Predicting the binding affinities between drug compounds and protein kinases using convolutional neural networks

Marcus Schubert

Regis High School, 55 E 84th St, New York, NY 10028

Abstract:

Deregulated protein kinases can cause diseases such as cancer. Discovering drugs to inhibit such kinases is currently inefficient, involving in vitro experimentation and leaving many compounds in trial phases. [2] It is also exceedingly expensive. [1] Other scientists have attempted to solve this problem more efficiently with various forms of machine learning, and I build on their work, using a convolutional neural network to predict compound-kinase interactions. [3] The first phase of my project concerned organizing data. I downloaded a dataset of more than one million drug-kinase interactions from drug target commons and filtered the data I deemed usable into an SQLite database. The strength of the interaction between a kinase and a potential inhibitor was denoted by IC50 values ranging from 0 nM to 10^12 nM with lower values indicating stronger interactions. The data was incomplete and had to be supplemented with amino acid sequences to represent protein kinase structures and standard InChIs and Morgan Fingerprints to represent molecular compound structures from outside sources. My coding was all done with Python, and I used Keras and TensorFlow for neural network implementation. I took different approaches to this project, using multiple neural network structures and different types of inputs to discover which method would prove most effective. First, I concatenated amino acid sequences with InChIs to form one long input and fed them into a neural network with two convolutional layers and three dense layers, achieving at maximum, correct predictions 85% of the time on a randomized test set that was 20% of the training set. Since concatenation of drug structure and amino acid sequence was not an ideal input, and the InChI string was not the best structural representation of a compound, I then modified my approach by using parallel neural network structure and replacing the InChIs with Morgan Fingerprints. The parallel structure ran the fingerprints and amino acid sequences through separate neural networks of two layers each before merging them into a single network, avoiding the issue of concatenation. This implementation achieved an accuracy of 58 to 93%, varying based on the proteins it was tested on.

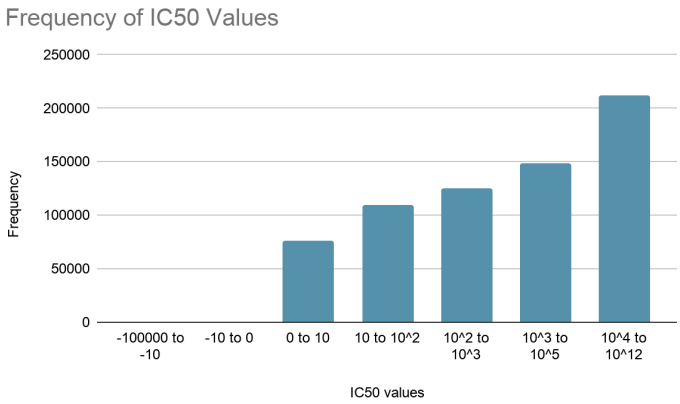
1. Introduction

The purpose of the following experimentation is to find a way to use neural networks to predict which drug compounds will inhibit certain protein kinases. This is an important issue to resolve as deregulated behavior of protein kinases can lead to many diseases, including cancer. Protein kinases are responsible for phosphorylation of proteins, which regulates many cell processes including proliferation, apoptosis, and signal transduction. Deregulated kinase activity can cause the formation of tumors and help cancer cells survive and spread. [4] Drug compounds to inhibit these deregulated proteins need to be discovered to halt the progress of such diseases by binding to the allosteric site of a certain protein kinase and rendering it inactive.[2] Potential inhibitors are traditionally discovered by the in vitro method of high throughput screening, a process that involves acquiring the compounds and having chemists perform assaying work, an expensive and time consuming process. [4] In 2014, a study found that the cost of developing a new drug was $2.87 billion. [1] Even in silico methods such as molecular docking are computationally expensive due to the complex simulations such approaches involve. [5] A successful approach to this problem using machine learning would be exceedingly useful. Other scientists have used methods such as kernel based regression and recurrent neural networks to approach this problem. [3] Building on their research, I chose to use a convolutional neural network, which is a type of machine learning network inspired by neurons in the brain. Since the input data I obtained is extremely long (60,000 to 700,000 rows of almost 2000 character inputs), a convolutional neural network is more computationally efficient than the

dense neural network alternative. Throughout this project, a combination of Keras and Tensorflow were used for modelling, Python was used as a coding platform, and Sqlite was used for database management. Inputs throughout this project consisted of an expression of the drug compound's structure, an amino acid sequence representing the protein kinases, and finally an IC50 value, which will be described later on, that essentially predicts the potency of the drug at inhibiting the protein.

