag{8}$$

Where Y is the actual value of y as vector, and $\hat{Y}$ is the predicted one.

For external dataset performance evaluation, the area under the receiver operating characteristic curve (AUC) was used as the evalution metrics, since the DUDE dataset doesn't have quantitative binding values but instead a label of active or decoy.

Data processing was carried out in python, while figure drawn using R package ggplot2 [10]. The RMSE, pearson correlation, and AUC calculation was done by sklearn.metrics package.

1.5 Chemical landscape visualization and model interpretation

Transformed features of the chemical features learned on the hidden layers were calculated, and the value dimensions were further reduced using PCA (down to 20) and then tSNE, thus we can visualize the chemical landscape and their evolving pattern as the layer went deeper on 2D plots. The calculation of PCA and tSNE were also done by sklearn packages.

In order to understand how the chemical substructures contributes to their binding affinity from the trained neural network model, tensowflow.gradients function was used to get the derivatives of each fingerprints, and then these derivatives were mapped back to the molecule structures as weights. By applying color map to the weights, we managed to visualize molecules directly on each atom with the help of RDKit drawing functions.

1.6 Pharamcophore building

The pharmacophore-based tool align-it [11] was applied for our pharmacophore related analysis. The open-source tool of align-it is capable of representing pharmacophoric features of the compounds as Gaussian 3D volumes. The analysis involves several steps as follows:

- generate the 3D conformation of the chemicals with RDkit using MMFF94 force field.
- extract pharmacophoric features of the chemicals using align-it. The features include aromatic and lipophilic, hydrogen bond donor and acceptor, as well as charge centers.
- align previously generated pharmocophores and calculate similarities also using align-it. The similarity was calculated based on the volume of overlap between pharmacophores, and TANIMOTO measure was adpoted,

$$\text{TANIMO} = \frac{V_O}{V_A + V_B - V_O} \tag{9}$$

2 Supplementary Results

2.1 The rationale of empowering CPI analysis with a deeper understanding of neural network

Firstly we argued that by understanding of the intermediate layer of deep neural network model trained with molecular fingerprint, we could have both the plausible performance in making quantitative CPI prediction, and interpretable insights in drug design. Useful information for CPI prediction is not just about the predictive power, but also to learn the 'binding modes' of ligands and their targets – for the purpose of the latter, deep neural network model with circular fingprints could be pratical and yet helpful.

As showed in Fig 1A, the 4 ways of molecule representation – the circular fingerprints, pharmacophores, 3D molecular conformation in binding pocket and the transformed features of intermediate neural network layer – depicted the 'binding mode' from different perspectives. Morgan fingerprint (Fig 1a) is one of the most widely used molecular discriptor in virtual screening. Through iterative algorithms, local substructures of chemicals are encoded

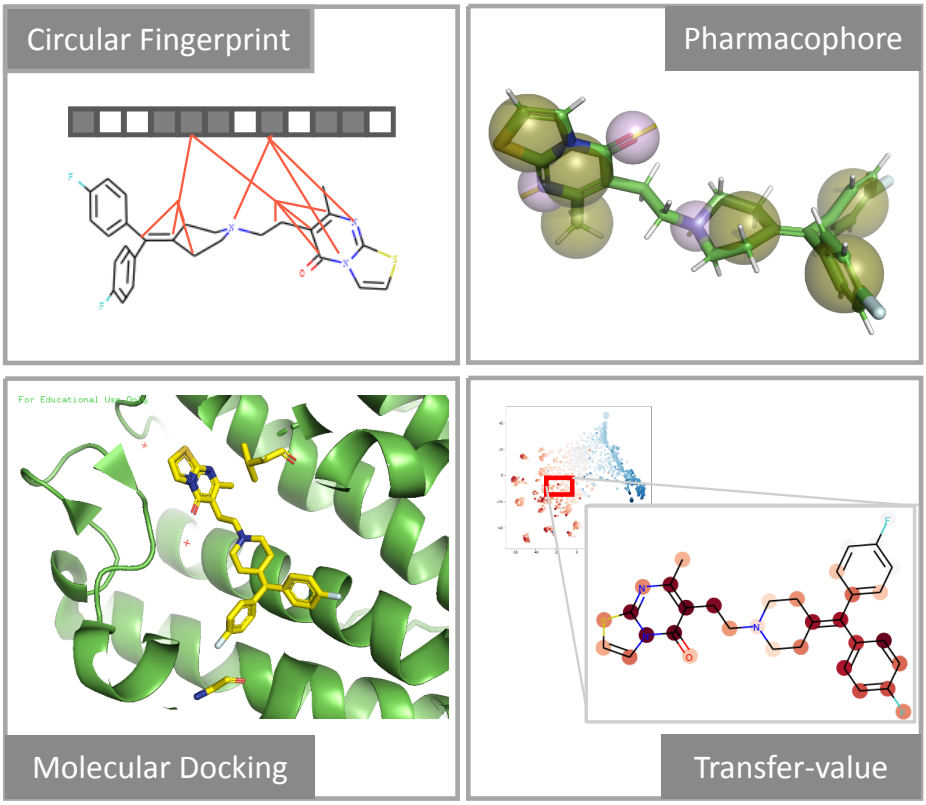

Figure 1: Illustration of different perspectives in understanding the binding mode. Ritanserin, which is a 5HT2c specific inhibitor, served as a representative of the chemicals. Fig1a is a schematic diagram of the substructure embeded as Morgan Fingerprint bits. Fig1b is the pharmacophore, with yellow and pink spheres representing aromatic and lipophilic sites and hydrogen bond acceptors respectively. Fig1c is the stucture of 5HT2c - ritanserin complex (6BQH solved by Peng et al. [12]). The interactions between RIT and G218$^{5.42}$ and V354$^{7.39}$ were identified as key for selectivity. Fig1d is an example of chemical landscape and the zoomed-in SAR of ritanserin regarding a specific target. The red color in landscape indicates higher binding affinity while blue color lower activity; the red box on the landscape shows the position of ritanserin; the dark red color mapping on the chemical structure is related to stronger contribution to the binding affinity of ritanserin.

and folded to bit positions of fixed length of 1024 or 2048. Pharmacophores (Fig 1b) are defined steric and electronic features with specific geometry that are known as important tools in de novo drug design [13]. In the framework of align-it, each pharmacophore is modeled as a 3D spherical Gaussian volume (with center position, the spread and the information on geometry orientation). For 3D molecular conformation in binding pocket (Fig 1c), the interaction surface of target binding pocket and ligands were showed, allowing people to identify the specific residue and functional group that contribute to the binding. This type of features are not always available for any chemical or target of interest, and embedded with the assumption that the crystal structures of the target is the same as that of the physiological states. While pharmacophore and molecular docking manage to provide rigid 3D shape description infered from target structure and part of the active ligands, circular fingerprints as binary vectors are more suitable as the input for similarity calculation and predictive model without further limitation and interpretable power.

