Deep Learning for the Virtual Screening

Synthesizing drugs can be expensive and complex. Traditional drug synthesis was done using chemical intuition and knowledge, where a scientist search for a promising drug candidate, change functional group and synthesize the said compound. This process could be time-consuming and inefficient and a chemist could spend months in the lab synthesizing compound that

The emergence of high throughput screening (HTS) was thought to be a promising technique to search lead compound[1] [2], HTS is still expensive and some academic labs that work in medicinal chemistry might not have access to HTS technology[1].

For medicinal chemists who have no access to HTS and seek to accelerate their drug discovery process, using virtual screening is an ideal alternative and can be a major time saver.

Deep Learning in Chemistry

In the past decade, deep learning has gained significant traction in many fields including computer vision and natural language processing. With the increasing availability of computing power and software packages, deep learning has become more and more accessible to the mass.

Deep learning also has shown promises in chemistry (and biology) with its stellar performance in QSAR competition and toxicity prediction through Kaggle, a popular data science platform where people can compete and showcase their skills.

Here I will describe a virtual screening experiment using deep learning. This post is meant for chemists who are interested in implementing state of the art technology for their research or are just curious about how machine learning can benefit the field. Given many hypes surrounding machine learning, I do want to point out that algorithms are not perfect and we do need to maintain our "chemist intuition" when using machine learning. Machine learning is a tool, and it is an extremely useful one that sooner or later will be the norm in the field. But like any other tool in the lab, it has its limitation and advantages.

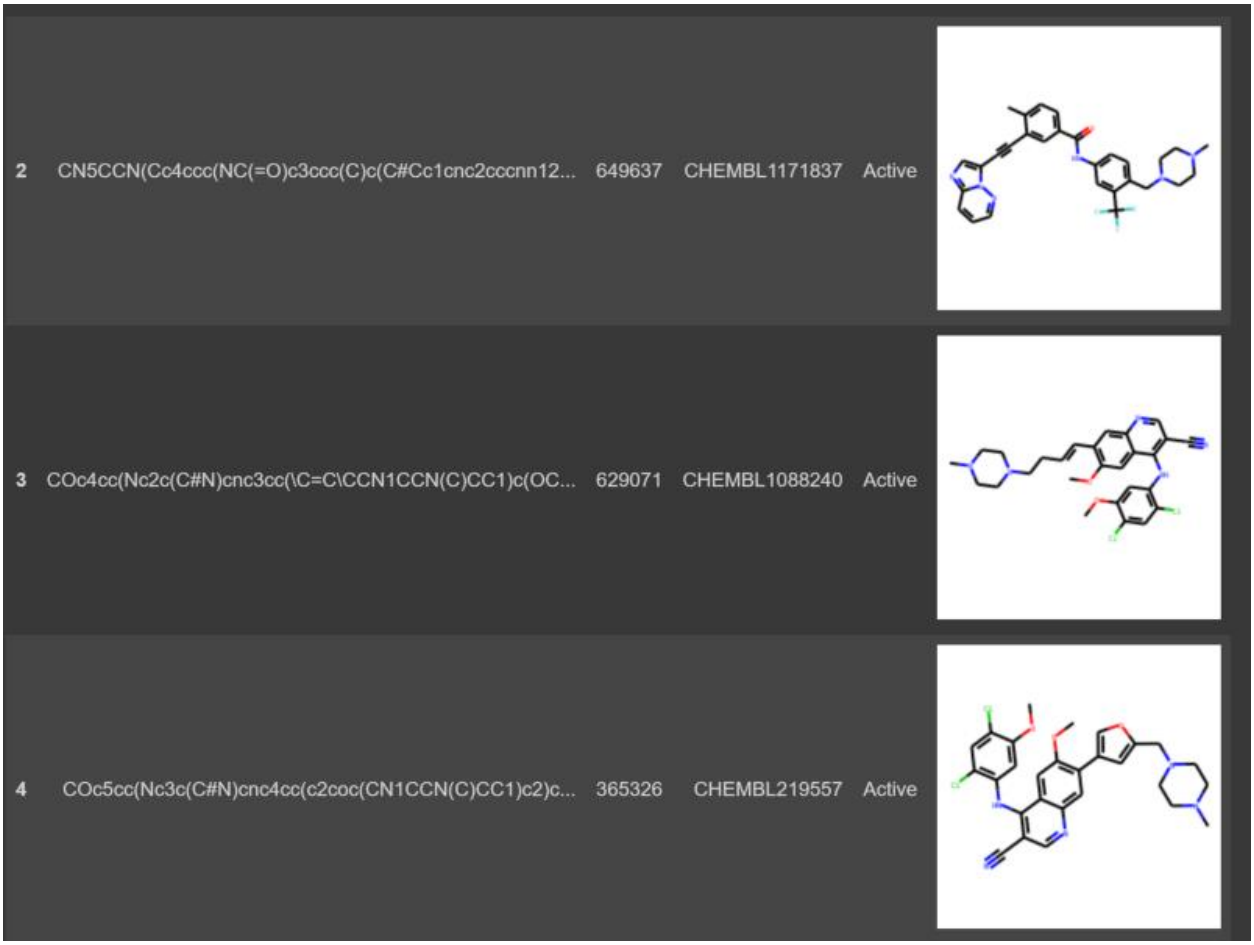
Just because an algorithm produced a certain prediction that does not mean we can follow it blindly. Ideally, when using virtual screening, we can do a docking experiment to further validate compounds that are picked.