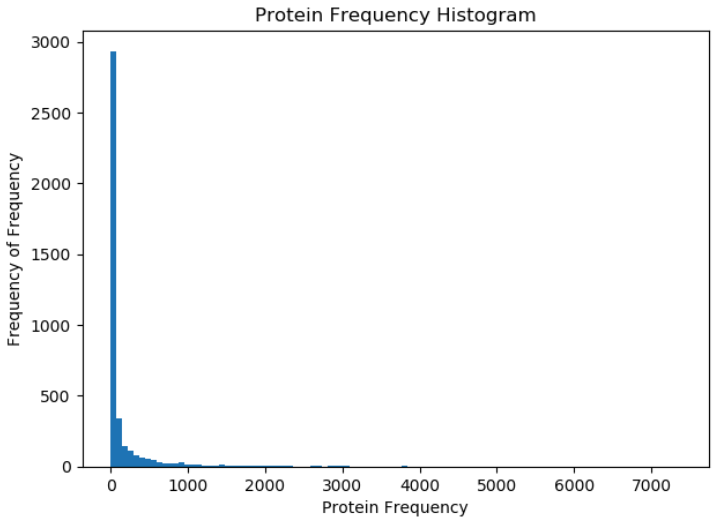
2. Exploratory Data Analysis

Before diving into the methods of my project, data was analyzed to detect any possible sources of bias and get ideas for how to understand the various results achieved in this project. This dataset consisted of more than a million drug-target interactions and was made available for download by a team of researchers. [6] The following graphs should help make sense of the vast amounts of data that were navigated throughout this project. They should reveal some unavoidable biases within data and provide information about trends of protein and compound occurrences in this dataset. Data analysis was conducted with matplotlib, and the data was stored on an sqlite3 database.

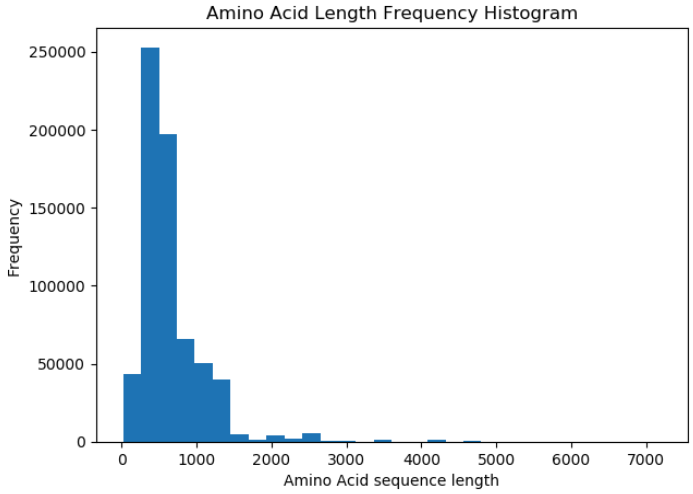


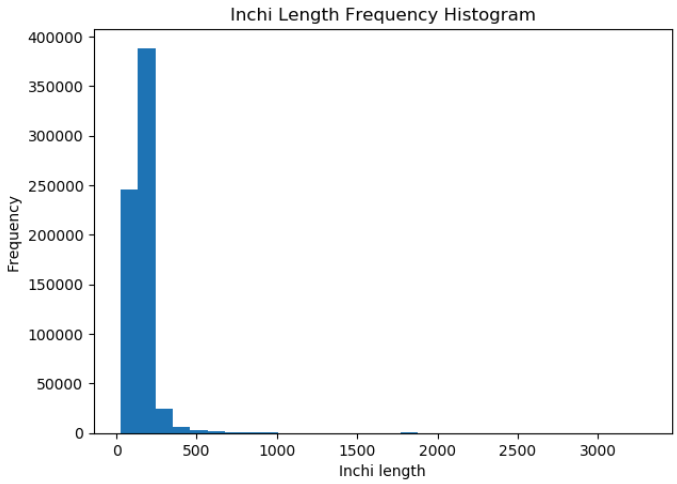
This figure reveals that IC50 values within this dataset are mostly higher, meaning no interaction (54% are greater than 1000 nM). This suggests predictions will lean towards no interaction and might not be as balanced as hoped. On the other hand, It is fairly consistent with

the world outside this experiment, as protein kinase inhibiting compounds are difficult to find. [7]

