**END-SEMESTER REPORT**

**ON**

**MACHINE LEARNING BASED PHASE I AND II DRUG METABOLISM PREDICTION**



BY

|  |  |  |
| --- | --- | --- |
| Srijan Verma | 2016A5TS0659P | BACHELOR OF PHARMACY |

Under the supervision of

Dr. Suman Sirimulla, and Dr. Vaibhav A. Dixit

Submitted in partial fulfillment of the requirements of

First Degree Thesis (BITS F421T)

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI**

December, 2019

**Acknowledgements**

I wish to express my sincere gratitude to Dr. Suman Sirimulla and Dr. Vaibhav A. Dixit for providing me the opportunity to work on this project and for giving their invaluable insights throughout the project period.

**Table of contents**

Acknowledgements………………..…………………………………………..... 1

Abstract.……………......……...……………………………………………........ 3

1. Introduction……………..……………………...……..………..………..…… 4

2. Data collection……..……..……..……..…………………………………….. 7

2.1 Knime……....…………….……….……………………….………… 7

2.2 Data collection with KNIME…………………….………………….. 8

2.3 Using python……....…………….……….…………………………. 8

2.4 SuperCyp Literature………….….……….…………………………. 8

3. Data preprocessing…………...……………………………………..……….. 9

3.1 Standard Activity comment……....…………….……….…………… 9

3.2 Activity filter………………………………..……………….……… 10

3.3 Labelling…..…………………………………..………….…….……. 10

4. Molecular Features…………………….......…………….............................. 10

5. Model development………...............………………………..………………. 11

6. Evaluation Metrics………...............………………………..………………. 12

7. Results……………..…...………………..………..………..……………….. 13

8. Improved Results with SULT regression models.…......……..……………… 14

9. References………..………………………………………….……………… 15

10. Supporting Information……..………..…………………….……………… 22

**ABSTRACT**

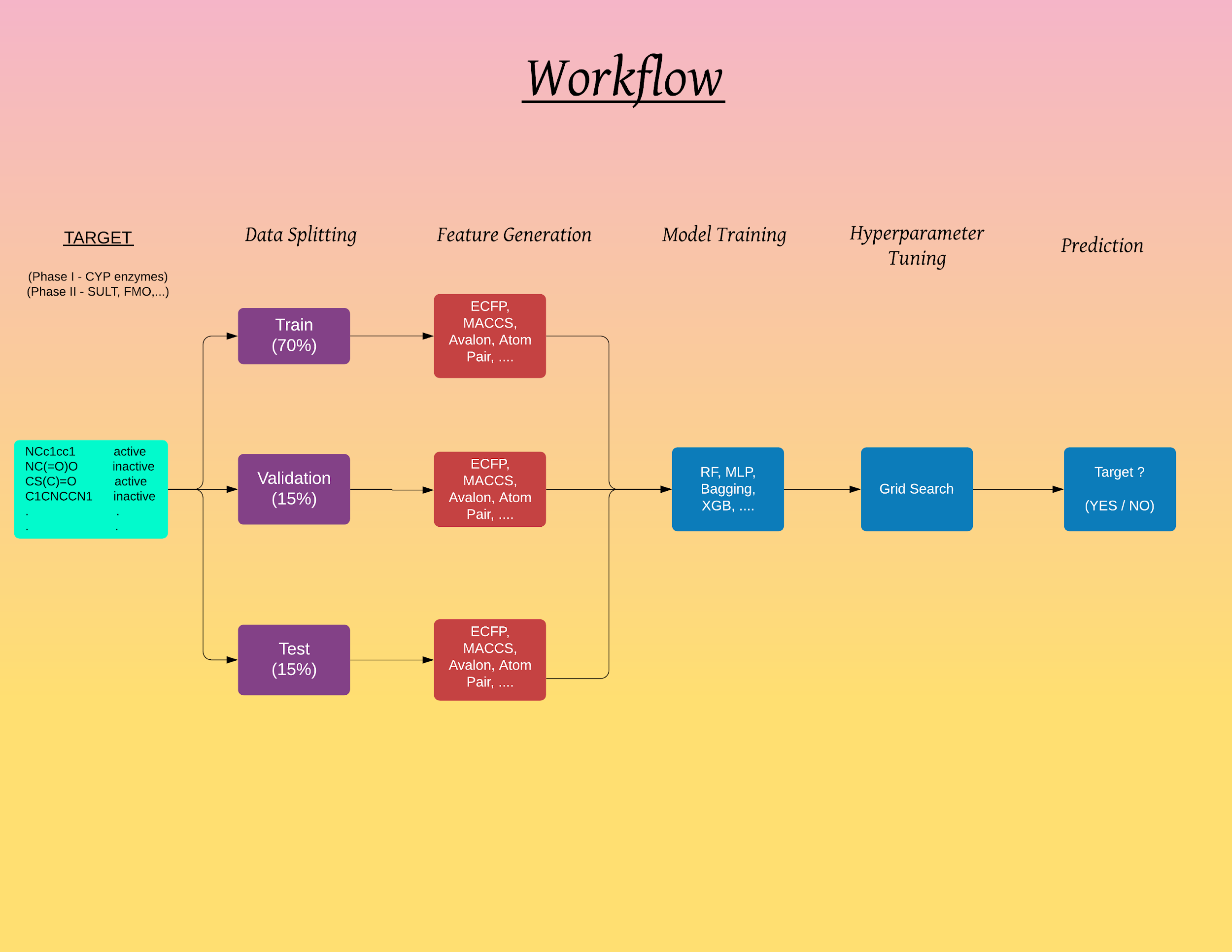
Discovering a drug involves multiple disciplines and interests. Usually, a large number of compounds are evaluated, during the discovery phase, for pharmacological and toxicological activities. Various types of problems with respect to pharmacodynamics, pharmacokinetics, and toxicity are commonly encountered during lead optimization and candidate selection. Majority of drugs are non-polar by design due to the requirement of reaching the site of action and effective interactions with its target(s). About 80% of drugs are metabolized by the liver in two phases which generally converts them into inactive compounds (metabolites). Phase I is mostly oxidation, while phase II involves the incorporation of additional polar groups to enhance water solubility and excretion.

Drug metabolism often has a decisive influence on drug toxicity. Thus, drug metabolism studies are performed early-on and play an important role in the optimal selection of compounds for further development, provide information on metabolites for possible improvement in drug design, and contribute to the identification of the appropriate animal species for subsequent toxicity testing. Thus, early assessment of metabolic pathways with robust, and accurate drug metabolism prediction tools are of utmost importance.   
Robust Phase I metabolism prediction models are available (for CYP450 enzymes) but phase II processes have been ignored so far. Our objective here is to develop and integrate Comprehensive and Holistic predictive models for both, phase I and II drug metabolism.

The applications of artificial intelligence (AI) in pharmaceutical research has emerged in recent years, and its utility has not only gone beyond bioactivity predictions but has also shown promising results in diverse problems of drug toxicity.

Here, we will use the techniques of machine learning and deep learning, to develop substrate/inhibitory classification and Site of Metabolism (SOMs) models for CYP450 and UGT isoforms.

## 1 Introduction

Drug discovery and development suffer from high failure rates. Drug metabolism mediated toxicity contributes significantly (25%) in these failures. Drug metabolism, thus, is a major area to study for therapeutics and toxicity. As such, drug metabolism is of great importance in medicinal chemistry and clinical pharmacology because it influences the deactivation, activation, detoxification, and toxification of most drugs. There is a huge impact of drug metabolites on the clinical trials, which results in their failures. Which is why there is an even more urgency for developing good predictive models.

A recent meta-analysis[[1]](#footnote-0) confirms the major role of reactions which are catalysed by cytochrome enzymes, but it also demonstrates the role of reactions catalysed by non-CYP450 enzymes (for example: oxidoreductases, hydrolases, transferases). 60% of first-generation metabolites are produced by CYPs, but the contribution of this superfamily strongly decreases in the second, third and higher generations. In the Phase II metabolic processes, transferases (GST, MET, NAT), conjugating enzymes (UGT, SULT) and some non-CYP oxidoreductases play major roles depending on the structure and polarity of the compounds and Phase I metabolites. A lot of information is available on the natural and designed inhibitors[[2]](#footnote-1) for these non-CYP450 enzymes, but limited work has been done on understanding/predicting substrates and sites of metabolism especially in molecules with multiple potential sites of metabolism.