As an example of the virtual screening experiment using deep learning, I will use Abl kinase (PDB id: 1FPU), a type of tyrosine kinase. Tyrosine kinase is a fascinating class of protein with a longstanding history of being involved in cancer[3] and recently it was uncovered that it might play a role in neurodegeneration[4]. I have obtained the dataset via DUD-E (http://dude.docking.org/). For more reference, please refer to my GitHub page:

Exploratory Data Analysis

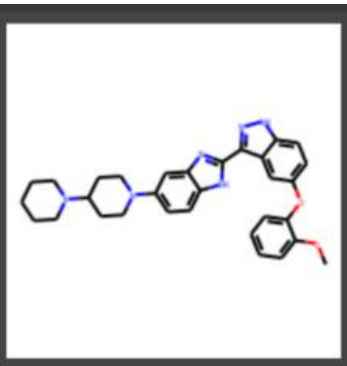
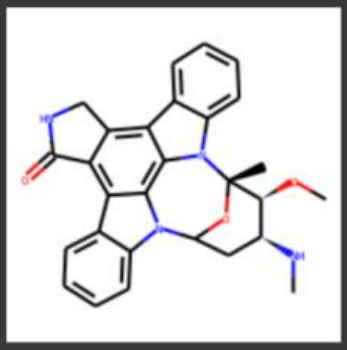
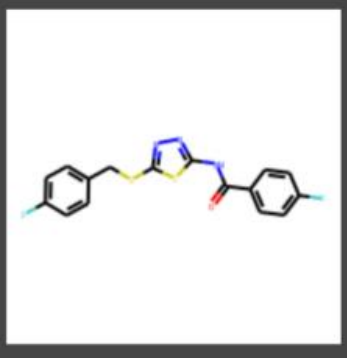
One of the most underrated BUT extremely important parts of any machine learning and data science practice is exploratory data analysis (EDA). Before virtual screening, I will do EDA of the active and inactive molecules.

Below are some of the active compounds. As we can see, the active molecules have aromatic scaffold and polar residues (I saw chlorine, methoxy, trifluoro, cyano). We can cluster our molecules using Butina clustering and then create a similarity map to make more inference



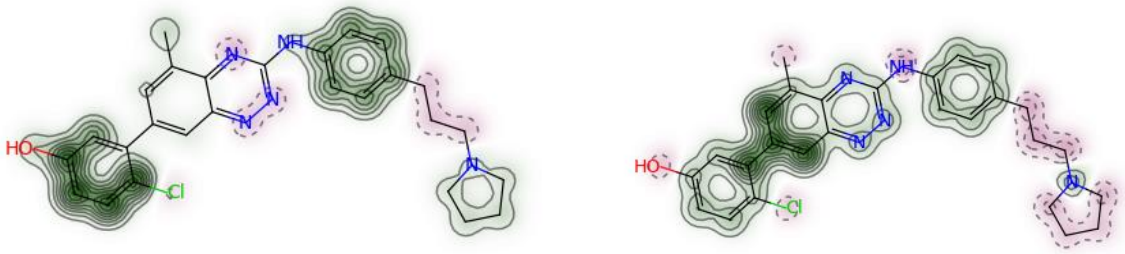
Three active compounds for Abl Kinase

Using Butina Cluster algorithm with 0.9 similarity cutoff, there are 9 unique clusters. Here are some of the clusters

