THESIS

A NEW APPROACH TO ADDRESSING TWO PROBLEMS IN PHARMACOKINETICS AND PHARMACODYNAMICS USING MACHINE LEARNING

Submitted by

Sohaib Habib

Department of Chemical and Biological Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Summer 2020

Master's Committee:

Advisor: Brad Reisfeld

Brian Munsky

Patrick Shipman

ABSTRACT

A NEW APPROACH TO ADDRESSING TWO PROBLEMS IN PHARMACOKINETICS AND PHARMACODYNAMICS USING MACHINE LEARNING

In this work, machine learning was applied to develop solutions for two problems related to drug pharmacokinetics (PK) and pharmacodynamics (PD). The first problem was finding a way to easily predict important pharmacological measures accurately representative of those from simulation results computed via a sophisticated model for drug absorption via oral dosing. This model (OpenCAT: Open source Compartmental And Transit model) comprises a system of differential equations describing the absorption of drugs into the gastrointestinal tract, including such factors as drug dissolution and spatially-distributed absorption, metabolism, and transport. For this problem, a machine learning framework was built to develop a self-contained random forest representation of the model predictions that could be queried for critical PK parameters such as maximum plasma concentration (C_{max}), time at which the maximum concentration occurs (t_{max}), and the area under the concentration-time curve (AUC). The random-forest representation was able to generate predictions for the targeted PK parameters close to the solution of the original OpenCAT model over a wide range of drug characteristics. The second problem involved predicting the pharmacodynamics (cholinesterase reactivation) of antidotes for nerve agents. In this case, a machine learning framework was built to use experimental data and corresponding theoretically-derived chemical descriptors to predict the pharmacodynamics of new candidate antidotes against both tested and untested nerve agents. Overall, this project has demonstrated the

utility of machine learning approaches in the fields of drug pharmacokinetics and pharmacodynamics.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Project Objective

Pharmacokinetics (PK) is the quantitative study of drug absorption, distribution, metabolism, and elimination (ADME). It is the field of pharmacology that addresses "what the body does to the drug". Pharmacodynamics (PD) is the field of study concerned with the effects of drugs on the biochemistry and physiology of the organism and attempts to quantify "what the drug does to the body." The two fields are complementary, and both are necessary in the development, assessment, and approval processes for nearly all drugs. Moreover, both PK and PD analyses are extensively used in the field of toxicology to help assess the safety and risk associated with environmental pollutants.

The project described here focused on developing a novel approach to solving two problems in drug pharmacokinetics and pharmacodynamics. Each of the main chapters of this work detail the problem and approach taken to solve a problem.

In Chapter 1, the aim is to describe the bridge between results from physiologically-based pharmacokinetic (PBPK) modeling via the OpenCAT model with the universe of machine learning. PBPK models have proven useful in describing the pharmacokinetics of drugs in a variety of contexts (Ando, Hisaka, & Suzuki, 2015)]; however, because these models are grounded in ordinary differential equations (ODEs), they are often difficult to incorporate into a larger simulation framework or be utilized in contexts where ODE solvers are computationally intensive or inconvenient. Thus, the successful development of the machine learning-based alternative with the ability to circumvent PBPK models and accurately predict PK parameters could offer an attractive alternative to drug-development scientists.

In Chapter 2, the focus is to describe the work aimed at describing how machine learning techniques can be used to facilitate the development of antidotes to nerve agents. In particular, this chapter focuses on how the synthesis of *in vitro* data and computational estimation of chemical properties can be used within a machine learning environment to provide predictive power to identify potential compounds that could act as antidotes against both quantified and unquantified nerve agents.

1 Chapter I: Machine Learning as Means to Augment the Utility of PBPK models

1.1 Introduction

In the process of drug discovery and evaluation, potential drug candidates are generally screened based on their ADME (absorption, distribution, metabolism, and excretion) properties. These are often determined using *in vitro* analysis and *in vivo* animal model (Gobeau, Stringer, Buck,, Tuntland, & Faller, 2016). Alternately, investigators can make use of the biologically-based mathematical models implemented in computational frameworks (i.e., *in silico* approaches) to make predictions about these properties. *In silico* methods have the potential to reduce drug development and assessment time and reduce the number of *in vivo* experimental procedures required for compound selection and development (Agoram, 2001). One of the most promising of such techniques is physiologically based pharmacokinetic (PBPK) modeling, which has a long history of use in the risk assessment for environmental pollutants and has begun to be used with increasing frequency in the drug development process.