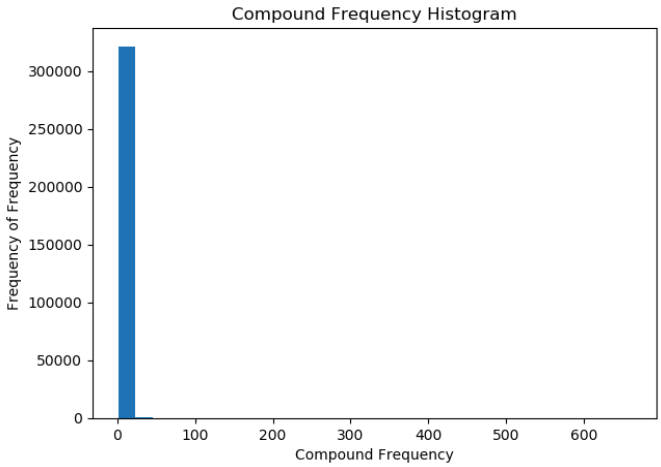


On the left, there is a clear indication that almost 3000 proteins appear only a handful of times in the dataset, which would lead to a more applicable model that has been exposed to diverse and balanced data. A caveat, however, is that this histogram is misleading. As a matter of fact, four different proteins appear in this dataset between 6000 and 7000 times. This occurrence just about balances out the thousands of proteins with only a handful of occurrences, which also could cause a form of bias to be discussed later. There are around 4000 different protein kinases represented in this dataset.





The above two figures depict a distribution of how long InChIs and amino acid sequences tend to be. It is evident that there is a great amount of variation in sequence length, which leads to hundreds of zeros being inserted into concatenated InChIs and amino acids later in this



project.

In the above graph, it is evident that nearly all of the compounds are present in this dataset between 2 and 22 times, indicating evenly distributed data. There are more than 300,000 different compounds present.

3. First Implementation

3.1. First Implementation Methods

My project went through two main implementations. The first implementation represented the drug molecules with IUPAC International Chemical Identifiers (InChIs), which

are a unique mixture of numerical and symbolic representations of molecules that a computer can generate.[8] Within the dataset, drug compounds were identified by name, ChEMBL index, and InChIKey. Therefore, it was a matter of downloading a dataset with InChIKeys and InChIs from PubChem and writing a Python script to create a new dataset of interactions with InChIs to identify the compounds. [9] It is important to note that the InChI key would not have been as useful, since it is a hashed, shortened version of the InChl which does not directly represent the molecule's structure and is not always unique. [10] In theory, the InChl is also not an ideal representation of the drug molecule's structure since it contains notation and shorthand. However, it is directly derived from the molecule's structure, so it should be meaningful as data for the neural network to process. Here is an example of InChI:

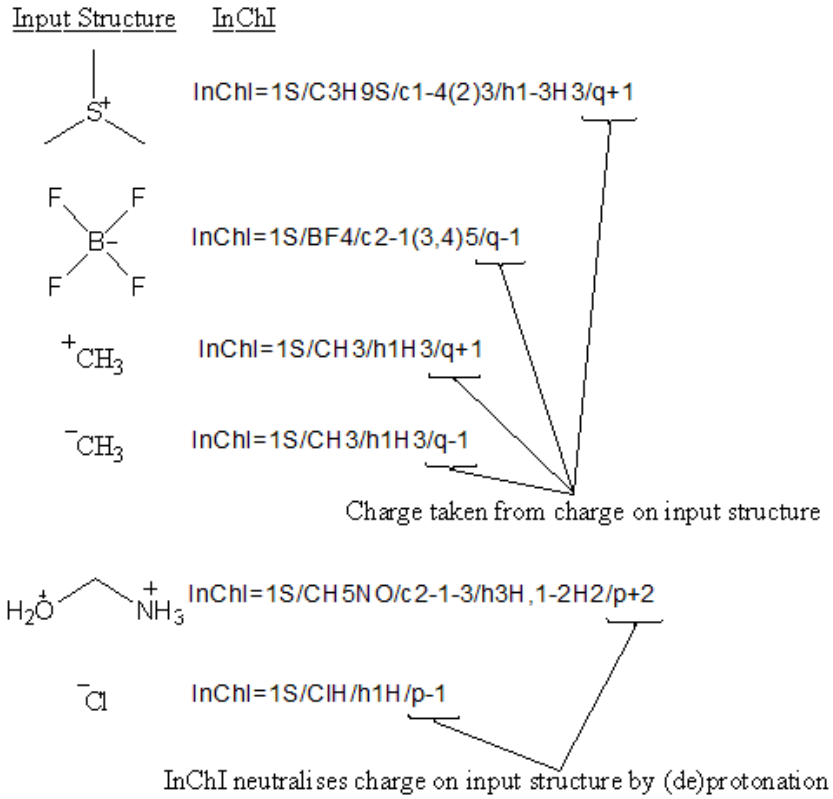


Fig 1. A visualization of how an InchI is calculated

The protein kinases were represented with their amino acid sequences. The dataset only contained protein identifiers, so I ran HTML code to get a protein kinase's amino acid sequence based on its identifier from the Uniprot database. [13] While it is in fact very difficult to

determine a protein kinase's structure, even for a computer, based solely on the amino acid sequence, the amino acid sequence does in fact directly determine a protein's structure, so it should be meaningful as data for this problem. [11] The potency of the interaction between kinase and drug was denoted by an IC50 value, which is the concentration of an inhibitor needed to reduce binding affinity between an inhibitor and a target by 50%. [12] The dataset I worked with had IC50 values ranging from 1 to 10^12 nM, with the low values indicating a high binding affinity and high values indicating the opposite. However, determining what IC50 value to consider high and what to consider low became a key area of fluctuation in this project, with different decisions yielding different results. (see results)

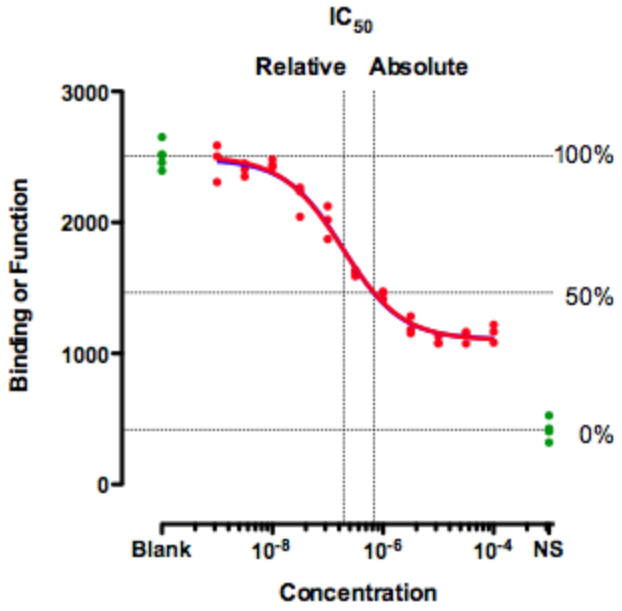


Fig 2. A visualization of the concentrations represented by IC50 values.

The InChI and amino acid sequence were concatenated, adding zeros at the end to maintain an InchIKeey length of 352 symbols and a protein sequence length of 1454 letters. They were expressed as vectors consisting of their ASCII values and these vectors were the inputs used for my machine learning model. The IC50 value was listed at 1 above a certain threshold value which was subject to change throughout the experimentation that occurred throughout this project and listed at 0 above that value. This simplified the problem into a

categorical one and also served to answer the problem statement directly, which was whether a given drug and protein bind together or not.

In terms of the machine learning model, I chose to use a convolutional neural network (CNN) partly because of the length of the InChIs and amino acid sequences. The CNN made this long input manageable since it only linked groups of values at a time from this long list of inputs. I imported Keras and TensorFlow into a virtual Python environment in order to create the model. The pooling functions and convolutional layer caused the model to be more efficient than using a dense layer immediately, which would result in 1806 x 1806 = 3 x 10^6 connections at a time for each of up to 6 x 10^6 rows of data depending on the subset of data the model performed on. This would not be manageable or efficient in any way, especially that the most powerful computing power I had access to was a MacBook Pro laptop. This convolutional approach had many possible areas for improvement, starting with my non structural representations of both the protein kinases and the drugs. The similarities between amino acids and their various properties are not addressed by arbitrary ASCII representations of their respective letters. [8] While I did not address this issue throughout this project, with increased resources, this is the first issue I plan to address in the future. Also, the InChI identifier is only one of many possible compound structure representations, so I use a different representation called a Morgan fingerprint in my second approach to this project to compare results. Furthermore, in concatenating the two sequences, the convolutional neural network would inevitably analyze a segment with values from both the protein sequence and InchI, which are not related to each other. With these possible issues in mind, the convolutional neural network was implemented, knowing that neural networks have a reputation for detecting patterns people fail to recognize. [14]

I experimented with the structure and variables of the neural network, specifically the number of convolutional layers, size of the dataset, number of kernels, dropout, and IC50 threshold values.