In the last decade site of metabolism (SOM) prediction tools like Xenosite[[3]](#footnote-2), RS-predictor and SMARTCyp[[4]](#footnote-3) have been developed. These methods give high quality predictions for major CYP450 isoforms and are being used widely by the drug discovery community.

Nonetheless, there are following limitations with these tools:

1. These SOM models don’t check for the substrate or inhibitor potential for the input compound. (For e.g. all these methods will predict SOM for metformin as N-Me groups, but metformin is not a substrate of any of the CYP450 isoforms).
2. These models only annotate the SOM and do not generate structures for expected products.
3. These Phase I and Phase II prediction models have not been integrated to give sequential, comprehensive and holistic predictions.
4. Moreover, only primary metabolites are predicted and the probability of secondary metabolism is altogether ignored. (Example: diamorphine, codeine, enalapril, and levodopa)
5. Additionally, these models only make qualitative predictions and kinetics (Km, Vmax, IC50) of the metabolic processes are ignored.

Here, we will develop novel machine learning based phase I and II integrated models for comprehensive drug metabolism prediction. That is to,

* Develop CYP450 substrate and inhibitor models
* Develop CYP450 site of metabolism (SOM) models
* Develop substrate and inhibitor models for phase II enzymes
* Integrate phase I and phase II models into a comprehensive and holistic model for prediction of a complete drug metabolite profile(s).

KNIME, python and machine learning techniques were employed for model development and deployment tasks.

KNIME is an open source chemoinformatics platform for tasks like data mining, data manipulation, machine learning and data integration. It is built on top of Java, and also supports Python integration. Data was collected, formatted, labelled and analyzed from the world’s largest chemical database, ChEMBL. Input features in the form of fingerprints/descriptors were generated. That is, data was transformed in various ways to benchmark a selection of machine learning techniques.

## 2 Data collection

### 2.1 Introduction to KNIME

KNIME is an open source chemoinformatics platform for tasks like data mining, data manipulation, machine learning and data integration. It is built on top of Java, and also supports Python integration. A graphical user interface allows users to manipulate data in different formats, including preprocessing for modeling, data analysis and visualization without, or with only minimal, programming.

KNIME is extensively been used for chemoinformatics or for dealing with medicinal data. It was first created in 2006. It allows users to have an interactive view for analysis and create workflows using a drag and drop functionality. It serves various use cases such as extracting data from a relational database such as SQL, SQLite. It can read many types of files ranging from pdf, doc, docx, excel, xml, ppt. Users can write and create their own self custom nodes, if required. Due to its main language being Java, it is computationally fast as well. For beginners, KNIME provides many free workflows which can be used as it is, for learning.

### 2.2 Data collection with KNIME

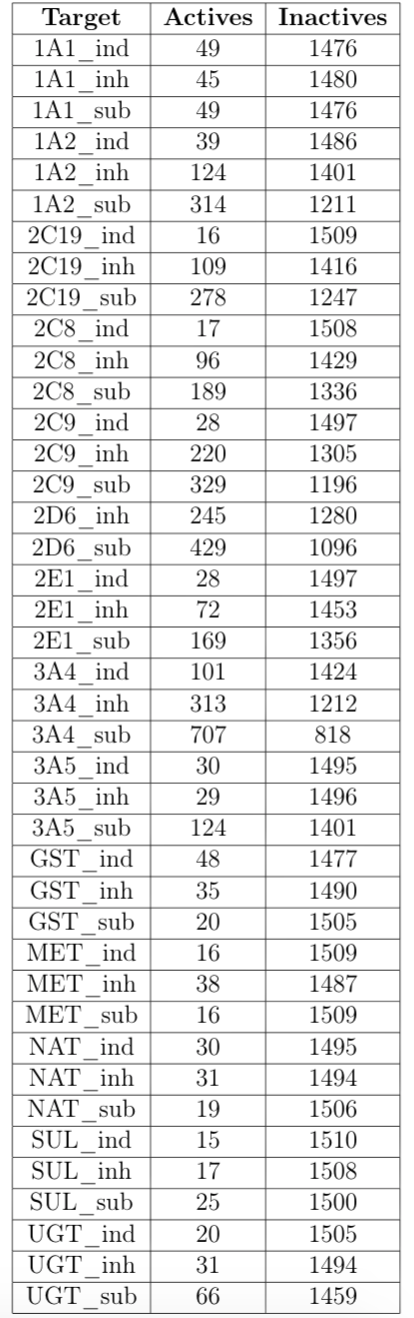
For CYP450 enzymes, data was collected using KNIME’s DB reader node. This specific node provides easy to access data interface for a structured database, like that of MySQL. Firstly, ChEMBL25 was loaded and populated in the server. CHEMBL mySQL database has a total of 77 tables containing data for bioactivity molecules and drug-like compounds (SI). SQL queries were used to extract specific data. ChEMBL’s SMILES data was stored in compound\_structures table and was mapped with molregno. Inner join of 6 tables, namely- assays, compound\_structures, target\_type, activities, target\_dictionary and molecule dictionary, was taken for getting the required data.

### 2.3 Python

Data for 47 human CYP450 enzymes was available in ChemBL25. Python was used for collecting the data for all CYP450 enzymes.

Twenty out of the 47 CYP450 enzymes had less than 28 data points, which were thus removed.

Table 1: Summary of dataset showing number of actives in each of the Phase I and II enzymes. Data for CYP450 3A4 substrate has the most number of actives in it.



### 2.4 SuperCyp and Literature

Substrate and Inducer data was collected from SuperCyp[[5]](#footnote-4) and two other

literature resources[[6]](#footnote-5),[[7]](#footnote-6). Python library Selenium[[8]](#footnote-7), BeautifulSoup[[9]](#footnote-8) and requests

were used for this task.   
 Example web-scraping code given below:-

|  |
| --- |
| **for** i **in** tqdm\_notebook(range(len(q\_name))):    url = 'http://bioinformatics.charite.de/transformer/index.php?site=fullinfo\_results&cname='+ q\_name[i] +'&cyp'  result = requests.get(url)  c = result.content  soup = BeautifulSoup(c)  soup\_string = str(soup)    all\_d\_name = []  all\_d\_type = []  # q\_str = 'Drug Interactions'  dyn\_str = 'Drug Interactions' #Dynamic  s1 = 'href="index.php?site=fullinfo&amp;cname=' #Constant  s2 = '</a>' #Constant  s3 = '<td>'  s4 = '</td>'  drug\_name = ''    **while**(soup\_string.find(s1, soup\_string.find(dyn\_str)) != -1):  drug\_name = soup\_string[soup\_string.find(s1, soup\_string.find(dyn\_str)) + len(s1):soup\_string.find(s2, (soup\_string.find(s1, soup\_string.find(dyn\_str)) + len(s1)))]  all\_d\_name.append(drug\_name.split('>')[1])  dyn\_str2 = drug\_name  drug\_type = soup\_string[soup\_string.find(s3, soup\_string.find(dyn\_str2)) + len(s3):soup\_string.find(s4, (soup\_string.find(s3, soup\_string.find(dyn\_str2)) + len(s3)))]  dyn\_str = drug\_name  all\_d\_type.append(drug\_type)    df = pd.DataFrame(columns = ['drug\_name','drug\_type'])  df['drug\_name'] = all\_d\_name  df['drug\_type'] = all\_d\_type  df.to\_csv('../dataset/transformer/' + prot\_name[i] + '.csv') |

A web crawler and a scraper was written in python, for this task. Step-wise

procedure given below:

* All the data was first scraped from Transformer[[10]](#footnote-9).
* If SMILES pattern was not found in transformer, for that, the data was extracted from PubChem/CHEMBL/DrugBank
* Additional data was collected from literature[[11]](#footnote-10) which contained 2D6, 2D9 and 3A4 enzyme data. Intersection of this with the data collected from Transformer was taken afterwards. Rest was scraped from PubChem.
* Data for 7 enzymes (1A2, 2C19, 2C8, 2C9, 2D6, 2E1, 3A4) was directly downloaded from literature source and was merged with the data collected above

## 3 Data preprocessing

### 3.1 Standard Activity comment

The ChEMBL table activities includes an activity\_comment field. If the activity\_comment was either "Active" or "active", the measurement was considered as active; if the comment was either "inactive", "Not Active" or "Not Active (inhibition < 50% 10 uM and thus dose-response curve not measured)" the measurement was considered to be inactive. In these cases, no further details about the measurements were considered.