As illustrated in Fig 1d, 2 steps were carried out to bridge the gap between fingerprint indices and 3D binding mode with the help of neural network models. Firstly we take the transformed features out of intermediate neural network, reduce dimension to 2 using PCA first and then tSNE in order to visualize the distribution of all chemicals in the dataset. As shown in the upper left figure, chemicals in the dataset appears to be arranged according to their binding affinity and structure similarity spontaneously, forming a smoothed chemical landscape based on the engineered features. Through building the landscape, we could view clusters of active compounds and grasp the complexity of the already tested chemical space against the target of interest from a global perspective. Then we zoom in for a specific chemical and calculate the derivative of each fingerprint and map the gradient back to each atom. Since in Morgan fingerprint, the same bit might refer to more than one substructures due to the hashing mechanism [14], the mapping may include noises but sufficient in serving the purpose of visualing SAR and indicating key functional groups. In the lower-right figure of ritanserin, one of the 4-fluorophenyl and thiazolopyrimidine were labeled as positively contributing to the binding, which is consistent with the finding in structural studies showed in Fig 1d. This kind of SAR didn't involve any information from the target, but succeeded in narrowing down the key functional groups participating in the interaction.

In summary, we showed in this section that Morgan fingerprint, pharmacophore and 3D molecular conformation in binding pocket are in fact different ways of depicting the binding between chemicals and targets of interest. Transformed-feature based landscape and visualized SAR could serve as bridges that translate what have been learned from predictive neural network models into human interpretable insights.

2.2 The performance compared with base models

Next we set out to build neural network models and showed that they have plausible predictive power and the generalization ability. Since as indicated by previous work, fairly shallow neural network structure would be rather sufficient for single task QSAR model [15], here we took a deep look at only the 2 or 3-layer model and focussed specifically on various pyramid architectures. Based on hyperparameter screening (Supplementary Fig 2 and Table 1), we chose the neuron network model architecture for the subsequent training and analysis.

For the next step a series of CPI datasets extracted from CHEMBL targeting proteins from GPCR family A and kinases were trained, and the performance of these models were compared to baseline methods support vector regression (SVR) and ridge regression (Rig). Supplementary Figure 2 shows that our single task (ST) and multi-task (MT) neural network models were performing marginally better than SVR models and compatible with Ridge models on most of the test sets. The results are summarised in Table 3. A further external validation was carried out using DUDE datasets and pretrained predictive models for kinases. The resulting AUC scores for single task neurtal network models, SVR and Ridge are listed

Table 1: Summary of dataset information

Task	sample_size	full_name	pref_name
T107	2951	Serotonin 2a (5-HT2a) receptor	5-HT2a
T108	2063	Serotonin 2c (5-HT2c) receptor	5-HT2c
T51	3202	Serotonin 1a (5-HT1a) receptor	5-HT1a
T106	808	Serotonin 1b (5-HT1b) receptor	5-HT1b
T105	883	Serotonin 1d (5-HT1d) receptor	5-HT1d
T10618	109	Serotonin 1e (5-HT1e) receptor	5-HT1e
T227	1064	Serotonin 2b (5-HT2b) receptor	5-HT2b
T168	396	Serotonin 4 (5-HT4) receptor	5-HT4
T10624	313	Serotonin 5a (5-HT5a) receptor	5-HT5a
T10627	2523	Serotonin 6 (5-HT6) receptor	5-HT6
T10209	1560	Serotonin 7 (5-HT7) receptor	5-HT7
T8	671	Tyrosine-protein kinase ABL	ABL
T9	201	Epidermal growth factor receptor erbB1	EGFR
T10188	249	MAP kinase p38 alpha	MK14
T10434	383	Tyrosine-protein kinase SRC	SRC
T10980	240	Vascular endothelial growth factor receptor 2	VEGFR2
T11408	117	c-Jun N-terminal kinase 3	MK10
T11451	356	Hepatocyte growth factor receptor	MET
T11636	262	Protein kinase C beta	KPCB
T11638	139	MAP kinase ERK2	ERK2
T10938	385	Tyrosine-protein kinase JAK2	JAK2
T11678	289	Cyclin-dependent kinase 2	CDK2

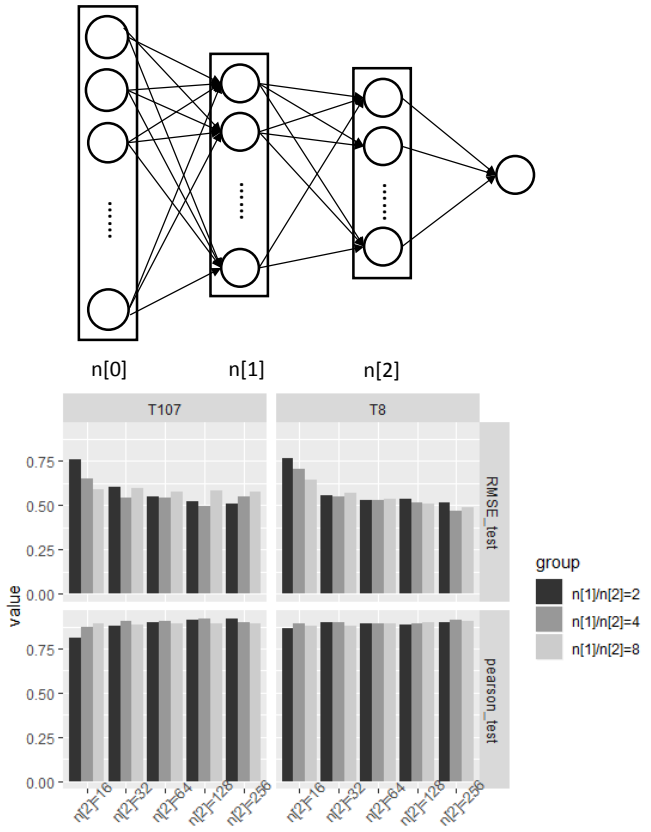

Figure 2: Structure of the neural network model and the test of structural robustness.