3.2. First Implementation Results

As a quick note, these results often use the term accuracy. This refers to the number of predictions the neural network got right over the total number of predictions it made.

Over the course of this first implementation, a few criteria remained consistent throughout all trials and changes of the neural networks. A single neural network was used (unlike in my second implementation, where a parallel neural network was used). The test set was randomly selected as 20% of the entire dataset. The inputs consisted of InchIs and amino acid sequences.

First, analysis was conducted on the entire dataset of 600,000 drug target interactions. A model with 2 convolutional layers, a batch size of 128, and 1 epoch achieved an accuracy of 0.7938. As a side note, running this test took hours given limited computing power, so epochs were limited to 1 and the dataset size was reduced to 60,000 for further analysis. One epoch appears to be too few, but experimentation revealed that increasing the number of epochs did not affect the accuracy of results tangibly, at least not enough to justify the extra time of running the model multiple times.

Total dataset of 60,000 interactions was used throughout. The test set was a randomly selected 20% of the entire dataset. Batch size was 128 rows, and the model went through 3 epochs.

Variables Affected	Accuracy
Control Run	**0.8157**
1 convolutional layer	0.8157
2 convolutional layers	0.8190
8 kernels	0.8177
32 kernels	0.8157
Dropout .4	0.8183
Dropout .2	0.8157
IC50 Cutoff 50	0.8533

IC50 Cutoff 200	0.8157
IC50 cutoff = 800, 2 kernels, 2 dense layer nodes, 607 total parameters	0.7216

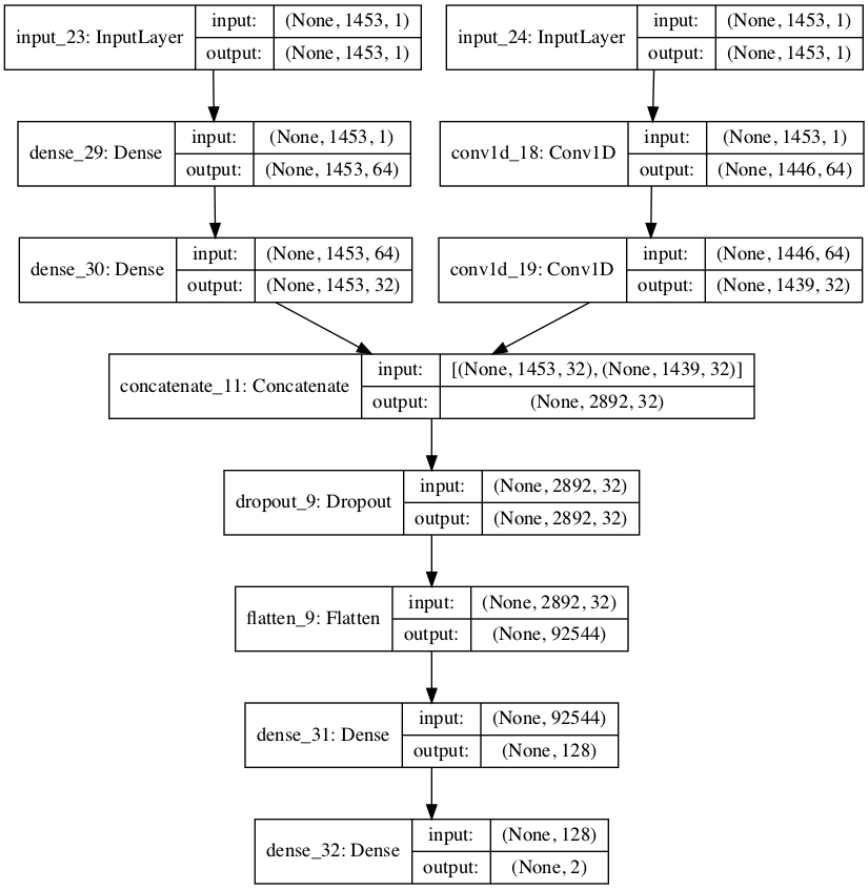
*Control run used 1 convolutional layer, 32 kernels, 0.2 dropout, and IC50 cutoff of 200