### 3.2 Activity filter

Of the measurements that had no standard activity comments as described in

3.1,

those measurements were discarded for which either standard\_value was empty, standard\_units was unequal to "nM" or standard\_relation was not associated with any of the standard relationship annotations {">", ">=", "<", "<=", "=", "~"}.

### 3.3 Labelling

Each of the data points were labelled in any one of the following: active,

inactive,

Indeterminate.

* Standard\_values having nM >= 31,622.2 or -log10[M] <= 4.5, were labelled as inactive.
* Standard\_values having nM <= 3,162.2 or -log10[M] >= 5.5, were labelled as active.
* Standard\_values whose -log10[M] > 4.5 and -log10[M] < 5.5 were labelled as indeterminate.

Format of labelled data is given below: This is an example for a dataset   
 having 1A1 inhibitors labelled as *positive* and rest labelled as *negative.* Similar data was

generated for other Phase I and II enzymes.

## 

## 4 Molecular Features

Total of 19 features of four different kinds : Circular, Path-based, Substructure keys[[12]](#footnote-11),[[13]](#footnote-12) and RDKit descriptors were generated for each of the enzymes. Fingerprints were converted to a bit-vector of either 1024 or 16,384 length, whereas RDKit had 200 different features respectively.   
**Circular fingerprints:**

These include the extended-connectivity fingerprints (ECFPx) and

feature-connectivity fingerprints (FCFPx), where x is 0, 2, 4, and 6 are the bond length or diameter for each circular atom environment. ECFP consists of the element, number of heavy atoms, isotope, number of hydrogens, and ring information; whereas FCFP consists of pharmacophore features.

**Substructure keys:**

Avalon and the public Molecular ACCess System (MACCS) are two

different types of fingerprints that are substructure keys. The Avalon fingerprint, used here, is a bit vector of size 1,024. It includes feature classes such as atom count, atom symbol path, augmented atom and augmented symbol path. MACCS structural keys are 166-bit structural key descriptors. Each bit here is associated with a SMART pattern and

belongs to the dictionary-based fingerprint class.

**Path-based fingerprints:**

This includes RDKx (where x is 5, 6, 7), topological torsion

(TT), HashTT, atom pair (AP), and HashAP. The size of fingerprints are 1024 for all of them. (SI point 2)

**Long Fingerprints:**

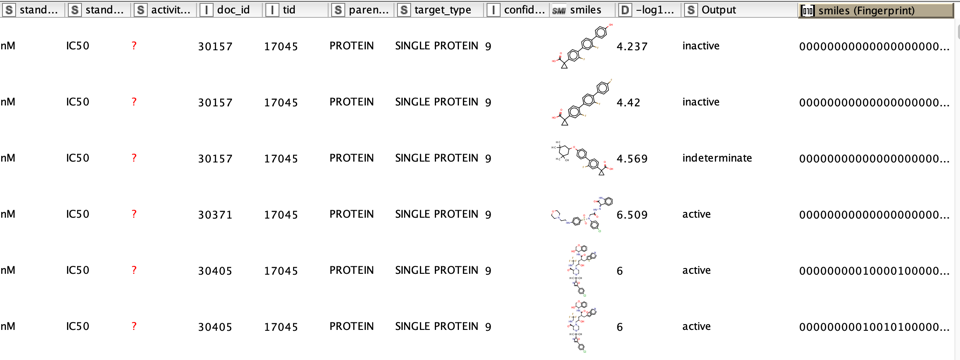
A longer version of fingerprint, of 16384 bits, was also used for comparison. This longer version is represented by the prefix "L": LAvalon, LECFP6, LECFP4, LFCFP6 and LFCFP4.  
**RDKit Descriptors:**

A set of 200 descriptors (SI point 3) were used which were obtained from RDKit[[14]](#footnote-13).

They are either experimental properties or theoretical descriptors, which are molar refractivity, logP, heavy atom counts, bond counts, molecular weight, topological polar surface area and so on.

**Combination Features:**

In this, we combined RDKit Descriptors with LECFP6, LAvalon and HashAP.

Fingerprint format given below

## 5 Model development

Models were trained and developed on TACC stampede2 clusters (more info on SI point 4). Sklearn machine learning library, having 28 different classifier (SI point 5, end of doc) models, were used. 70% of the data was used for training, 15% for validating and 15% for testing. Models were set to their default settings. 7,448 models were trained in total.

Since the data was binary, classification models having 2 classes (active, inactive) were used.