Table 2: Parameters of select for neural network model training

Task	**layers**	**dropout ratio**
T107	(512,64)	0.4
T108	(512,128)	0.2
T10209	(512,64)	0.4
T105	(512,128)	0.2
T106	(512,64)	0.4
T10618	(512,128)	0.4
T10624	(512,128)	0.2
T10627	(512,64)	0.2
T168	(512,128)	0.2
T227	(512,64)	0.4
T51	(512,128,64)	0.2

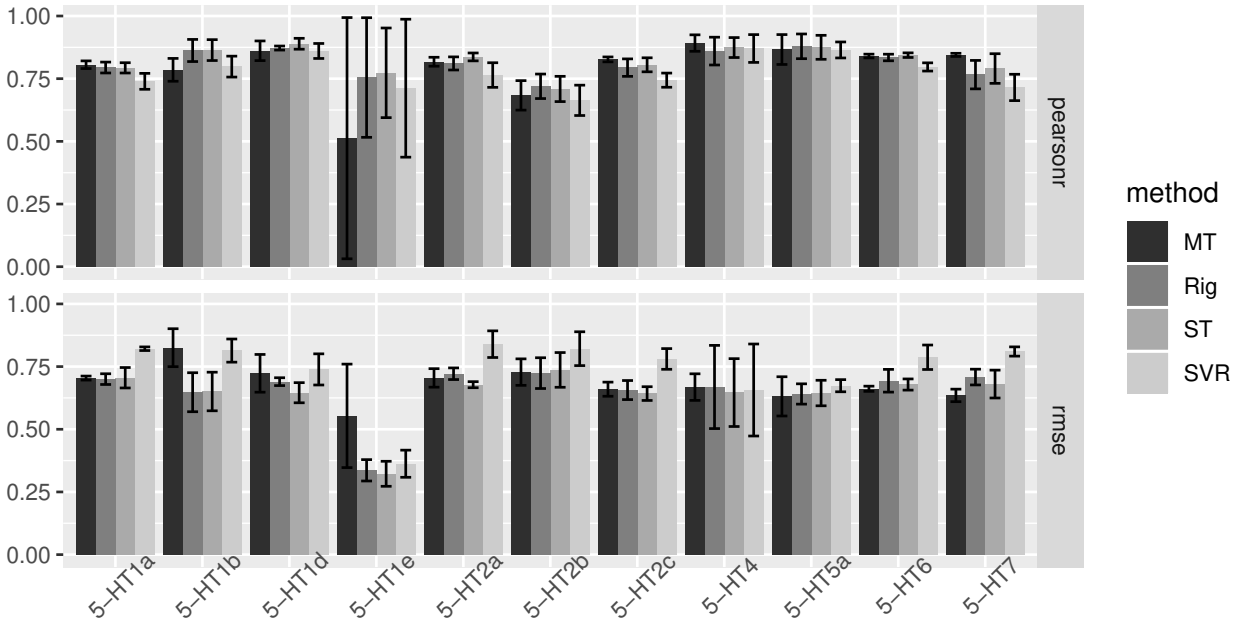

Figure 3: Predictive performance of the neural network model compared with baseline methods.

Table 3: Performance on internal dataset of serotonin receptors

Task	Target	pearsonr				rmse			
		MT	ST	SVR	Rig	MT	ST	SVR	Rig
T51	5-HT1a	**0.81**	0.79	0.74	0.79	**0.70**	0.71	0.82	**0.70**
T106	5-HT1b	0.79	**0.86**	0.80	**0.86**	0.83	**0.65**	0.81	**0.65**
T105	5-HT1d	0.86	**0.89**	0.86	0.87	0.72	**0.65**	0.74	0.69
T10618	5-HT1e	0.51	0.77	0.71	**0.75**	0.55	**0.32**	0.36	0.34
T107	5-HT2a	0.82	**0.84**	0.76	0.81	0.70	**0.68**	0.84	0.72
T227	5-HT2b	0.68	0.71	0.66	**0.72**	0.73	0.74	0.82	**0.72**
T108	5-HT2c	**0.83**	0.81	0.74	0.79	0.66	**0.64**	0.78	0.66
T168	5-HT4	**0.89**	0.87	0.87	0.86	0.67	**0.65**	0.66	0.67
T10624	5-HT5a	0.87	**0.88**	0.86	**0.88**	**0.63**	0.64	0.67	0.64
T10627	5-HT6	**0.84**	**0.84**	0.80	0.83	**0.66**	0.68	0.79	0.69
T10209	5-HT7	**0.84**	0.79	0.71	0.77	**0.64**	0.68	0.81	0.71

Table 4: Performance on external dataset of kinases

Task	Assay	ST	SVR	Rig
T8	Tyrosine-protein kinase ABL	0.63	**0.66**	0.63
T11678	Cyclin-dependent kinase 2	**0.67**	0.66	0.63
T9	Epidermal growth factor receptor erbB1	0.90	0.90	**0.91**
T10938	Tyrosine-protein kinase JAK2	**0.87**	0.77	0.75
T11636	Protein kinase C beta	**0.63**	0.54	0.61
T11451	Hepatocyte growth factor receptor	**0.84**	0.74	0.80
T11638	MAP kinase ERK2	**0.93**	0.90	0.87
T11408	c-Jun N-terminal kinase 3	**0.82**	**0.82**	0.73
T10188	MAP kinase p38 alpha	**0.85**	0.78	0.80
T10434	Tyrosine-protein kinase SRC	**0.89**	0.83	**0.89**
T10980	Vascular endothelial growth factor receptor 2	**0.82**	0.65	0.59

in Table 4. Here we found that ST models had the highest AUC score for most external datasets, and for those that were not the best, ST models didn't fall behind much. This suggested a fair generalization ability of our neural network models.

Similarly with the findings reported by Liu et al. [16], the predictive power of neural networks didn't appear to be far better than baseline models. Rather, the applicable domain as well as the limitation of performance seems to be set by the training datasets. We argued that the though it's probably true that most predictive models overfit and 'reward memorization rather than generation' [17], it's still worth the effort if we could take the full advantage the 'memory' instead of just shooting for better metrics. And that's the reason why we need the strategy to dissect the learned features from QSAR models.