For orally-administered drugs, where details of drug absorption are particularly important, the PBPK ACAT (advanced compartmental and transient) model is often used to predict critical pharmacokinetic quantities. This model, developed by Agoram et al. (Agoram, 2001), is an improved version of CAT model described by Yu and Amidon (1999) (Yu & Amidon, 1999). The ACAT model focuses on the human gastrointestinal (GI) tract and consists of nine compartments linked in series, each of them representing a different segment of the GI tract (stomach, duodenum, two jejunum compartments, three ileum compartments, caecum, and ascending colon). To account for the drug that is unreleased, undissolved, dissolved, and absorbed (entering the enterocytes), a

further compartments subdivision was applied (Djuris, 2013). Figure 1-1 shows the structure of this model (Agoram, 2001).

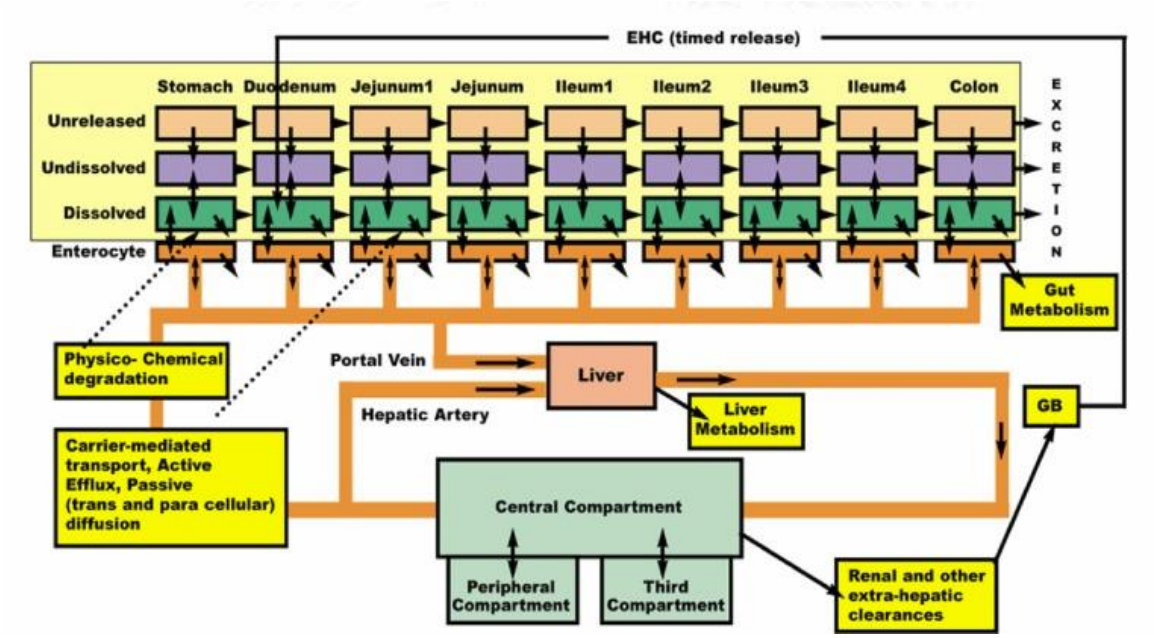


Figure 1-1 ACAT model schematic

In addition to the investigation of dissolution-rate limited absorption, the ACAT model facilitates the exploration of the effect of drug formulation release rate on oral pharmacokinetics. This model has been tested by a number of investigators, including Lalka et al (Lalka, Griffith, & Cronenberger, 1993)., who investigated the ability of the model to improve PK prediction for drugs that undergo significant first-pass metabolism and Ando et al. (Ando, Hisaka, & Suzuki, 2015), who examined the model predictions across a wide range of drugs.