Given below image shows the classification of ML algorithms

### 

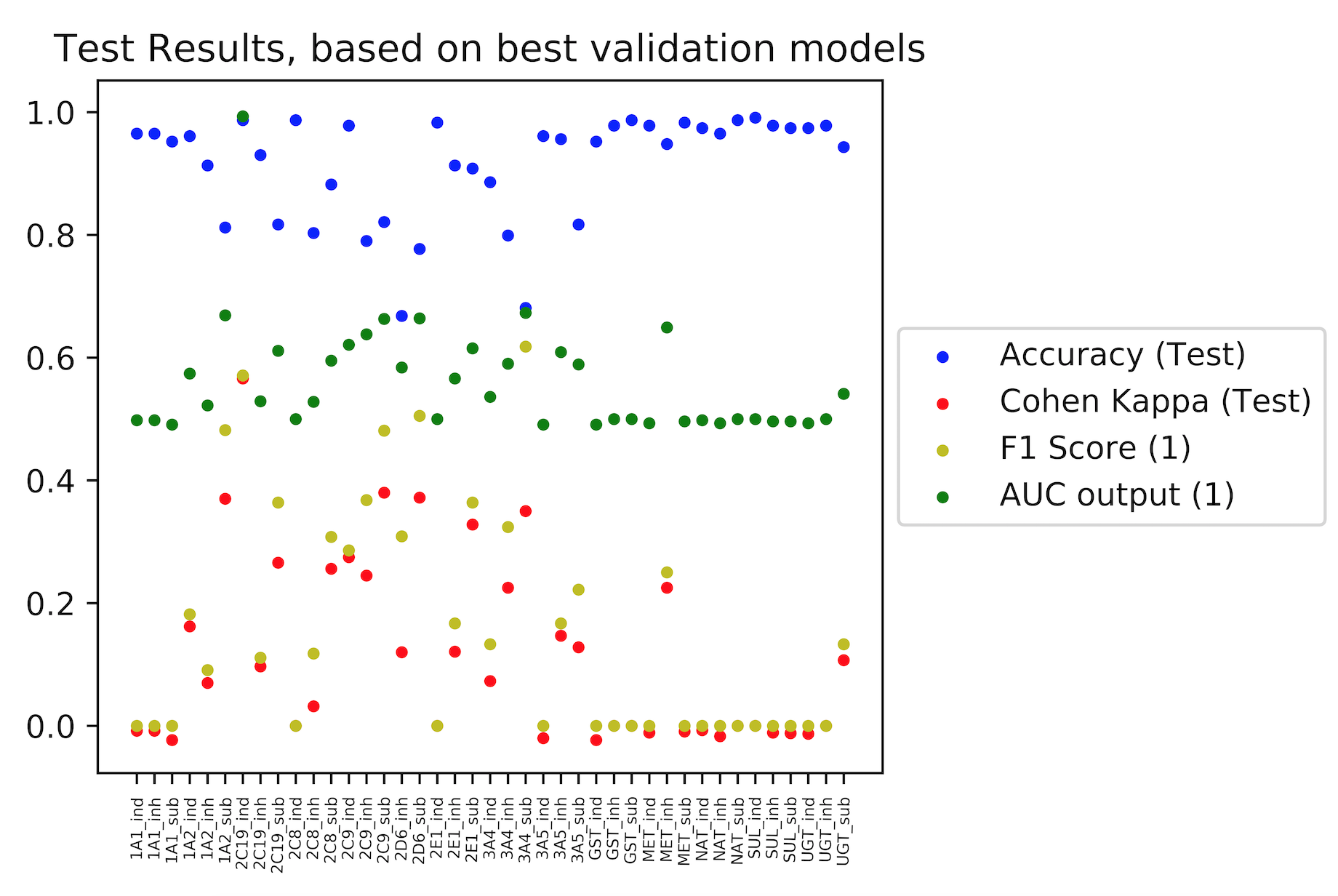
## 6 Evaluation metrics

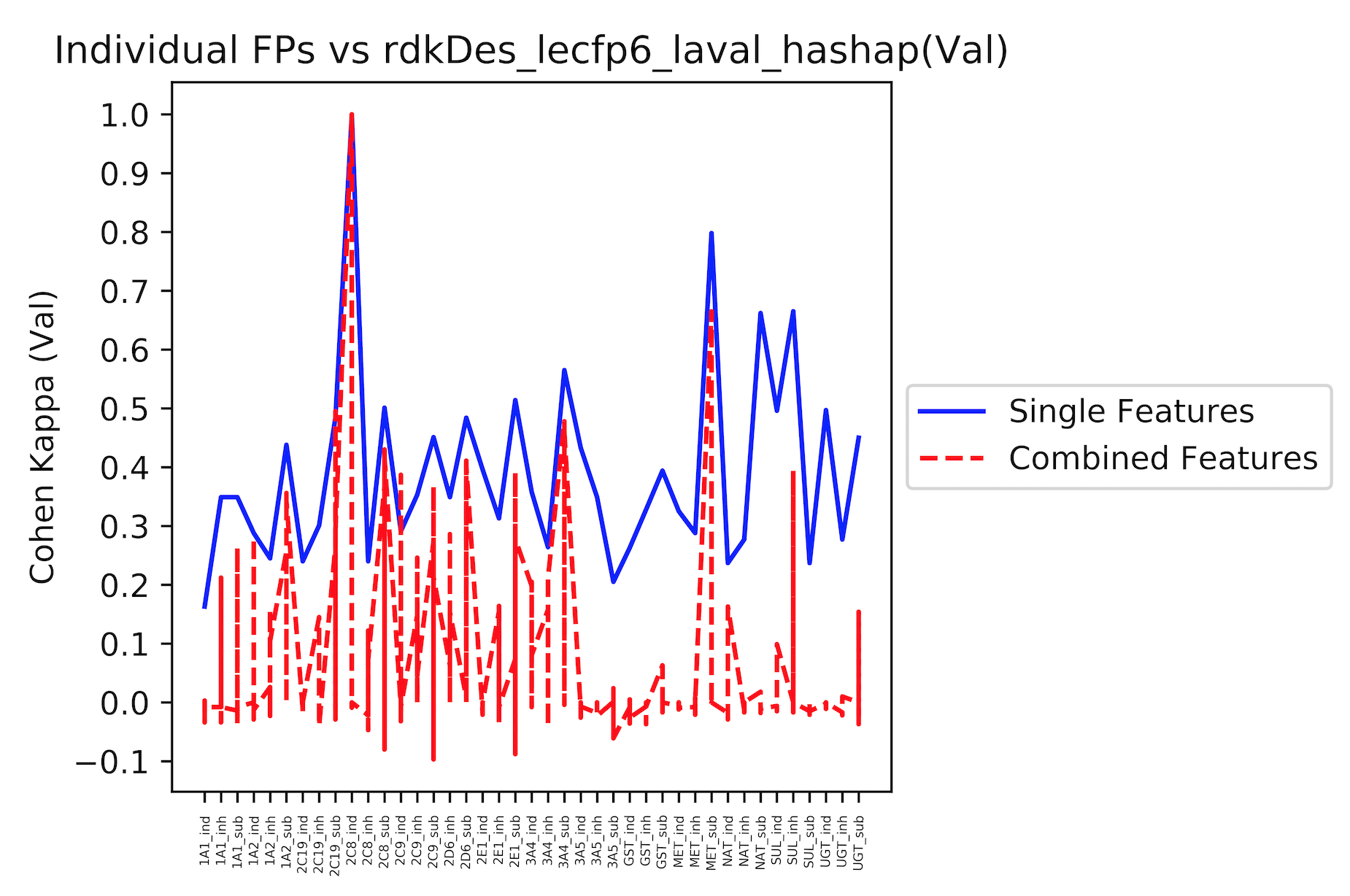
Standard metrics (AUC-ROC, F1….) were used for measuring the performance of all the models.

* AUC[[15]](#footnote-14): It is a performance measurement for classification problem at various thresholds settings. Degree of measurement is evaluated by AUC and probability curve is measured by ROC. It tells how much model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting the classes.  
  In multi-class model, we can plot N number of AUC ROC Curves for N number classes using One vs ALL methodology. For example, If we have three classes named K, M and N, we will have one ROC for K classified against M and N, another ROC for M classified against K and N, and a third one of N classified against K and M.
* F1 score[[16]](#footnote-15): F1 Score is a measure of precision and recall. Precision can be defined as true positives divided by sum of true positives and false positives, whereas recall can be defined as true positives divided by the sum of true positives plus false negatives. Precision can also be defined as the ratio of true positives to number of predicted positives and recall can also be defined as the ratio of true positives to the number of actual positives. F1 score ranges between 0 and 1, where 0 is a perfect mismatch of predicted to actual classes and 1 being the perfect match.
* Cohen Kappa[[17]](#footnote-16): The agreement between 2 raters or classes who classify M items into K mutually excl. Category is done by Cohen Kappa. It can also be used for comparing different classes or subjects into several categories. Cohen K. is analogues to accuracy, except that it is only used for measuring classification tasks. It is a normalized version of random chance on a particular dataset. It is a more useful measure to use on problems that have an imbalance in the classes (e.g. 60-40 split for classes 0 and 1 & can achieve 70% accuracy by predicting all instances are for class 0).
* Accuracy[[18]](#footnote-17): It is defined as the percentage of correctly classified data out of all the available data points.