4. Second Implementation

4.1 Second Implementation Methods

The next approach to this problem had a few significant changes. Instead of InChIs, I used fingerprints, specifically Morgan fingerprints. This is the most popular molecular fingerprint, and it describes the structure of the molecule numerically, which is great as an input to a convolutional neural network. [15] For example, a numerical representation of the Morgan fingerprint for Aspirin is [389, 456, 650, 695, 807, 909, 1017, 1035, 1047, 1057, 1088, 1199, 1380, 1410, 1447, 1468, 1616, 1729, 1750, 1775, 1873, 1917, 1970, 1991]. Each of these numbers is a non-zero bit location out of 2048 bits. This numerical representation is more memory efficient since it would be impractical to have a 2048 bit vector as an input. The fingerprint is generated by labeling each atom with an identifier, updating each identifier based on its neighbors, removing duplicates, and finally generating a 2048 bit vector. Through the use of RDKit, it was possible to use an InChI to produce a Morgan fingerprint. [21] In this implementation, the protein kinases were again represented by their amino acid sequences, as I was unable to discover an algorithm such as the Morgan fingerprint tool for representing protein structure.

Another key difference between this implementation of the neural network and the previous one is the use of parallel structure in this instance.



As can be seen in this figure, through a parallel neural network, the two inputs (proteins and drugs) are analyzed separately, so the problem of overlap with the concatenation does not occur. Instead the two inputs undergo separate layers of analysis before being concatenated. The amino acid sequences underwent two convolutional layers of analysis, and the fingerprint underwent two dense layers of analysis. This was possible due to the fingerprints being shorter in length than the amino acid sequences, making it manageable memory and parameter wise. After concatenation, two more dense layers of analysis occurred before yielding a categorical prediction of either 1 or 0. As a side note, a dropout layer was included of 30% to prevent overfitting of the model, and a dataset of 60,000 interactions was used throughout this implementation.

4.2. Second implementation Results

4.2.1 Second main implementation, 1st experiment

The purpose of this first experiment was to maintain a consistent IC50 cutoff of 100, keeping all variables constant and only changing from single to parallel neural networks in order to determine if parallel neural networks result in higher accuracy for this specific project.

The interactions between drugs and three different commonly occurring proteins were used as test sets to crystallize the different performance of the neural networks given specific proteins with thousands of interactions with different drugs.

Protein 1 (Epidermal Growth factor receptor) 6857 interactions.
Protein 2 (Vascular endothelial growth factor receptor 2) had 6856 interactions.
Protein 3 (Potassium voltage-gated channel subfamily H member 2) had 6574 interactions.
*I will refer to these proteins by number in my report for clarity and convenience.

Test set:	Single Neural Network	Parallel Neural Network
Interactions with protein #1	67% accuracy	67% accuracy
Interactions with protein #2	88% accuracy	93% accuracy
Interactions with protein #3	58% accuracy	58% accuracy
Interactions with all 3 proteins	73% accuracy	73% accuracy

At this point, the criteria changed here, as all of the models were parallel neural networks, and the test set consisted of drug interactions with Protein #2. The purpose was to determine the significance of IC50 values on accuracy of predictions.

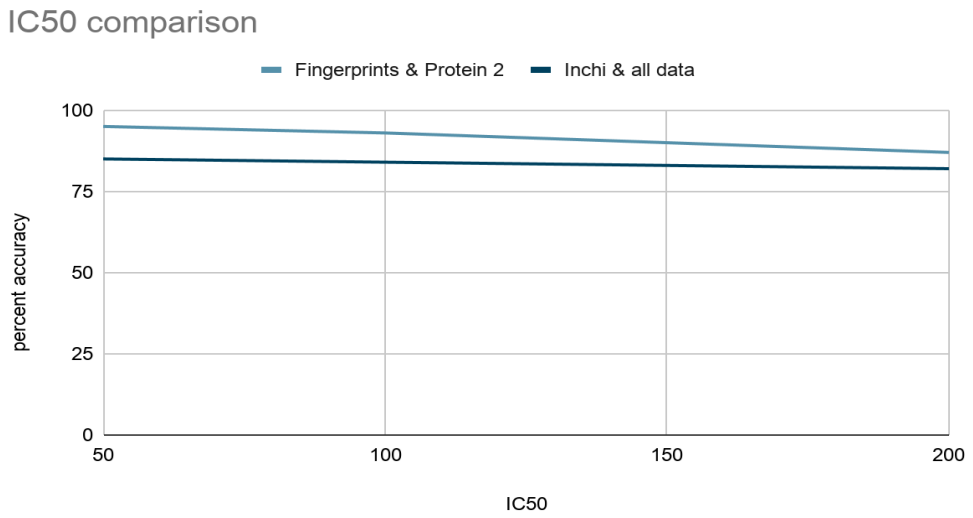
IC50 Cutoff	Accuracy of Prediction
50	95%
100	93%
200	87%