75	COc1ccccc1Oc7ccc6[nH]nc(c5nc4cc(N3CCC(N2CCCCC2...		5
84	CN[C@@H]1CC5O[C@@](C)([C@@H]1OC)n7c2ccccc2c8c3...		7
101	Fc3ccc(CSc2nnc(NC(=O)c1ccc(F)cc1)s2)cc3		4

Three out of 9 unique clusters described by Butina algorithm

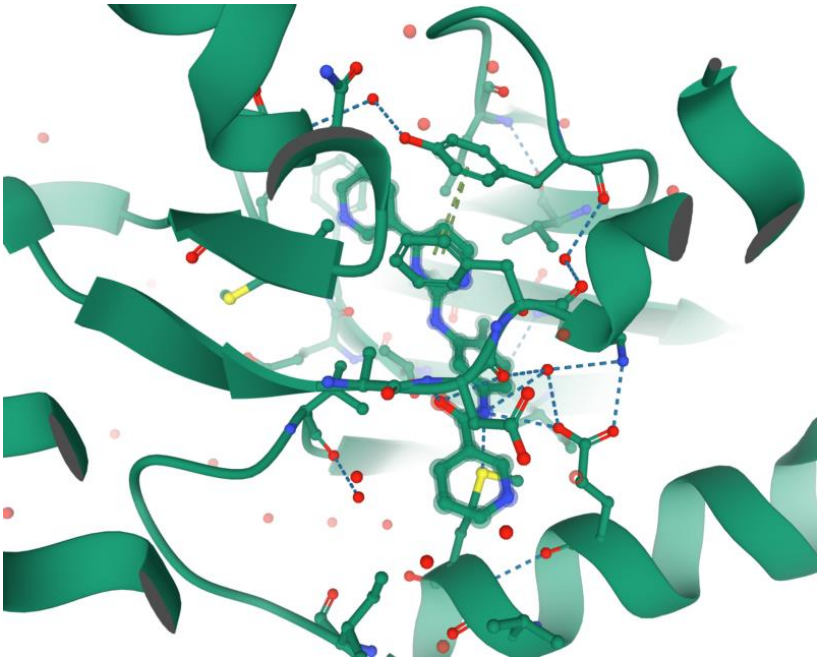
Three out of 9 unique clusters described by Butina algorithm
These clusters seem to have polar residues. We seem can infer that the active site of the protein must have a lot of polar interactions. To have a more concrete similarity, we can construct a similarity map. For the similarity map, I will use the first cluster and compare it with other clusters. For the demonstration, I will show two similarity map (full similarity map can be found through my Github ipynb file):