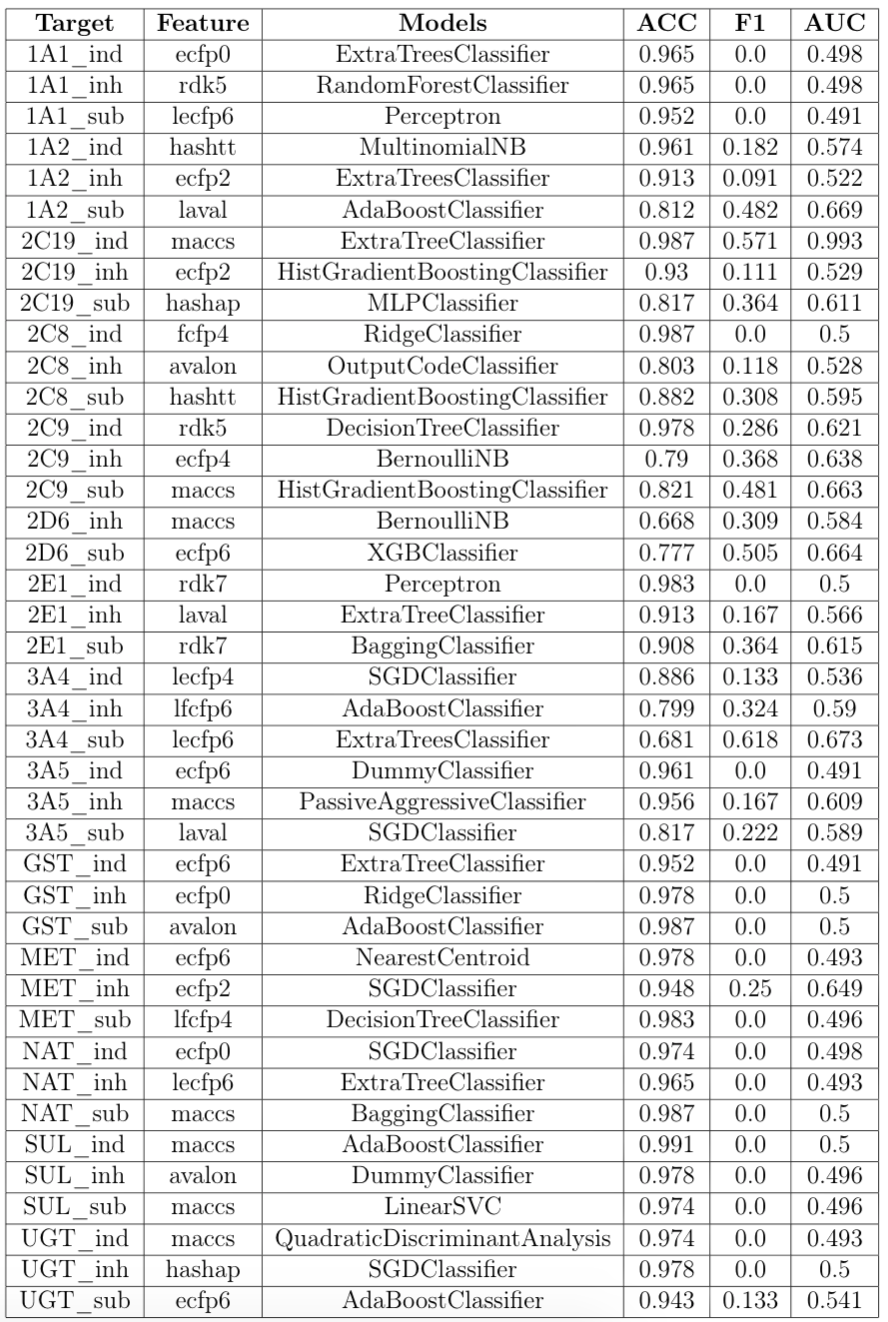
## 7 Results

Performance of all the models were measured across various metrics.   
Validation and test results for best models are shown below:

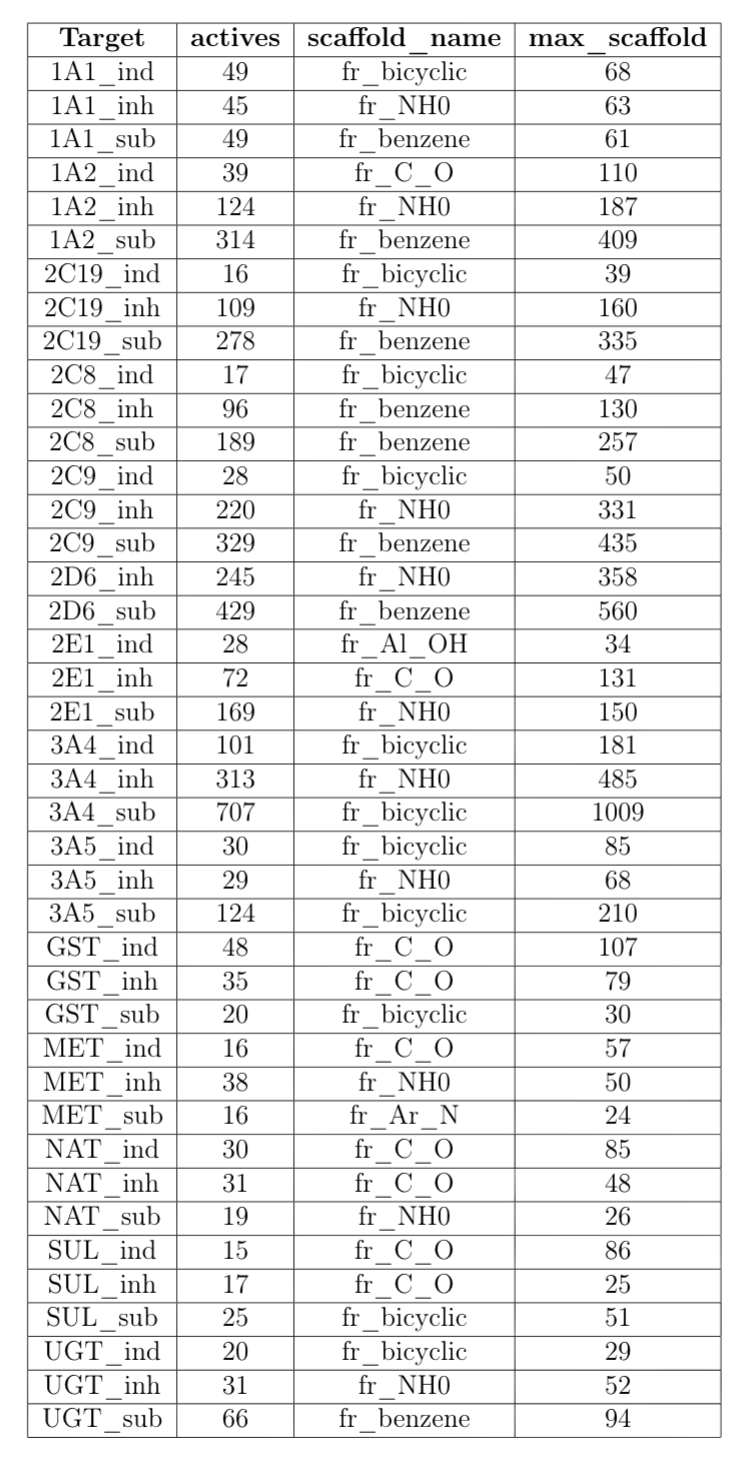
Below figure showing results on test set, for Phase I and II enzymes