Although PBPK models, like the ACAT model, have proven very useful in predicting drug PK, because for such models, to obtain the targeted PK parameters one has to follow a multi-steps procedure with each step require some knowledge of computer programing. For this application, one must prepare and execute some MCSim codes involving solving systems of ordinary

differential equations to obtain concentration profile/s for drug/s. Next, the resulted profile/s require further statistical processing to calculate the targeted PK parameters. This procedure is not convenient to everyone and is often difficult to implement in certain environments, computationally intensive, and difficult to incorporate into larger simulation frameworks. A novel approach to address these limitations involves machine learning (ML). In particular, a self-contained computing modules (or callable function) could be created that would take as input drug-specific information and would output key pharmacokinetic parameters, such as maximum concentration (C_{max}), time at which the maximum concentration occurred (t_{max}), area under the concentration-time curve (AUC), clearance rate (CL), and mean residence time (MRT). In this chapter, we show how such a module can be created based on machine learning algorithms trained on PBPK model output. Though machine learning is increasingly used in drug discovery (Hartmanshenn, Scherholz, & Androulakis, 2016) there is little work in the literature on using ML to facilitate pharmacokinetic analyses for drugs.

Such a callable ACAT function could potentially replace much of the tedious work associated with coding and running the ACAT PBPK model; moreover, such a module could be incorporated into a larger drug evaluation framework that could facilitate more rapid screening of drug candidates.

1.2 Methodology

1.2.1 Data Gathering and preparation

Drug properties were collected to span the four categories of the Biopharmaceutics Classification System (Charalabidis, Sfouni, Bergström, & Macheras, 2019): Class I - high permeability, high solubility; Class II - high permeability, low solubility; Class III - low

permeability, high solubility; and Class IV - low permeability, low solubility. To compute drug pharmacokinetics, an open-source implementation of the ACAT model was created using current literature information related to the ADME associated with oral drug administration and GI physiology. This model, OpenCAT (Bois, 2018) was implemented using the MCSim language and software (GNU MCsim, 2020).

To span drug properties across the classes in the Biopharmaceutics Classification System, the following physical properties were varied: molecular weight, molar volume, acidic dissociation constant, effective permeability, precipitation rate constant, drug solubility, particle radius, and drug density. The values of each property were distributed uniformly based on the limits for actual drugs found in the literature. In addition to the above properties, ratios of the drug unbound fraction over its partition coefficient in each compartment, metabolic parameters, dose magnitude, and subject body weight were varied. All told, these parameter variations led to 15,000 different combination of parameters to be explored. These data were then used as input to OpenCAT model, resulting in a plasma concentration-time profile for each combination.

Two of the measures of interest when evaluating drug ADME are the maximum drug concentration, C_{max}, and the time at which the maximum concentration occurs, t_{max}. Also, of interest are the area under the concentration-time curve, AUC, which a measure for the actual drug exposure,

$$AUC = \int_0^\infty C(t)\,dt$$

and the total clearance, CL,

$$Clearance = \frac{dose}{AUC}$$

1.2.2 Model building

The approach when building the machine learning model was to use the results from PBPK simulations in which parameters were systematically varied over appropriate ranges to develop a model that could map between drug properties as inputs and important pharmacokinetics parameters as outputs. The specific approach used to achieve this aim was random forest (RF) regression implementation (Random Forest Regression, 2020) from The Python package scikit-learn(v 0.22.2) (sci-kit learn machine learning in python, 2020) .

There are several important elements when building such a machine learning model, including optimizing the regressor's hyper parameters, optimizing the size of the training set out of the data set, and assessing feature importance (Appendix I).

The fact that we have multiple PK parameters (i.e., labels) associated with each drug related properties combination (i.e., features) suggested two approaches to the design of our RF model. The first approach would be to create distinct *individual regressors*, one to predict each PK parameter. The second approach would be to develop a single, *multi-output (MO) regressor* that can predict a vector of PK parameters such that each component of the predicted vector represents a PK parameter. The key factor used in determining which of these approaches was relevant was the degree to which the predictions matched the 'data' from the PBPK model.

1.2.2.1 Individual regressor

See Appendix I for details about the characteristics and approach used to develop the individual regressors.