2.3 Neural network derivative-based SAR of compounds showed consistency with key binding sites validated by structural study

As illustrated in Section 2.1, a visualized structure-activity relation (SAR) was one of the helping tool for us to validate what have been learned in neural network models. In supplementary Figure 4 we explored three compounds – ergotamine (ERG), ritanserin(RIT) and lysergic acid diethylamide (LSD) – targeting a series of serotinin receptors, and showed that the neural network derivative-based SAR of these compounds were consistent with the reported structural core binding sites.

ERG is promiscous inhibitor of serotonin receptors, and several structural studies showed that its binding mode is conserved through 5HT1b, 5HT2b and 5HT2c [12, 18]. The ergoline

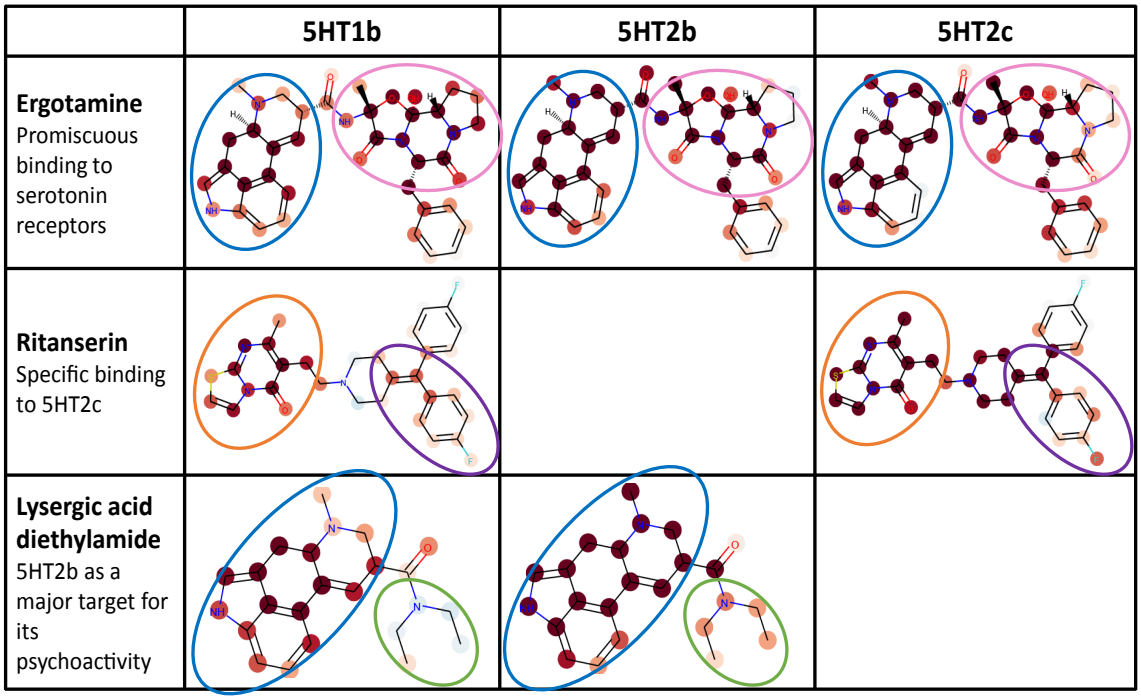

Figure 4: Visualized SAR of ERG, RIT and LSD against serotonin receptors. Blue circle: ergoline core; Pink circle: tripeptide site; Purple circle: 4-fluorophenyl group; Orange circle: thiazolopyrimidine group; Green circle: diethylamide group

core of ERG is recognized by nine key residues from protein binding pockets, serving as the key interaction sites; the ergoline structure in our SAR results was identified as highly postivie contribution. As for cyclic tripeptide and benzyl substituents, only non-specific side-chain contacts are identified, allowing them with more flexibility in interaction; in SAR analysis tripeptide site was also labeled as positive contribution but with differences in detail for 3 targets, and benzyl site is weaker compared with ergoline core and tripeptide sites.

For RIT, the thiazolopyrimidine were marked red for both 5HT1b and 5HT2c, while the aromatic ring, C6 ring and its linker were stronger contributor in 5HT2c binding compared to that of 5HT1b. According to Peng et. al., RITs 4-fluorophenyl and thiazolopyrimidine interacting with respective residues are indeed the primary reason for RIT's 5-HT2 selectivity, suggesting that our SAR succeeded in finding both the key points in RIT-5HT2 interactions.

Similar with ERG, LSD has the ergoline moiety, and the occupation of ergoline core in the same orthosteric pocket is a well-established binding site [18]. Additionally, LSDs diethylamide group binds extended binding pockets and its conformation is identified to be important the potency and activity of LSD at 5HT2b receptors [19], while in the SAR of LSD against 5HT2b, diethylamide group stands out as responsible for the binding.

In summary, figure 4 showed that the mapping of the weights-of-contribution back to each atom of the chemicals gave a clear and insightful mark of the core structure, and could be validated by related structural studies. Baseline models on the other hand failed at picking key binding sites (Supplementary Figure 5 and 6). Without information from the specific binding pocket, visualized SAR derivated from trained neural network models that ultimately learned from quantative binding affinities showed their potential in translating what have been learned from the predictive model to useful understanding of the binding modes.

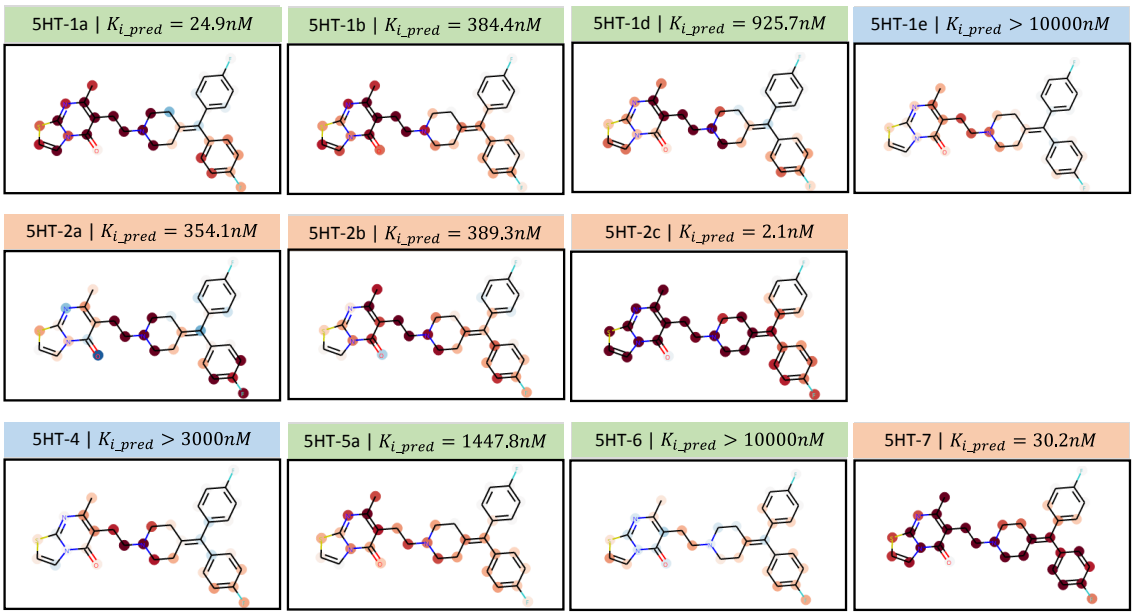

Figure 5: Visualized SAR of RIT based on trained weights of deep neural network.