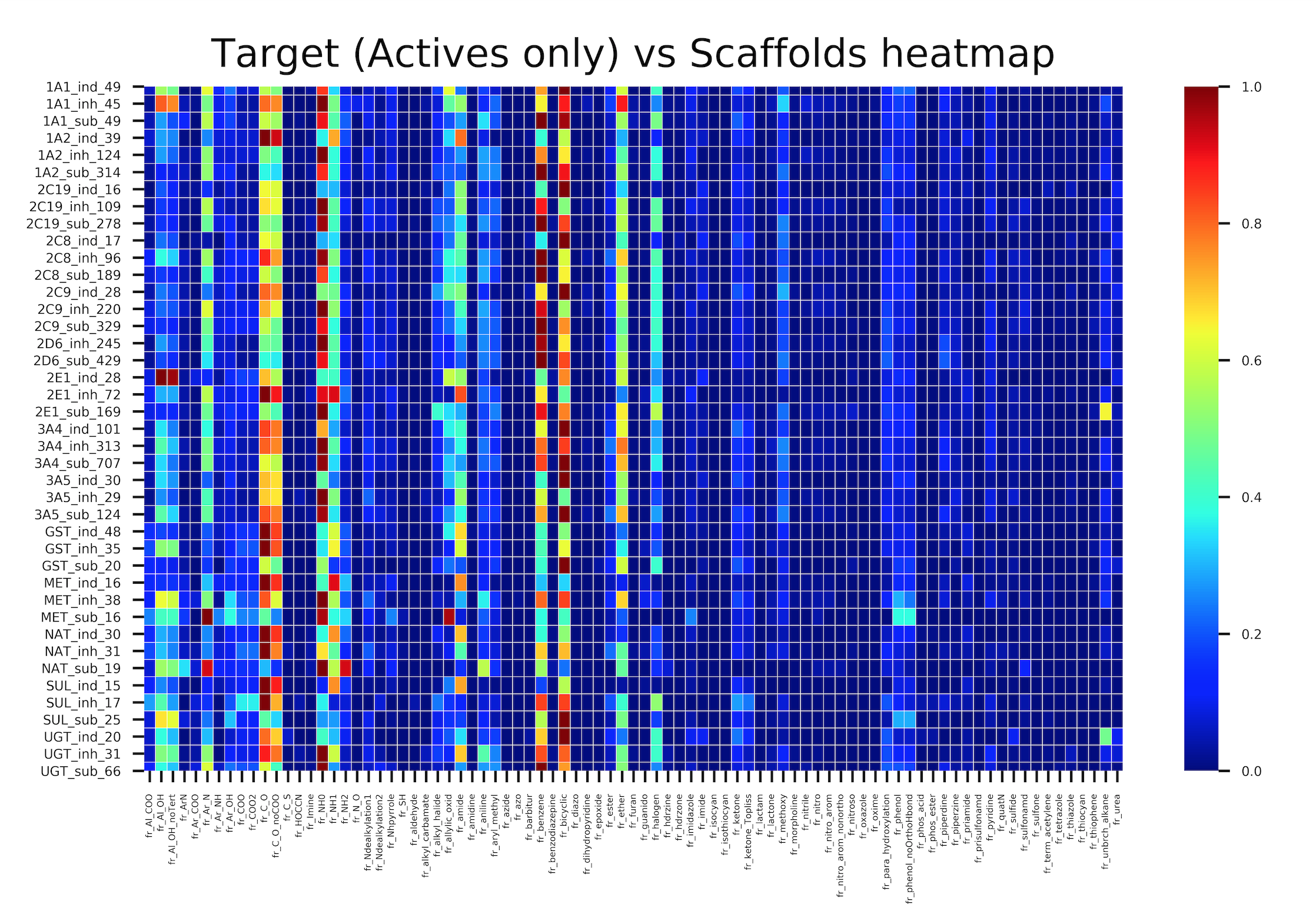


Summary of Validation Scores given below: Most of the enzymes showed good results with Long fingerprints. 3A4 substrate showed better results compared to other Phase I enzymes since data available in it was more.



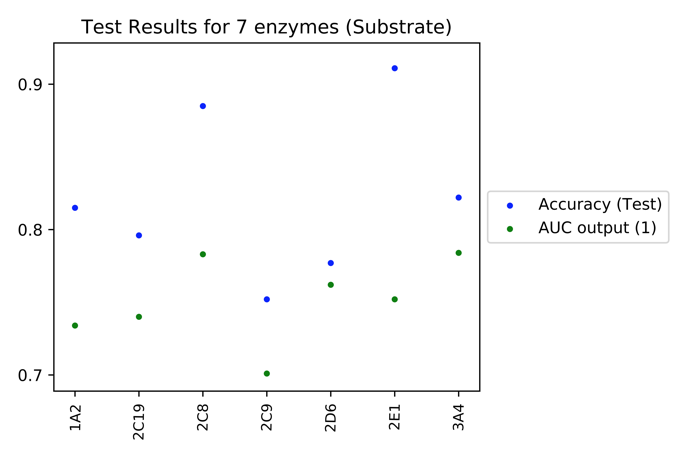
Summary of Test Scores given above:



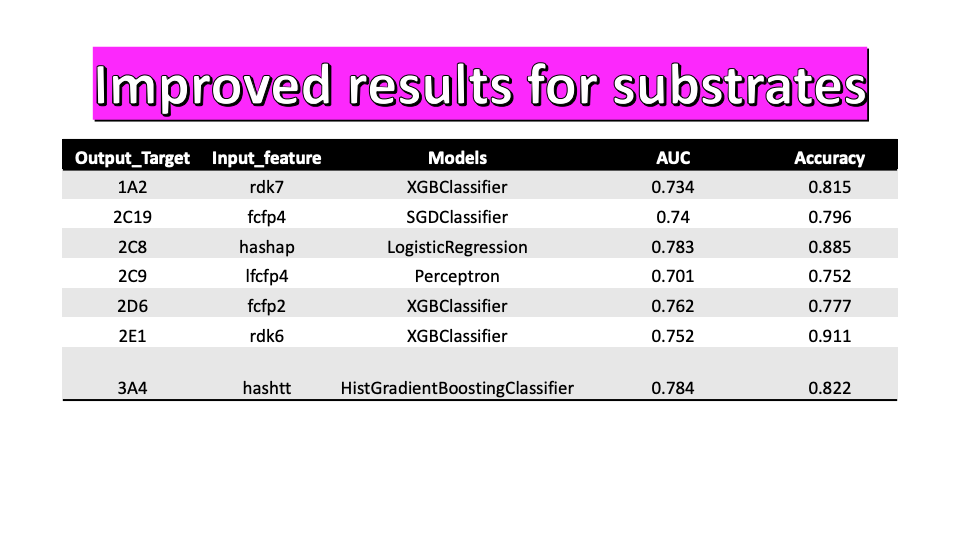
Below figure showing analysis of the scaffold present in each of the enzyme targets. Major scaffolds present were bicyclic and COO, as can be seen below

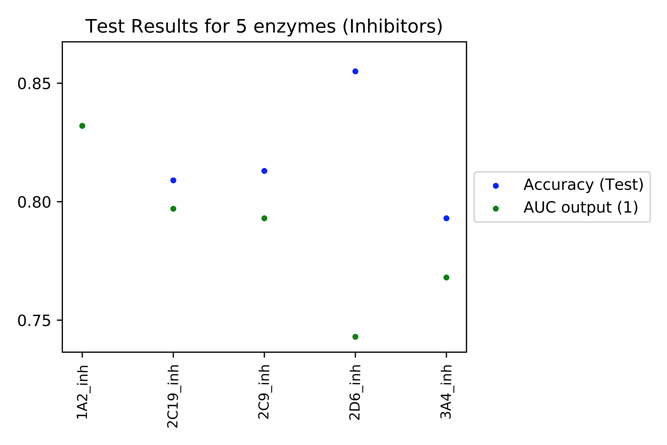
## 8 **Improved Results with SULT regression models**

For improving the AUC, additional data was collected from WhichP450 and PubChem

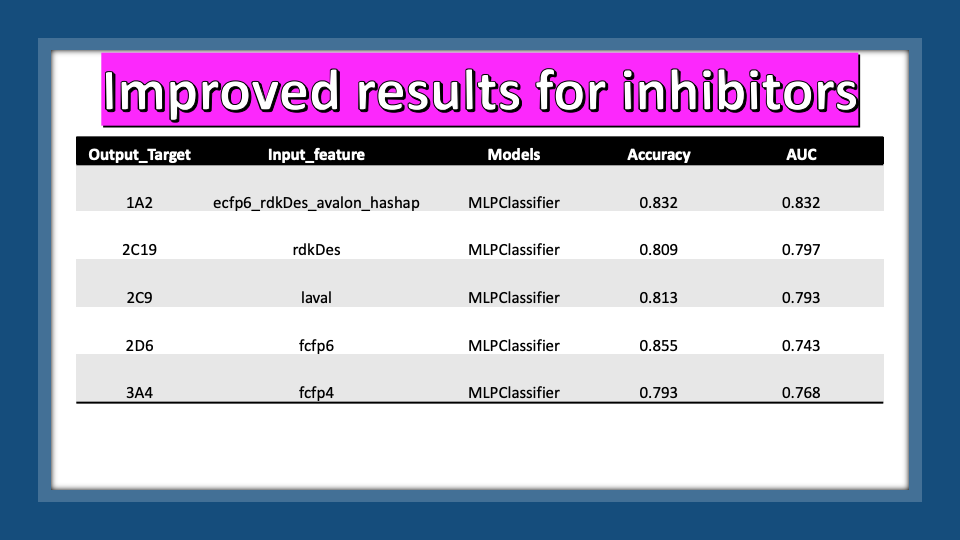


Data was split into 85:15 (Train:Test)