1.2.2.2 Multi-output regressor

Within the software framework used, the RF regressor natively supports multi-output classification/regression problems. In the multi-output problem, the regressor is to predict a vector of the required outputs (labels) based on the set of input features.

1.2.2.3 Parametric (feature) variations

The range and number of values used for each of the PBPK parameters are shown in Table 1-1ACAT parameters' distribution These values comprised the RF model features.

Table 1-1ACAT parameters' distribution

Parameter	Abbreviation	Lower bound	Upper bound	unit
Molecular mass	MM	200	1200	$g/gmol$
Molar volume	Mol_vol	200	1200	ml/mol
acidic dissociation constant	pKa	0	14	ND
particle radius	G_radius	5	50	µm

drug density	G_Density	1	2	g/ml
Drug Solubility	Solubility	1×10^{4}	1×10^{6}	µg/L
precipitation rate constant	K_precip	1×10^{-2}	1	hr^{-1}
Effective permeability of GI tract epithelia	P_{eff}	1	7	ND
ratios of the drug unbound fraction over its partition coefficient in compartment i	$Kpuu_i$	1×10^{-2}	1	ND
Dose magnitude	Dose	1	5×10^{3}	µmol
Subject body mass	BDM	45	125	Kg
metabolic parameters	Km_met_vitro	1×10^{-2}	1×10^{4}	µM
	Vmax_met_vitro	1×10^{-6}	1×10^{-2}	µmol/min. mg microsomal proteins

1.2.2.4 Methods

The general modeling workflow is shown in Figure 1-2. The core of the methodology was as follows: for each variation of features, a PBPK simulation was conducted and the outputs (labels) C_{max}, t_{max}, and AUC/dose were computed. In total, 15000 combinations were simulated to create the dataset. Results were evaluated based on the mean squared difference between the RF model predictions and those from the full PBPK model.

Training and test sets were randomly taken from the full datasets as follows:

- 20 % of the original dataset was reserved for accuracy verification.
- The remaining 80% of the dataset was split into training and testing datasets.