Similarity map between one cluster with two other clusters

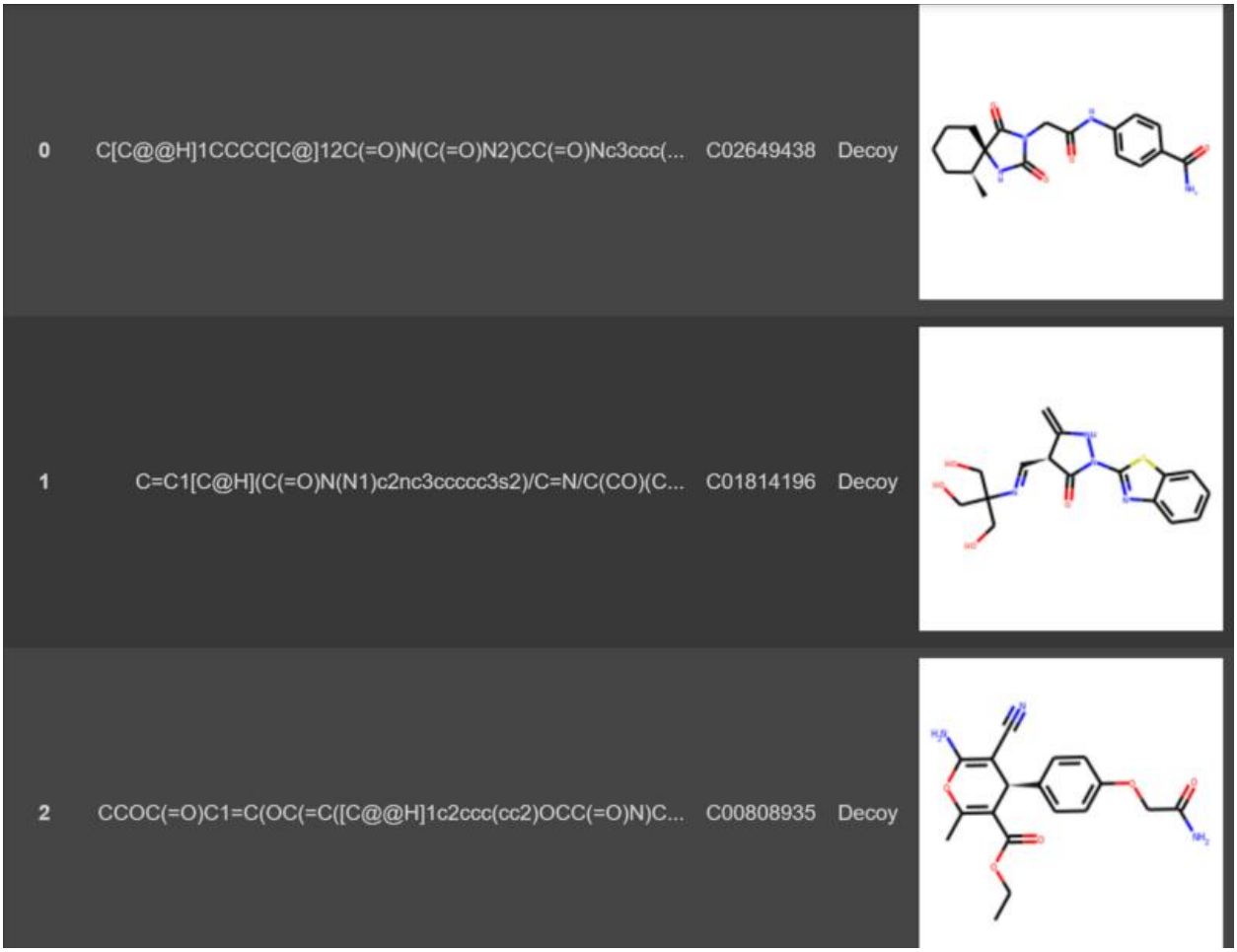
From the above similarity maps (the greener the color, more similarity), it seems that all of our clusters share at least the same one functional group (for example, aniline and all of our phenyl substituent). Our pyrrolidine scaffold seems to be important as well, as a lot of our clusters contain heterocyclic nitrogen compounds.

As mentioned before, our active compounds seem to be polar, to confirm our inference, we can check ligand interaction with our molecule through PDB.



A screenshot of ligand interaction with the protein

From PDB, we can see that a lot of interactions involve polar interaction such as hydrogen bond, pi stacking, and cation-pi interaction. Let's try to take a look at our decoy molecules next:

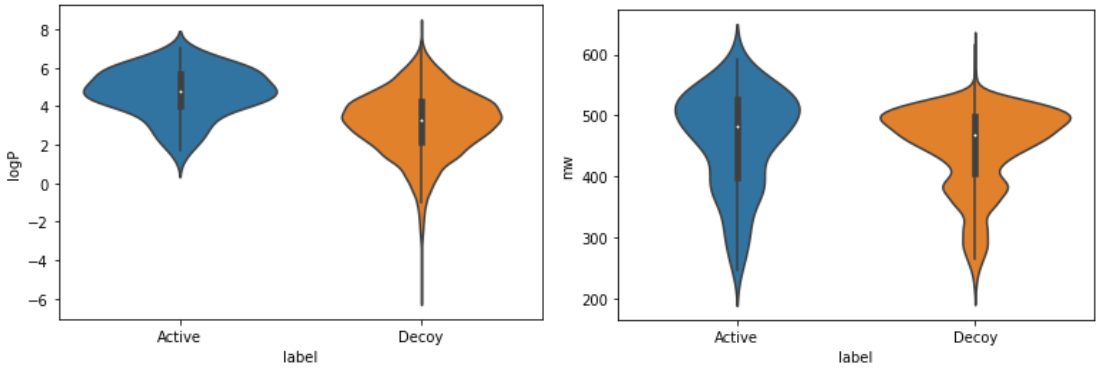


Decoy molecules for Abl Kinase

It seems that our decoy molecules have a lot of polar residues, from above, we can also see functional groups such as cyano, amide, and esters. Sometimes it takes only a single functional group to make a difference. We probably want an algorithm that can distinguish help us distinguish some subtle differences between our active and decoy molecules.