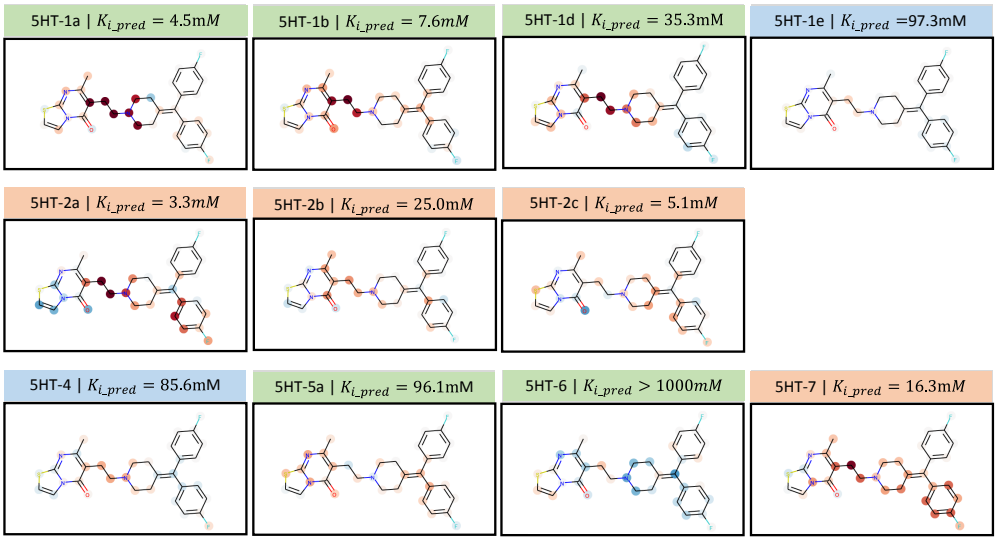

Figure 6: Visualized SAR of RIT based on RidgeCV model coefficients.

2.4 The landscape based on transformed features of the neural network resembles semi-supervised clustering and provide better understanding of the active pharmacophore

With the help of visualized SAR and weights-of-contribution for each atom in each chemical, we further suggested that conventional pharmacophore models could benefit from this helping tool and re-frame our understanding of the chemical landscape. We will next seek to illustrate this point by firstly showing the landscape constructed by transformed features of pretrained neural network, and then zoom in and inspect the clusters on the landscapes.

The formation of a smoothed chemical landscape resembles semi-supervised clustering process, but with quantitative datapoints as the label to learn from, more detailed. Before model training, the clusters on the landscape were mixed with high (red color points) and low activity (blue color points) compounds, forming 'activity cliffs'. As the learning steps increases, the similarities between compounds would no longer depend only on the original fingerprint but on weighted new features, so that the landscape would smoothed where the chemical features correlate better with the biological activity [20]

An example of chemical landscape using transformed features of neural network model (Fig 7a) and Morgan Fingerprint (Fig 7b) trained for target 5HT-2a was shown, with pink shades of the points indicating strong binding affinity and light blue shades weak binding affinity. Since for Morgan Fingerprint landscape, the clustering of chemicals was purely dependent on structure similarity, clusters may be mixed with both active and inactive chemicals of similar structure, and the clusters are far apart from each other. As for transformed-feature landscape, the distance between the chemicals was re-defined based on engineered features learned from the quantitative predictive model, and therefore the exact same cluster on Morgan Fingerprint landscape (labeled as blue points in Fig 7a) was splitted into two on transformed-feature landscape, one with higher binding affinity, another one branching towards other direction.

The distribution of compounds is schematically showed in Fig 7c and 7d, with blue color representing low binding affinity clusters of chemicals, and pink/orange/red color correlated with stronger binding affinity. While on Morgan Fingerprint landscape the clusters were more independent of each other (discretely arranged after dimension reduction by tSNE), on transformed-feature landscape chemicals were arranged according to their binding affinity, shaping a smoothed lanscape along which structures 'evolved from inactive to active' based on rules learned from neural network models.

The clusters of active compounds could be used to build pharmacophore models in a classic process, but the resulting models are with relative weights on each pharmacophoric feature. Instead of manually picking rules for 'active' [13], now with the landscape and SAR in mind, we know that each pharmacophore model is strictly defined within a local applicable domain, and across the landscape an accurate number of such model could be identified and used for guiding de novo drug design. Figure 8 showed two examples of the local pharmacophore model. In the upper panel, two groups of the aromatic and lipophilic sites were identified as common pharmacophore of the selected cluster of compounds, but according to SAR visualization of these compounds, we would notice that the two groups were not equal in contributing to the binding. Benzoisothiazol or benzoisoxazol groups (left part of the pharmacophore) together with the piperazine linker structure (hydrogen bond donor in the middle) were labelled as common positive contributor, while the quinoline structure and other substitute groups (right part of the pharmacophore) were weaker contributors. For the lower panel, similar pattern exists, with methyl-pyrazole group along with methoxyphenylurea recognized as responsible for the binding affinity (corresponding to the left part of the pharmacophore). Other functional groups, though commonly seen in this cluster, were not as core as them. These observations provide powerful insights in distinguishing core binding sites and other sites on the chemicals that are open to modification.