5. Discussion

Looking at the 2019 results, we can see that the most significant factor affecting accuracy of the CNN (convolutional neural network) was changing the IC50 value threshold. A change from a cutoff of 50 to 200 resulted in a 4% decrease in accuracy. By looking at the IC50 value bar graph, the vast discrepancy between the number of interactions with IC50 values below 50 and above 50 suggests this change in accuracy could reflect some bias in the data such as the model always favoring picking a “0” prediction when the IC50 threshold is set to 50. However, this increase in accuracy also likely reflects a low IC50 being indicative of a strong interaction between a given drug and protein and therefore the model can pick up on more definitive patterns to base its predictions on. The rest of the results indicated that dropout and number of kernels had no discernible effect on accuracy of predictions and increasing model layers from 1 to 2 led to a .33% increase in accuracy. The further effects of increasing layers could not be tested due to limited computing power, unfortunately. Furthermore, the last test with the IC50 threshold raised to 800 and overall parameters reduced to a very low 607, accuracy remained at 72% which further raises the question of how the CNN managed to remain accurate despite dramatic reductions in computations done. Analysis on the data yielded no significant discrepancies in data as different drug compounds and protein compounds were distributed evenly throughout the dataset with the exception of 4 protein kinases which appeared in at least 6000 interactions each out of 60,000 total interactions. However, upon further analysis, the ratio of inhibition to no inhibition was around 70-80% depending on what IC50 threshold was chosen

for measuring. Therefore, the source of bias must be more subtle and could be similar in nature to the infamous example of a neural network recognizing wolves and dogs based on the snow in the background.

In the next iteration of the project, the CNN was tested on interactions with specific proteins. In this way it could be guaranteed that the test set remained exactly the same each time and the effects of changing IC50 value and using parallel neural networks could be convincingly determined. On protein #2, as the IC50 threshold changed from 50 to 200, accuracy decreased by 8%, which was twice as great as the 4% change in accuracy in the first iteration of the project with InchI inputs and a generalized test set. This could be attributed to a mixture of the effectiveness of InchI inputs vs Morgan fingerprint inputs, a more diverse test set diluting results, or an effect of parallel neural networks vs a single neural network.



However, it could be conclusively determined that a parallel neural network structure was beneficial to this specific problem since it resulted in a 5% increase in accuracy of predictions of interactions for protein #2. Furthermore, both stages of the project confirmed that picking an IC50 value threshold has a significant impact on accuracy of the model and thus on how the model is trained, with accuracy ranging from 87% to 95% from only changing IC50 cutoff. It will be necessary to ascertain a proper threshold value that would be accepted as indicative of a compound consistently inhibiting a protein kinase. There is much area for flexibility in acceptable IC50 thresholds depending on the situation, but at least there appears to be consensus

that an IC50 of 10^6 nM is far too large.[16] Hopefully, in the future I can continue this project at college with access to stronger computational power, which would allow me to create a neural network with more layers and parameters. I also aim to approach the IC50 cutoff issue differently, making predictions scaled instead of categorical and using a different method such as graphing predictions to calculate accuracy, since they would no longer simply be right or wrong. Furthermore, I must do even more research on representations of molecular and protein structures as my current use of Morgan fingerprints and amino acids may not be ideal representations, especially the amino acid sequences.

7. References

[1] 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.

[2] Gagic, Z et al. In silico Methods for Design of Kinase Inhibitors as Anticancer Drugs. Frontiers in Chemistry 2020.

[3] Cichonska, A et al. Computational-experimental approach to drug-target interaction mapping: A case study on kinase inhibitors. PLOS Computational Biology 2017.