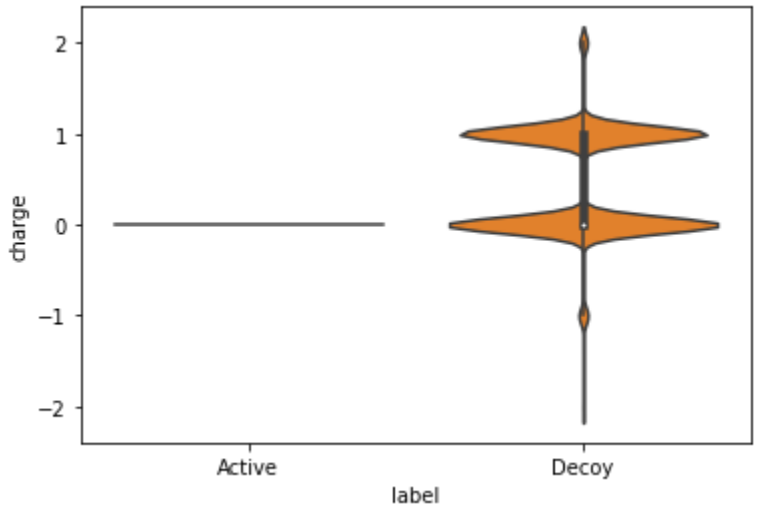
Data Preparation

To ensure that our dataset is ready for modeling, we need to ensure that our active and decoy molecules share similar properties. Visualizing logP and molecular weight:

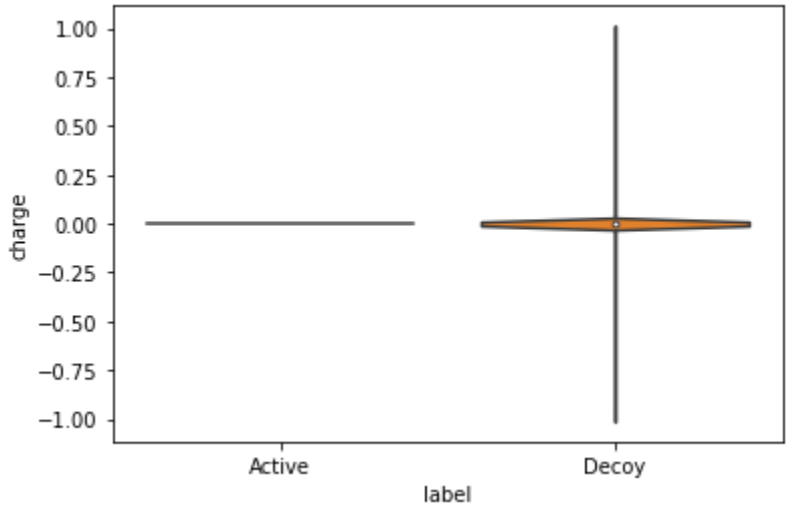


Violinplot of LogP and molecular weight for both active and decoy molecule

However, when visualizing charge we get this:



Luckily, rdkit provide a code that can convert charged molecule to neutral in its SMILES form, after neutralizing the charge:



Training Model

For virtual screening, I will use a weave model and graph convolution for our neural network. Both models seem to be promising for virtual screening. I will use the dude_abl.csv file that I have created during my EDA to train my model. My metric will be ROC-AUC score.

The performance of weave vs. graph convolution is different:

```
-----------------------------------WEAVE RESULT-----------------------------------------
Train score:
{'mean-roc_auc_score': 0.5053541978554302}
Validation score:
{'mean-roc_auc_score': 0.5262478485370052}
-----------------------------------GRAPH CONVOLUTION RESULT-----------------------------
Train score:
{'mean-roc_auc_score': 0.9921662062756098}
{'mean-roc_auc_score': 0.9871574208923606}
```