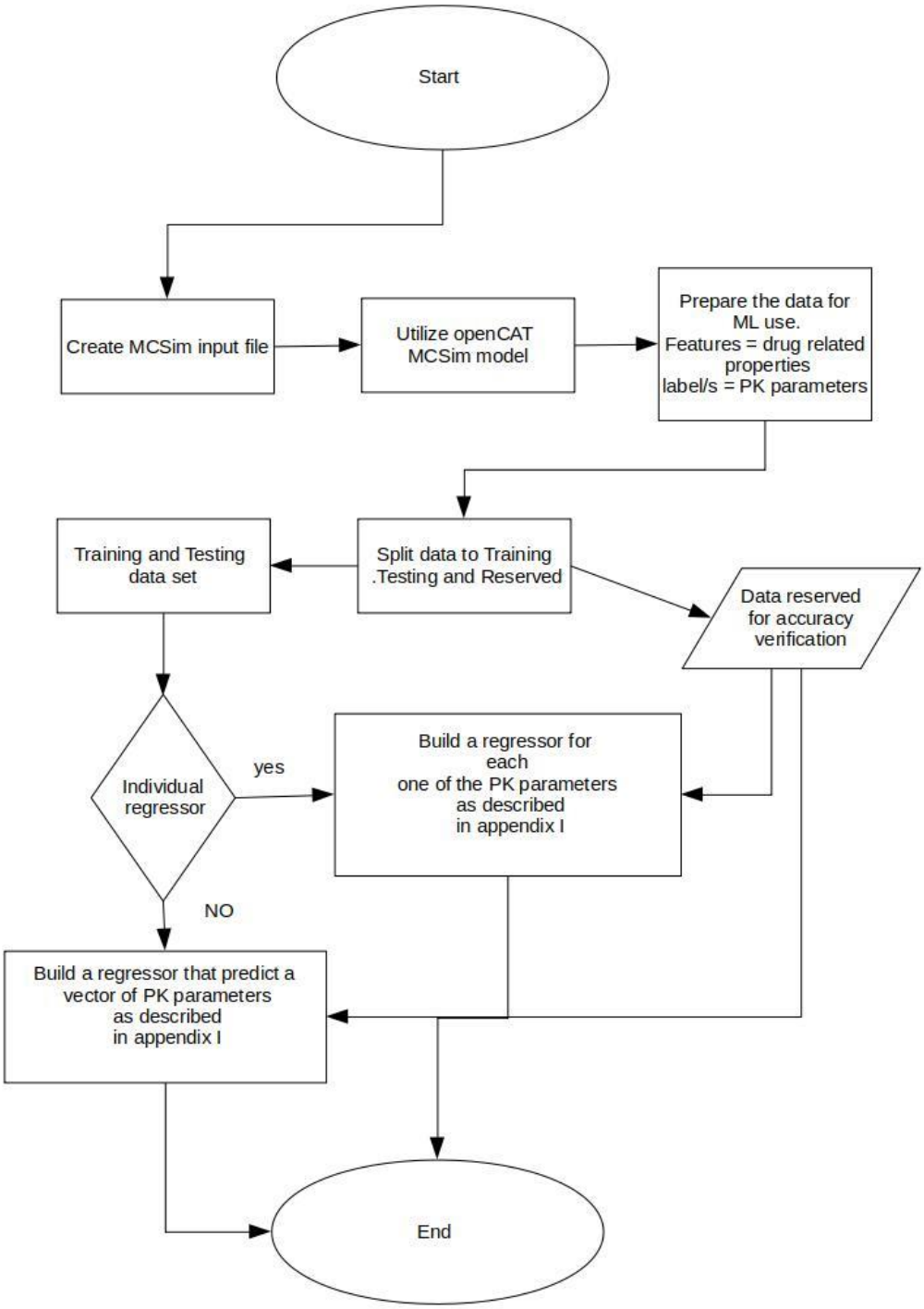


Figure 1-2 Flowchart detailing the construction of the random forest model

1.3 Result and Discussion

1.3.1 Individual regressor

1.3.1.1 Full features training

The predictions of these trained regressors compared to the actual output of the OpenCAT model is shown in Figure 1-3. The mean squared error (MSE) for the individual regressors (C_{max}, t_{max}, AUC/dose) was 0.025, 0.002, 0.043 for the training data and 0.103, 0.007, 0.145 for the testing datasets.