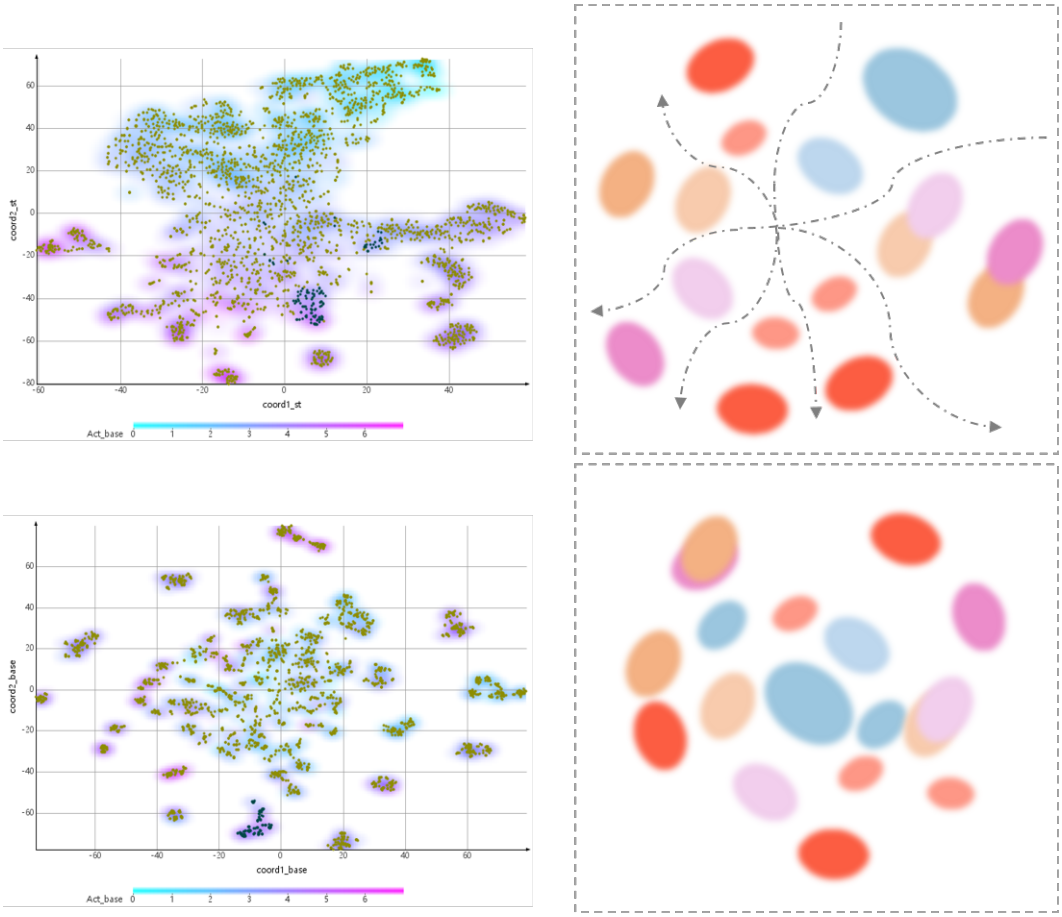

Figure 7: Example of chemical landscapes of 5HT-2a. For scatter plots on the left, pink shading indicates high binding affinity, while cyan shadings indicates low binding affinity. The highlighted blue dots on the two figures are the same group of chemicals. Schematic plots on the right correspond to transformed-feature landscape and original Morgan fingerprint landscape respectively, with blue color indicating low binding affinity and other colors indicating higher binding affinity with possibly various binding modes (could be depicted as various kinds of active pharmacophore model).

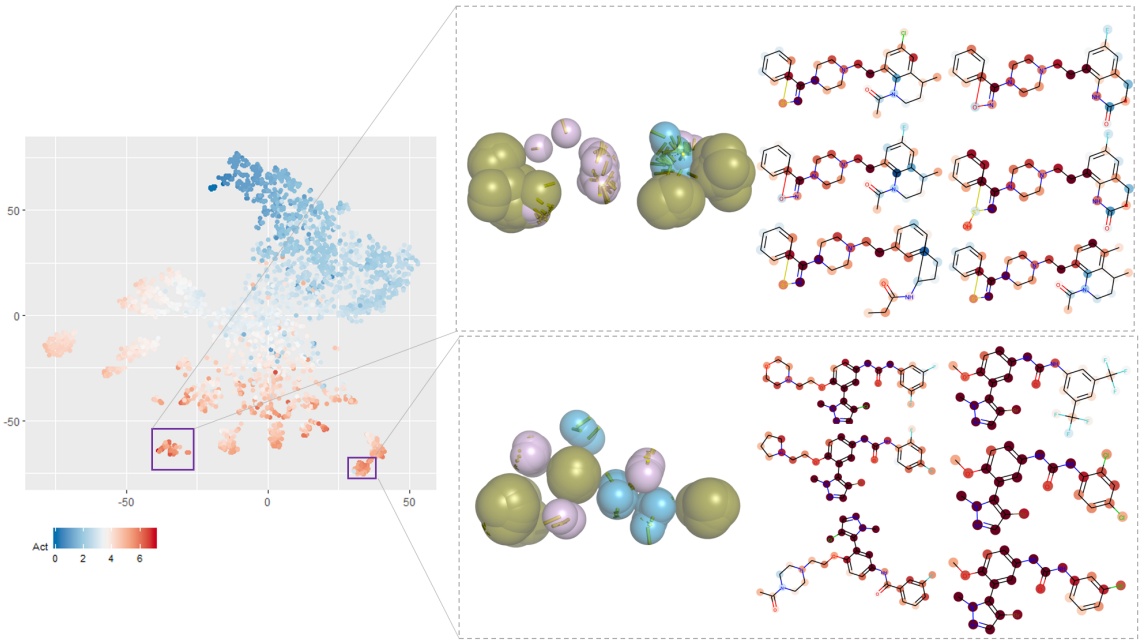

Figure 8: Example of binding modes based on local active area on the chemical landscape of 5HT-2a.

Since this insight only depended on the quantitative CPI information in CHEMBL database without any information on the protein target, we argued that with relatively small CPI datasets this type of analysis is also applicable, and could intuitively guide the design for selective compounds.

3 Usage instructions

The source code and usage instructions for VISAR is available on github.