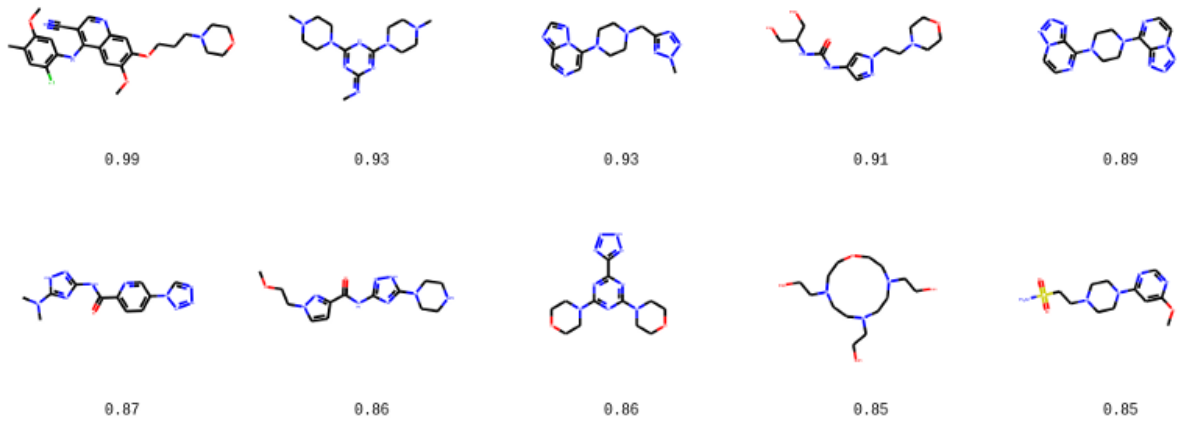
Performance between Weave Model vs. Graph Convolution Model

At first, I was expecting the Weave model to perform better as it does global convolution to all atoms in the molecule.

Since graph convolution performed the best, I will use my graph convolution model for prediction. I obtained a dataset for prediction through Zinc database.

Prediction

This is the result of my prediction:



Top 10 results

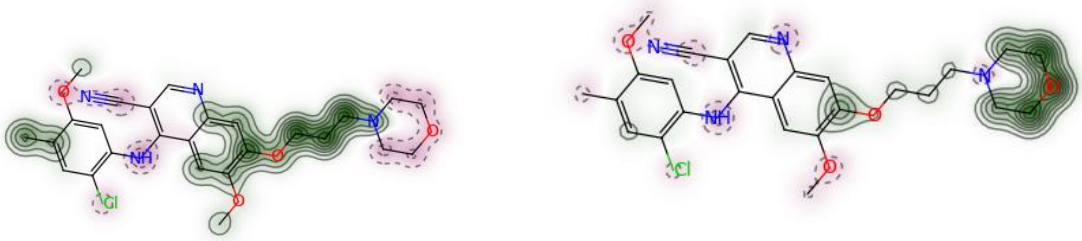
As we can see, the top results seems to be an amalgamation of different active molecules. One of the results (0.85) seem to be a large macrocycle, which is extremely hard to synthesize due to its

low yield. If you don't have access to any HTS in your lab, large macrocycle might be a bit tricky. We can get away with using large dilution, cross-coupling, and even maybe ring closing metathesis.

As we can see, the top results seem to be an amalgamation of different active molecules. One of the results (0.85) seems to be a large macrocycle, which is extremely hard to synthesize due to its low yield. If you don't have access to any HTS in your lab, large macrocycles might be a bit tricky. We can get away with using large dilution, cross-coupling, and even maybe ring-closing metathesis.

We will try to evaluate molecules that have the highest positive value and see if any features are similar with one and another. Remember that it is up to one's judgment whether to use the molecules or not. We could do a docking experiment before synthesizing to get more confirmation regarding the molecule's activity. Alternatively, we can also design our compound and use the algorithm for prediction.