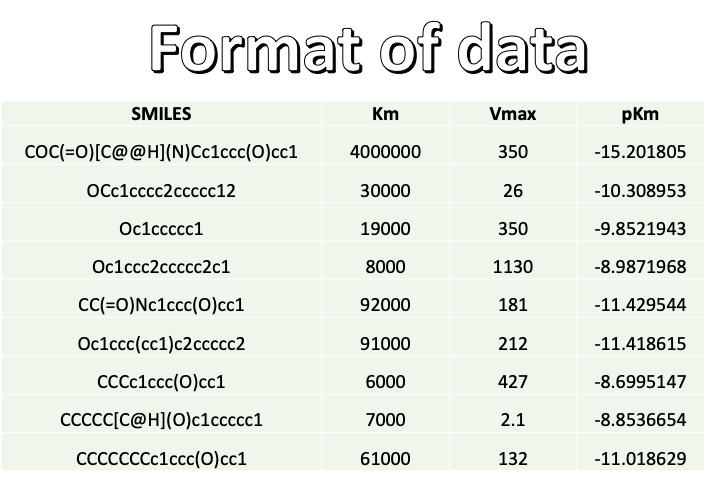


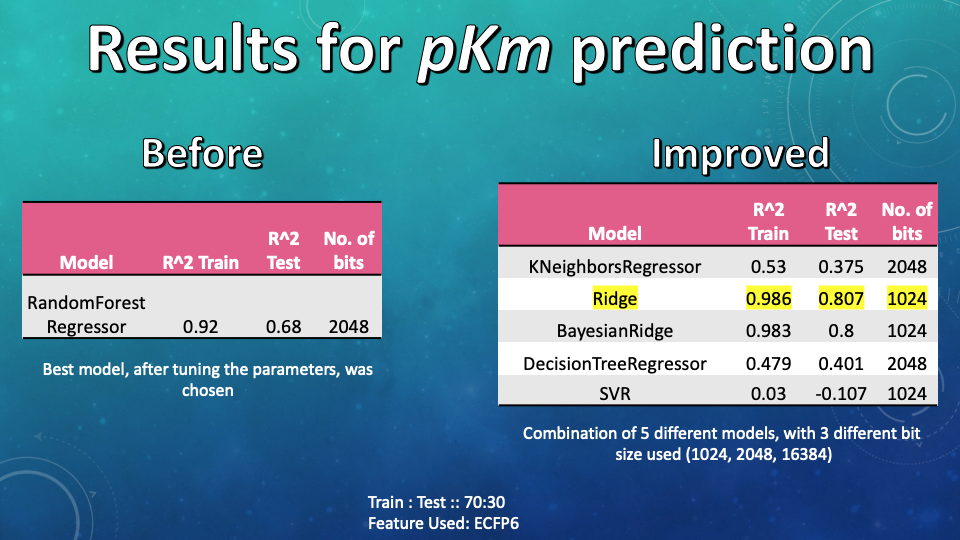
Data was split into 80:20 (Train:Test)

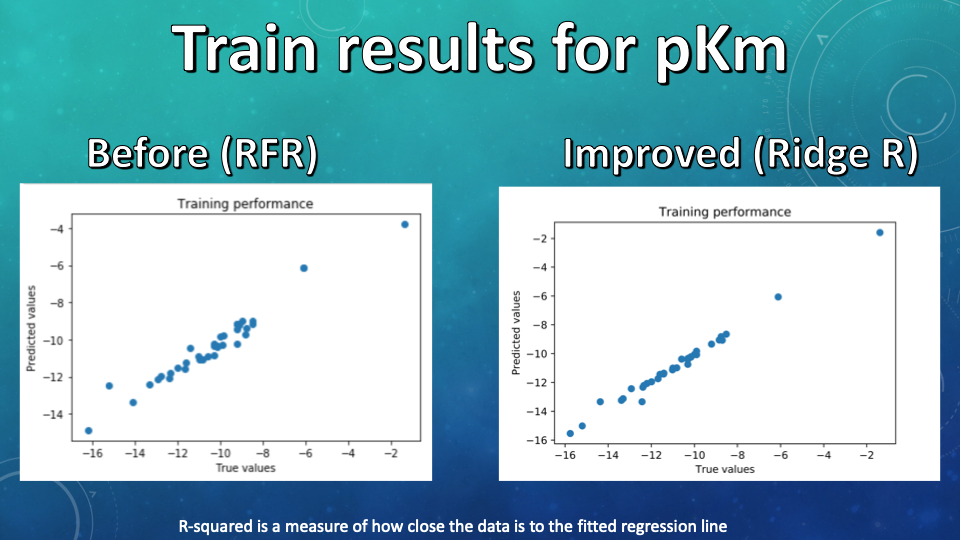


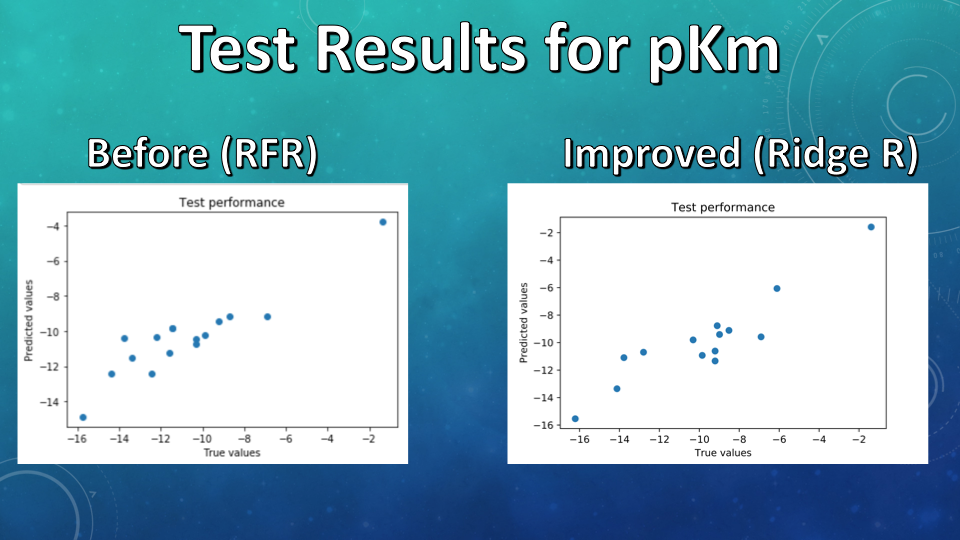
**Regression Models for SULT, for predicting Vmax and pKm values given below:**

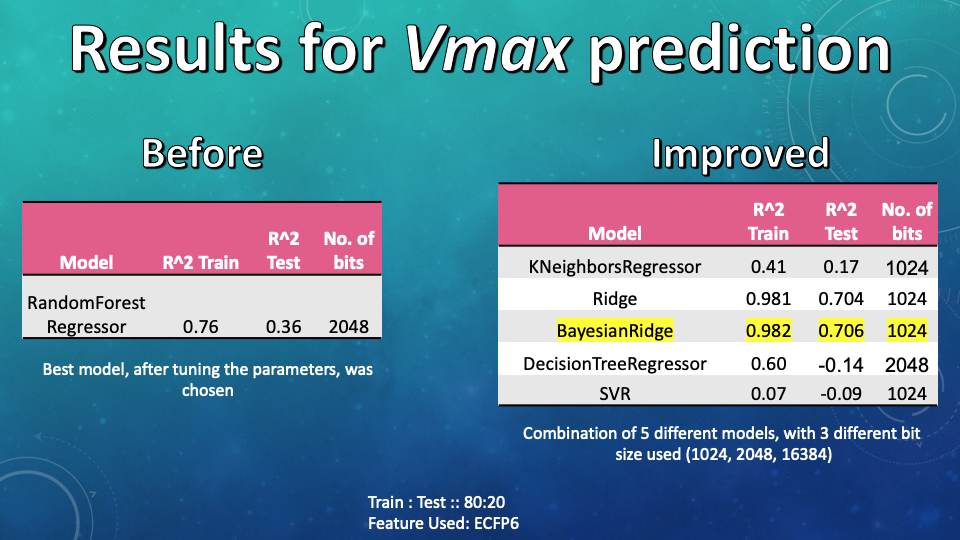
**Models were developed for 53 compounds (data collected from ChemBL) which are metabolised by SULT enzymes.**