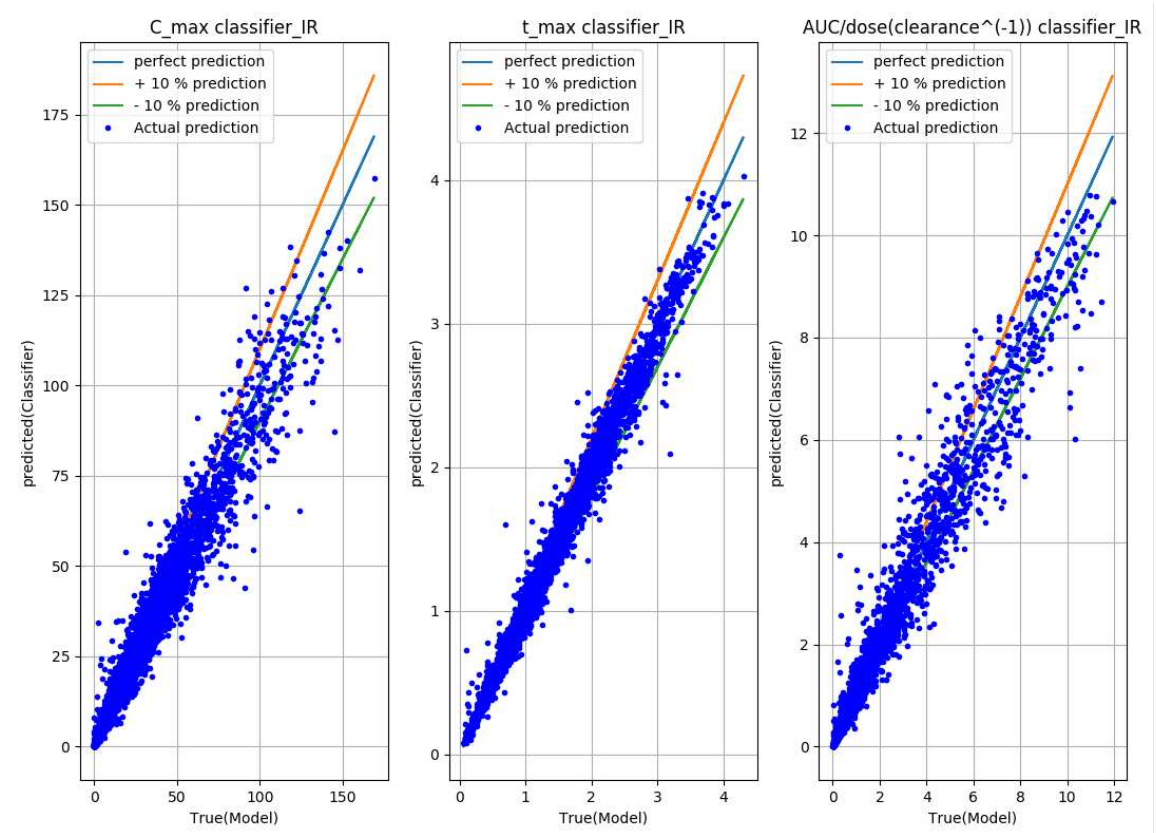


Figure 1-3 individual regressor accuracy full feature training

In general, most of the values predicted by the RF models were within 20% of those of the differential equation-based OpenCAT model. In order to further quantify the accuracy of the regressor predictions, histogram plots were generated to show the probability of obtaining a prediction with a certain error percentage for each regressor Figure 1-4 . The abscissa of the histogram plots is error percentage ranged from -100% to +100%. The axis has been divided in to 11 bins creating a bin with a width of 20. The ordinate of the plots represents the probability of obtaining certain error percentage out of a regressor divided by the bin's width.

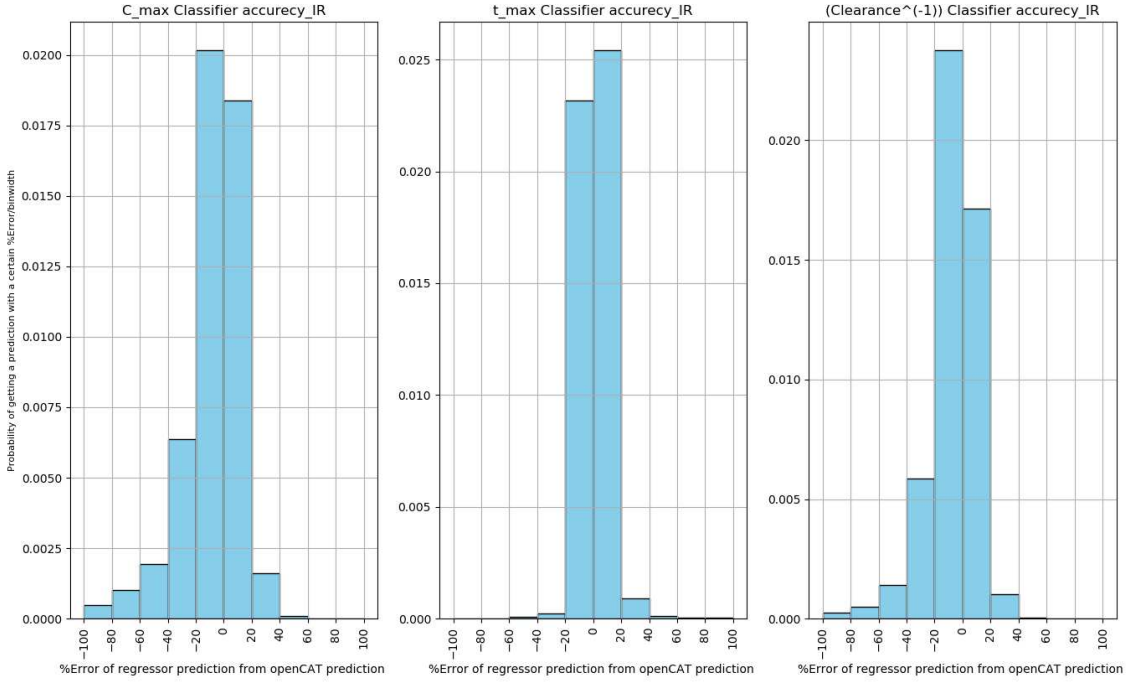


Figure 1-4 Histogram representation of individual regressor accuracy

The probabilities of the RF model being within 20% of the OpenCAT model predictions are 0.765, 0.968, and 0.820 for C_{max}, t_{max}, and AUC/dose, respectively. Encouragingly, all the three histograms show low to zero probability of getting predictions with error percentage greater than 40%.