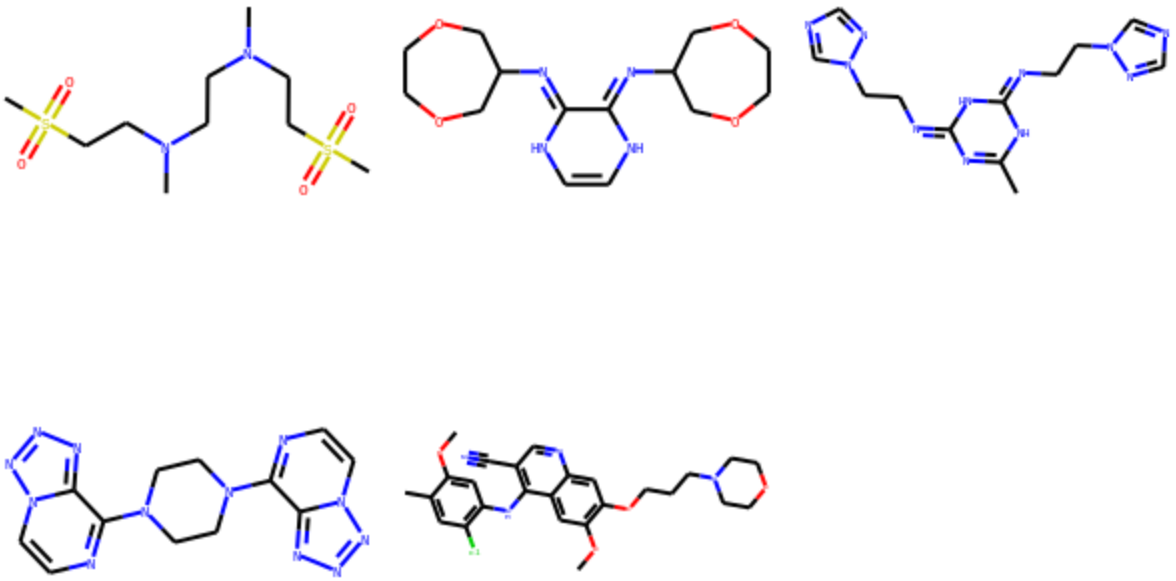
The similarity maps show that the ether group could be important



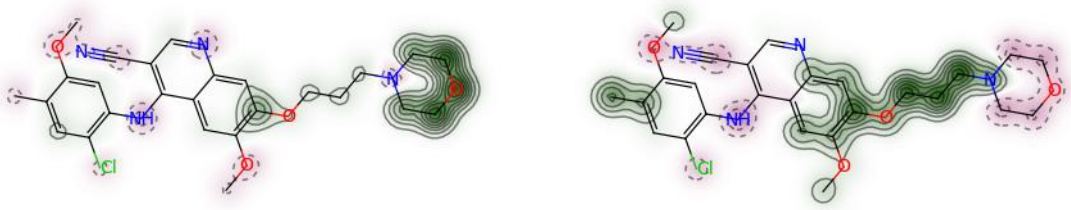
Two similarity maps for comparison

If we look at all similarity maps (available on my ipynb file on GitHub), it seems that the morpholine group might be important. We could also try with similar but different scaffolds such as tetrahydropyran or piperidine.

I tried to cluster top 100 compounds predicted by the algorithm via Butina cluster (0.9 similarities) and found there are 5 unique clusters:



Those are some interesting clusters, especially the one with sulfone. The one with two large macrocycles could be hard to synthesize, we might want to change to another scaffold such as tetrahydropyran. If we perform a similarity map we see a similar trend:



Seems that the morpholine and the ether linkage could be an important feature. As I mentioned, while this algorithm is not perfect, we can potentially validate our prediction using docking

experiment. Once we finalize our prediction and their scores, that's when we can attempt to spend our time synthesizing our targets.

Seems that the morpholine and the ether linkage could be an important feature. As I mentioned, while this algorithm is not perfect, we can potentially validate our prediction using a docking experiment. Once we finalize our prediction and their scores, that's when we can attempt to spend our time synthesizing our targets.

Conclusion

Overall, we can see how deep learning can benefit chemists especially those who work in a lab that does not have access to HTS. We also saw how some predictions might be tricky to synthesize but we could get around that by using a different scaffold.

Hopefully, this will inspire more chemists to use deep learning for their research. Even if you are not interested in actually writing algorithms, it is a good idea to be aware of the up and coming technology.

References

[1]Chan, H. C. S.; Shan, H.; Dahoun, T.; Vogel, H.; Yuan, S. Advancing Drug Discovery via Artificial Intelligence. *Trends in Pharmacological Sciences* **2019**, *40* (8), 592–604.

[2] Szymański, P.; Markowicz, M.; Mikiciuk-Olasik, E. Adaptation of High-Throughput Screening in Drug Discovery—Toxicological Screening Tests. *Int J Mol Sci* **2011**, *13* (1), 427–452.

[3] Schindler, T.; Bornmann, W.; Pellicena, P.; Miller, W. T.; Clarkson, B.; Kuriyan, J. Structural Mechanism for STI-571 Inhibition of Abelson Tyrosine Kinase. *Science* **2000**, *289* (5486), 1938–1942.

[4] Hebron, M.; Moussa, C. E.-H. Two Sides of the Same Coin: Tyrosine Kinase Inhibition in Cancer and Neurodegeneration. *Neural Regen Res* **2015**, *10* (11), 1767–1769.