3.1 Installation

For model training, environment with python=3.5 is recommended, and the environment is depended on: Deepchem, Rdkit, Keras, Tensorflow, Numpy, Pandas, Sklearn, Scipy. The VISAR python package is available for pip install.

```
pip install visar
```

Preparing the working environment for visualization with Conda is recommended.

```
## Create and activate an environment called visar
conda create -n visar python=3.6
conda activate visar

## Install packages via conda
conda install bokeh
conda install -c rdkit rdkit  # Installs also numpy and pandas
conda install -c conda-forge scikit-learn  # Installs also scipy
conda install jupyter  # Installs also ipykernel
conda install -c conda-forge seaborn  # Installs also matplotlib
conda install -c samoturk pymol  # optional
conda install -c conda-forge pmw  # optional
```

3.2 Model training utilities

VISAR package provided users with preprocessed datasets, which include 3060 biochemical assays originated from CHEMBL database, and more than 100,000 compounds. To train the model, users could setup parameters for model and use hyperparameter screening and training functions provided by VISAR package.

The following code is a demonstration of model training with Single Task (ST) model.

```
import os
from visar.model_training_utils_v2 import ST_model_hyperparam_screen,
                                          ST_model_training
# step1: Model parameter setup
task_names = ['T107', 'T108']
MT_dat_name = './data/MT_data_clean_June28.csv'
FP_type = 'Circular_2048'
params_dict = {
  "n_tasks": [1],
  "n_features": [2048], ## need modification given FP types
  "activation": ['relu'],
  "momentum": [.9],
  "batch_size": [128],
  "init": ['glorot_uniform'],
```

```
  "learning_rate": [0.01],
  "decay": [1e-6],
  "nb_epoch": [30],
  "dropouts": [.2, .4],
  "nb_layers": [1],
  "batchnorm": [False],
  "layer_sizes": [(1024, 512),(1024,128) ,(512, 128),(512,64),(128,64),(64
                                          ,32),
  (1024,512,128), (512,128,64), (128,64,32)],
  "penalty": [0.1]
}
RUN_KEY = 'ST_sample' # user specified name
log_path = './logs/'
os.system('mkdir %s%s' % (log_path, RUN_KEY))

# step2: Hyperparameter screening
log_output = ST_model_hyperparam_screen(MT_dat_name, task_names, FP_type,
                                        params_dict,
                                    log_path = './logs/'+RUN_KEY)
# manually pick the training parameters, referring to hyperparam_log saved
                                         in RUN_KEY directory
best_hyperparams = {'T107': [(512,64,1), 0.4],
  'T108': [(512,128,1), 0.2]
}

# step3: Model training
output_df = ST_model_training(MT_dat_name, FP_type, best_hyperparams,
                          result_path ='./logs/'+RUN_KEY)

# step4: Performance testing
from VISAR_model_utils_v2 import generate_performance_plot_ST
import seaborn as sns
plot_df = generate_performance_plot_ST('./logs/ST_2019_8_14_986/
                                        performance_metrics.csv')
g = sns.catplot(x = 'task', y = 'value', hue = 'method',
            col = 'tt', row = 'performance',
            data = plot_df, kind = 'bar')
```

Users could referred to Single task model training template and Robust Multitask model training template.

Next, for visualization, VISAR provided a function to read trained model files and compose dataframes from web application rendering, as demonstrated below:

```
from visar.VISAR_model_utils_v2 import generate_RUNKEY_dataframe_ST
RUN_KEY = 'ST_sample'
log_path = './logs/'
prev_model = log_path + RUN_KEY + '/T107_rep2_50.hdf5'
output_prefix = 'T107_rep2_50_'
task_list = ['T107']
add_features = None
dataset_file = log_path + RUN_KEY + '/temp.csv'
FP_type = 'Circular_2048'

generate_RUNKEY_dataframe_ST(prev_model, output_prefix, task_list,
                          dataset_file, FP_type, add_features, n_layer = 1)
```

3.3 Visualization with web application

To start the interactive application, users need to firstly get the local copy of the VISAR repository by direct dowloading or

```
git clone https://github.com/Svvord/visar.git
```

and then start the web application by

```
cd /path/of/visar
bokeh serve --show VISAR_webapp
```

By default, the web application would use the sample dataframes to render the panel, presenting the interface as Figure 9. The general steps for interactive analysis are

1. Set the location (including the prefix) of the pre-composed dataframes and the mode of your training. After clicking 'Run' button on the upper panel, the whole interface would update according to your settings.

2. Explore the activity profile of the chemical space on the left panel. There are several places allowing for interactive exploring, including: A. color options for the scatter plotting, enabling different color rendering based on eg. different activity of the compounds; B. number of bi-clusters, which correlated with the arrangement of the heatmap on the bottom panel (through trying out different bi-cluster numbers, users could gain an idea of how the activity profile is distributed on the chemical landscape); E. information fo the compounds when hovering your mouse on the scatter plot, displaying its ID, batch ID and the color code for the bi-cluster where it belongs; F. information of the batch when hovering your mouse on the heatmap, displaying its ID and color code for the bi-cluster where it belongs.

3. Upon selecting the batch or indivisual compounds on the left panel, visualize chemical structures along with the SAR pattern on the right panel. There are two ways for batch selection: first is to directly click on the heatmap, second is to use the drop-down list (C). As for compound selection, use the tap mode of the scatter plot and click on the points. Since for RobustMT mode, multiple tasks give their corresponding SAR patterns for the compound; thus by selecting SAR task (D), the SAR pattern of the compounds would update accordingly.

3.4 Pharmacophore analysis

VISAR provided handy function for converting selected dataframe to SDF files:

```
import os
import numpy as np
import pandas as pd
from visar.model_landcape_utils_v2 import df2sdf

compound_df = pd.read_csv('T107_rep2_50_compound_df.csv')
selected_batch = 0
select_df = compound_df.loc[compound_df['label'] == selected_batch]

df2sdf(select_df, 'T107_rep2_50_batch0.sdf', selected_batch = 0)
```

With SDF files of compounds of interest, users could adopt other software for further analysis, including pymol, MOE, DataWarrior, Schrodinger, TeachOpenCADD and etc. VISAR also provide a workflow using Align-it, and the details could be referred to in the jupyter notebook.