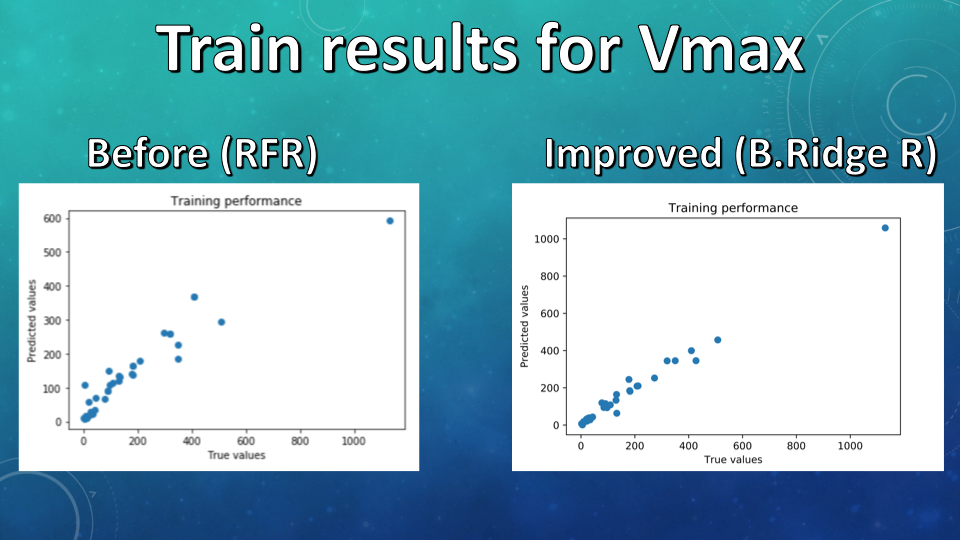
****

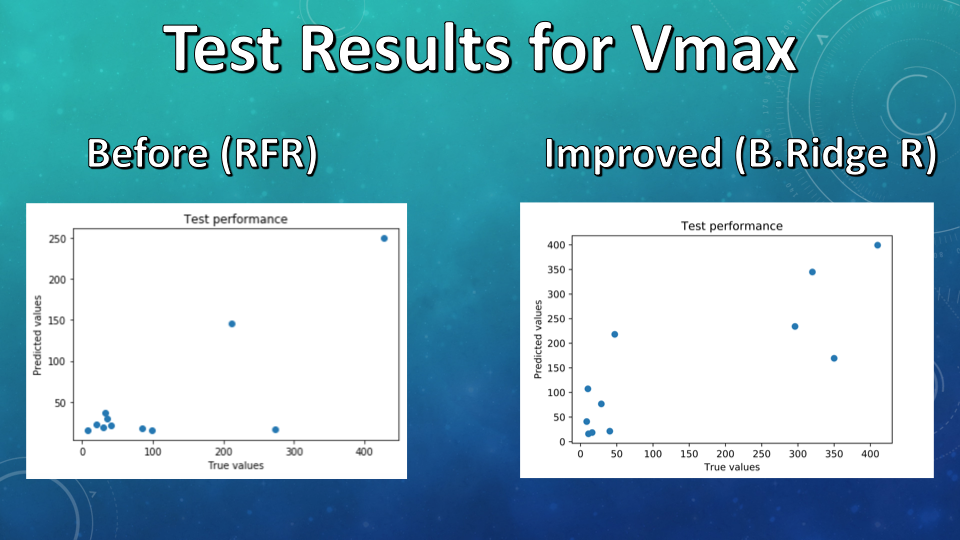
****

****

****

****

****

****

## **9.** **Bibliography/References :**

* <https://link.springer.com/chapter/10.1007/978-1-4615-4855-3_13>
* <https://www.ncbi.nlm.nih.gov/pubmed/16918470>
* <https://www.wiley.com/en-us/Drug+Metabolism+Prediction-p-9783527335664>
* <https://www.quora.com/Why-is-machine-learning-being-given-so-much-importance>
* <https://en.wikipedia.org/wiki/Cohen%27s_kappa>
* <https://www.rdkit.org/UGM/2012/Landrum_RDKit_UGM.Fingerprints.Final.pptx.pdf>
* <http://www.dalkescientific.com/writings/diary/archive/2014/10/17/maccs_key_44.html>
* <https://chemfp.readthedocs.io/en/latest/fingerprint_types.html>
* <https://pubs.rsc.org/en/content/articlelanding/2018/sc/c8sc00148k>
* <https://www.ncbi.nlm.nih.gov/pubmed/26108525>
* <https://www.ncbi.nlm.nih.gov/pubmed/23649703>
* <https://pubs.acs.org/doi/10.1021/jm049934e>
* <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0066566>
* <http://dmd.aspetjournals.org/content/41/7/1433>
* <https://www.nature.com/articles/srep31557>
* <https://www.jpharmsci.org/article/S0022-3549(16)32180-3/fulltext>

Supporting Information found [here](https://docs.google.com/document/d/1TBy-OkdS3wsmCsehi1bo5CO0vjyV5gEcnyJ8Ino35sc/edit?usp=sharing)

1. "Drug Metabolism Prediction | Johannes Kirchmair, Raimund ...." <https://zh.b-ok2.org/book/2359342/115704>. Accessed 9 Dec. 2019. [↑](#footnote-ref-0)
2. "Drug Metabolism Prediction | Johannes Kirchmair, Raimund ...." <https://zh.b-ok2.org/book/2359342/115704>. Accessed 9 Dec. 2019. [↑](#footnote-ref-1)
3. "XenoSite Web Predictor-Home - S. Joshua ...." <https://swami.wustl.edu/xenosite>. Accessed 27 Nov. 2019. [↑](#footnote-ref-2)
4. "SMARTCyp." <https://smartcyp.sund.ku.dk/>. Accessed 27 Nov. 2019. [↑](#footnote-ref-3)
5. "SuperCYP - Structural Bioinformatics Group." <http://bioinformatics.charite.de/supercyp/>. Accessed 27 Nov. 2019. [↑](#footnote-ref-4)
6. "Prediction of Cytochrome P450 3A4, 2D6, and 2C9 Inhibitors ...." <http://pubs.acs.org/doi/abs/10.1021/ci0500536>. Accessed 27 Nov. 2019. [↑](#footnote-ref-5)
7. "Prediction of CYP450 Enzyme-Substrate Selectivity Based on ...." 11 Oct. 2019, <https://www.ncbi.nlm.nih.gov/pubmed/31603319>. Accessed 27 Nov. 2019. [↑](#footnote-ref-6)
8. "Selenium with Python — Selenium Python Bindings 2 ...." <https://selenium-python.readthedocs.io/>. Accessed 27 Nov. 2019. [↑](#footnote-ref-7)
9. "Beautiful Soup 4 Python - Pythonforbeginners.com." 9 Mar. 2016, <https://www.pythonforbeginners.com/beautifulsoup/beautifulsoup-4-python>. Accessed 27 Nov. 2019. [↑](#footnote-ref-8)
10. "Transformer - Structural Bioinformatics Group." <http://bioinformatics.charite.de/transformer/>. Accessed 30 Nov. 2019. [↑](#footnote-ref-9)
11. "Prediction of Cytochrome P450 3A4, 2D6, and 2C9 Inhibitors ...." <http://pubs.acs.org/doi/abs/10.1021/ci0500536>. Accessed 30 Nov. 2019. [↑](#footnote-ref-10)
12. "Extended-Connectivity Fingerprints | Journal of Chemical ...." 28 Apr. 2010, <https://pubs.acs.org/doi/10.1021/ci100050t>. Accessed 27 Nov. 2019. [↑](#footnote-ref-11)
13. "Open-source platform to benchmark fingerprints for ligand ...." 30 May. 2013, <https://www.ncbi.nlm.nih.gov/pubmed/23721588>. Accessed 27 Nov. 2019. [↑](#footnote-ref-12)
14. "A software suite for cheminformatics, computational ... - RDKit." <http://www.rdkit.org/RDKit_Overview.pdf>. Accessed 27 Nov. 2019. [↑](#footnote-ref-13)
15. "The Area Under an ROC Curve." <http://gim.unmc.edu/dxtests/ROC3.htm>. Accessed 27 Nov. 2019. [↑](#footnote-ref-14)
16. "sklearn.metrics.f1\_score — scikit-learn 0.21.3 documentation." <http://scikit-learn.org/stable/modules/generated/sklearn.metrics.f1_score.html>. Accessed 27 Nov. 2019. [↑](#footnote-ref-15)
17. "Cohen's Kappa Statistic - Statistics How To." 8 Dec. 2014, <https://www.statisticshowto.datasciencecentral.com/cohens-kappa-statistic/>. Accessed 27 Nov. 2019. [↑](#footnote-ref-16)
18. "sklearn.metrics.accuracy\_score — scikit-learn 0.21.3 ...." <https://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html>. Accessed 27 Nov. 2019. [↑](#footnote-ref-17)