In evaluating a RF model, it can be important to assess the influence of each of the features on the accuracy of the predictions. Figure 1-5 shows the results of such an assessment.

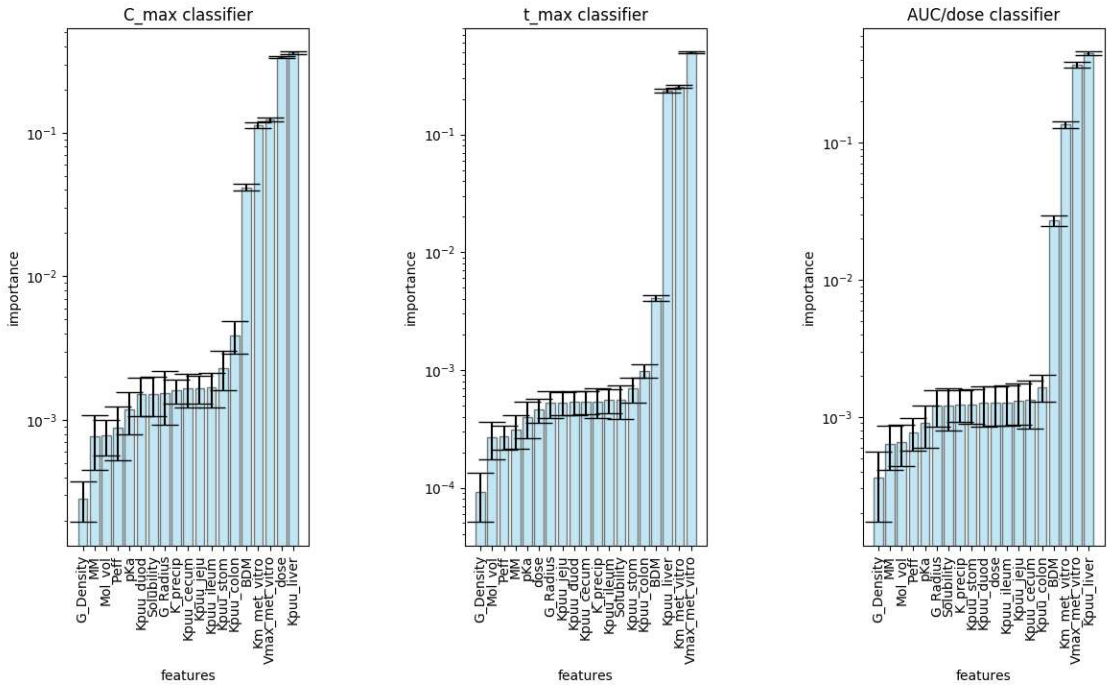


Figure 1-5 Feature importance for the individual regressors

This figure shows that the features of most importance in predicting C_{max} were the dose magnitude, the patient's body weight, the two metabolic constants, and the ratio of the drug fraction unbound over its partition coefficient in the liver ($Kpuu_{liver}$) are the most influential features. For t_{max}, the metabolic constants were again important constants along with $Kpuu_{liver}$. Finally, for the prediction of AUC/dose, $Kpuu_{liver}$, the metabolic constants, and the patient's body weight were the most important features.

1.3.1.2 Training using reduced features

To evaluate the extent to which a reduced set of features would affect the regressor accuracy, an additional set of simulations was conducted. The goal of this study was to determine (i) which set of reduced feature set would lead to the lowest mean squared error compared to the OpenCAT predictions and (ii) to compare the accuracy of predictions from the reduced feature set with those from the full feature set.