[4] Kannaiyan R, Mahadevan D. A comprehensive review of protein kinase inhibitors for cancer therapy. Expert Rev Anticancer Ther. 2018;18(12):1249-1270. doi:10.1080/14737140.2018.1527688

[5] Alejandra, and Sean. "Molecular Docking: Bioinformatics in Drug Discovery: TSC." *The Skeptical Chemist*, 28 May 2020, theskepticalchemist.com/molecular-docking-bioinformatics-drug-discovery/.

[6] Tang, Jing, et al. "Drug Target Commons." Cell Chemical Biology, 2018, https://drugtargetcommons.fimm.fi/

[7] Smyth LA, Collins I. Measuring and interpreting the selectivity of protein kinase inhibitors. J Chem Biol. 2009 Aug;2(3):131-51. doi: 10.1007/s12154-009-0023-9. Epub 2009 Jun 6. PMID: 19568781; PMCID: PMC2725273.

[8] Heller, Stephen R, et al. "InchI, the IUPAC International Chemical Identifier." Journal of Cheminformatics, Springer International Publishing, 30 May 2015.

[9] Kim S, Chen J, Cheng T, Gindulyte A, He J, He S, Li Q, Shoemaker BA, Thiessen PA, Yu B, Zaslavsky L, Zhang J, Bolton EE. PubChem 2019 update: improved access to chemical data. Nucleic Acids Res. 2019 Jan 8;47(D1):D1102-D1109.

[10] Willighagen, Egon. InChIKey Collision: the DIY Copy/Pastables, 17 Sept. 2011, chem-bla-ics.blogspot.com/2011/09/inchikey-collision-diy-copypastables.html.

[11] Deng H, Jia Y, Zhang Y. Protein structure prediction. Int J Mod Phys B. 2018 Jul 20;32(18):1840009. doi: 10.1142/S021797921840009X. Epub 2017 Dec 11. PMID: 30853739; PMCID: PMC6407873.

[12] Aykul S, Martinez-Hackert E. Determination of half-maximal inhibitory concentration using biosensor-based protein interaction analysis. Anal Biochem. 2016 Sep 1;508:97-103. doi: 10.1016/j.ab.2016.06.025. Epub 2016 Jun 27. PMID: 27365221; PMCID: PMC4955526.

[13] "Uniprot." Uniprot, National Institutes of Health, 2002, https://www.uniprot.org/

[14] Jennifer Chu | MIT News Office. "Deep-Learning Technique Reveals 'Invisible' Objects in the Dark." MIT News | Massachusetts Institute of Technology, news.mit.edu/2018/deep-learning-dark-objects-1212.

[15]Fingerprints in the RDKit. www.rdkit.org/UGM/2012/Landrum_RDKit_UGM.Fingerprints.Final.pptx.pdf.

[16] Lavogina, Darja. (2017). Re: Is there any threshold (IC50) between strong and weak small molecule inhibitors? When small molecule with moderate inhibition become a better option?. Retrieved from: https://www.researchgate.net/post/Is-there-any-threshold-IC50-between-strong-and-weak-small-molecule-inhibitors-When-small-molecule-with-moderate-inhibition-become-a-better-option/5a3fb62edc332d42202fcda0/citation/download.

[17] "IDG-DREAM Drug-Kinase Binding Prediction Challenge." Synapse, 2020 Sage Bionetworks, https://www.synapse.org/#!Synapse:syn15667962/wiki/

[18] Cicenas, Jonas, et al. "Kinases and Cancer." Cancers, MDPI, 1 Mar. 2018, www.ncbi.nlm.nih.gov/pmc/articles/PMC5876638/.

[19] Lakhani, Paras, et al. "Hello World Deep Learning in Medical Imaging." Journal of Digital Imaging, Springer International Publishing, June 2018, www.ncbi.nlm.nih.gov/pmc/articles/PMC5959832/.

[20] Brownlee, Jason. "Handwritten Digit Recognition using Convolutional Neural Networks in Python and Keras." Machine Learning Mastery, Machine Learning Mastery Pty. Ltd., June 2016, https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/

[21] Landrum, Greg. "Getting Started with the Rdkit in Python." Open Source Cheminformatics and Machine Learning, 2007, https://rdkit.readthedocs.io/en/latest/GettingStartedInPython.html#

Fig 1. “Technical FAQ.” *InChI Trust*, 30 Nov. 2018, www.InchI-trust.org/technical-faq-2/.

Fig 2. “50% Of What? How Exactly Are IC50 and EC50 Defined?” *GraphPad*, www.graphpad.com/support/faq/50-of-what-how-exactly-are-ic50-and-ec50-defined/.