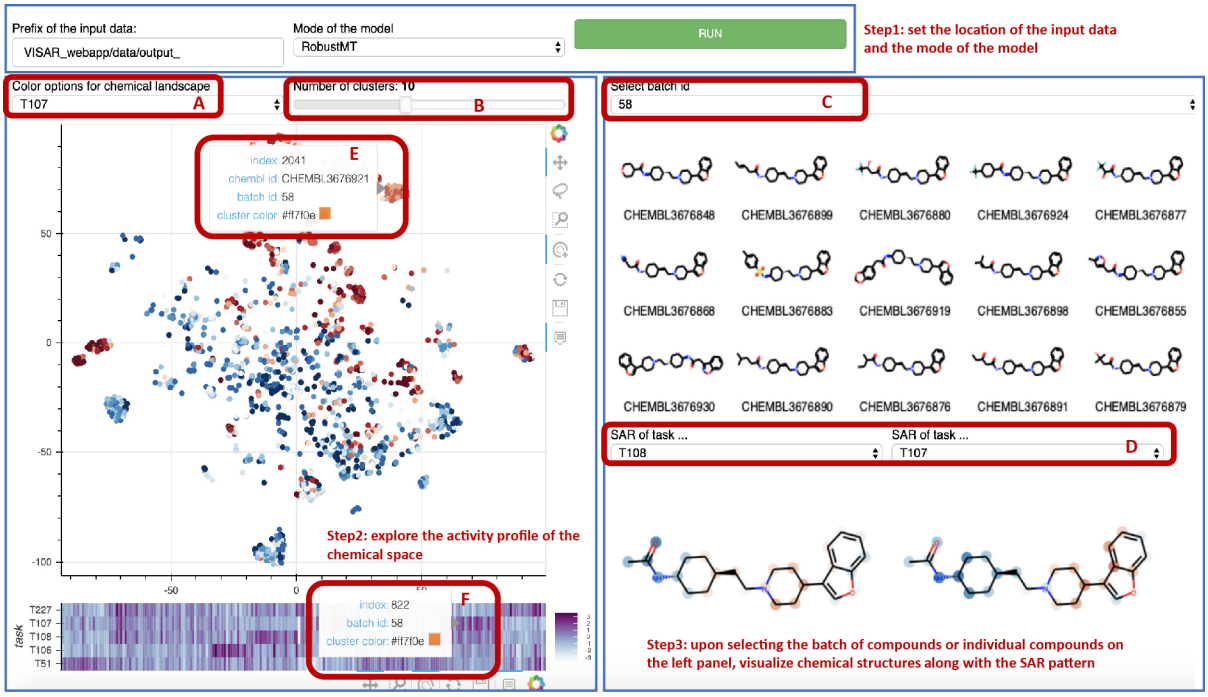

Figure 9: Example of binding modes based on local active area on the chemical landscape of 5HT-2a.

References

[1] A. P. Bento, A. Gaulton, A. Hersey, L. J. Bellis, J. Chambers, M. Davies, F. A. Krüger, Y. Light, L. Mak, S. McGlinchey, *et al.*, "The chembl bioactivity database: an update," *Nucleic acids research*, vol. 42, no. D1, pp. D1083–D1090, 2014.

[2] M. M. Mysinger, M. Carchia, J. J. Irwin, and B. K. Shoichet, "Directory of useful decoys, enhanced (dud-e): better ligands and decoys for better benchmarking," *Journal of medicinal chemistry*, vol. 55, no. 14, pp. 6582–6594, 2012.

[3] B. Ramsundar, P. Eastman, P. Walters, V. Pande, K. Leswing, and Z. Wu, *Deep Learning for the Life Sciences*. O'Reilly Media, 2019.

[4] G. Landrum, "Rdkit: Open-source cheminformatics," *Online). http:// www. rdkit. org. Accessed*, vol. 3, no. 04, p. 2012, 2006.

[5] A. Gobbi and D. Poppinger, "Genetic optimization of combinatorial libraries," *Biotechnology and bioengineering*, vol. 61, no. 1, pp. 47–54, 1998.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[7] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Józefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *CoRR*, vol. abs/1603.04467, 2015.

[8] F. Chollet, "keras." https://github.com/fchollet/keras, 2015.

[9] B. Ramsundar, B. Liu, Z. Wu, A. Verras, M. Tudor, R. P. Sheridan, and V. Pande, "Is multitask deep learning practical for pharma?," *Journal of chemical information and modeling*, vol. 57, no. 8, pp. 2068–2076, 2017.

[10] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2009.

[11] J. Taminau, G. Thijs, and H. D. Winter, "Pharao: Pharmacophore alignment and optimization," *Journal of Molecule Graphics and Modelling*, vol. 27, no. 2, pp. 161 – 169, 2008.

[12] Y. Peng, J. D. McCorvy, K. Harpsøe, K. Lansu, S. Yuan, P. Popov, L. Qu, M. Pu, T. Che, L. F. Nikolajsen, *et al.*, "5-ht2c receptor structures reveal the structural basis of gpcr polypharmacology," *Cell*, vol. 172, no. 4, pp. 719–730, 2018.

[13] S.-Y. Yang, "Pharmacophore modeling and applications in drug discovery: challenges and recent advances," *Drug discovery today*, vol. 15, no. 11-12, pp. 444–450, 2010.

[14] D. Rogers and M. Hahn, "Extended-connectivity fingerprints," *Journal of chemical information and modeling*, vol. 50, no. 5, pp. 742–754, 2010.

[15] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure–activity relationships," *Journal of chemical information and modeling*, vol. 55, no. 2, pp. 263–274, 2015.

[16] R. Liu, H. Wang, K. P. Glover, M. G. Feasel, and A. Wallqvist, "Dissecting machine-learning prediction of molecular activity: Is an applicability domain needed for quantitative structure–activity relationship models based on deep neural networks?," *Journal of Chemical Information and Modeling*, vol. 59, no. 1, pp. 117–126, 2018.

[17] I. Wallach and A. Heifets, "Most ligand-based classification benchmarks reward memorization rather than generalization," *Journal of chemical information and modeling*, vol. 58, no. 5, pp. 916–932, 2018.

[18] C. Wang, Y. Jiang, J. Ma, H. Wu, D. Wacker, V. Katritch, G. W. Han, W. Liu, X.-P. Huang, E. Vardy, *et al.*, "Structural basis for molecular recognition at serotonin receptors," *Science*, vol. 340, no. 6132, pp. 610–614, 2013.

[19] D. Wacker, S. Wang, J. D. McCorvy, R. M. Betz, A. Venkatakrishnan, A. Levit, K. Lansu, Z. L. Schools, T. Che, D. E. Nichols, *et al.*, "Crystal structure of an lsd-bound human serotonin receptor," *Cell*, vol. 168, no. 3, pp. 377–389, 2017.

[20] F. Grisoni, D. Ballabio, R. Todeschini, and V. Consonni, *Molecular Descriptors for Structure–Activity Applications: A Hands-On Approach*, pp. 3–53. New York, NY: Springer New York, 2018.