The reduced feature set was found by iteratively computing the feature importance among the three PK measures that reduced the overall MSE. The resulting features corresponding to the lowest overall MSE for the C_{max} regressor were (i) dose magnitude, (ii) patient's body weight, (iii&iv) the two metabolic constants, and(v) $Kpuu_{liver}$. For the t_{max} regressor the reduced feature set that correspond to the lowest MSE consists of (i&ii) the two metabolic constants, (iii) $Kpuu_{liver}$, (iv) patient's body weight, (v) $Kpuu_{colon}$, and (vi) $Kpuu_{stomach}$. Finally, for the AUC/dose regressor the reduced set consists of (i) $Kpuu_{liver}$, (ii&iii) the two metabolic constants, (iv) the patient's body weight, and (v) $Kpuu_{colon}$.

Similar to the previous analysis, histograms were generated to indicate the probability of obtaining a prediction within a certain error percentage when training with a reduced features set Figure 1-6.

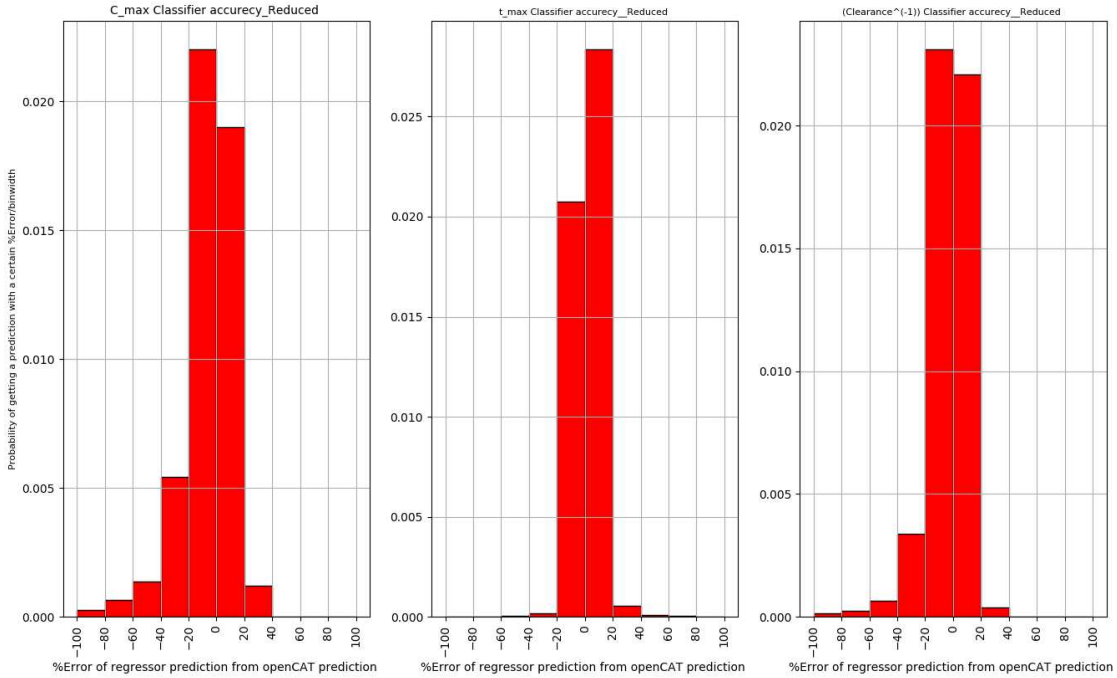


Figure 1-6 individual regressor accuracy Histogram Reduced features training

In this case, the probabilities of the RF model being within 20% of the OpenCAT model predictions are 0.826, 0.982, and 0.9 for C_{max}, t_{max}, and AUC/dose, respectively. The histograms show exceptionally low probability of predicting a PK parameter value with an error percentage greater than 40 %.

1.3.2 Multi-output regressor

1.3.2.1 Full features training

For end-user convenience and to eliminate the computational expense that is associated with training a standalone regressor for each PK, a regressor was developed to predict all the PK parameters at once. Initially, the training of this multi-output classifier was carried out using the full features dataset. The predictions of this trained multi-output regressor compared with the outputs of the OpenCAT model is shown in Figure 1-7. As for the individual regressors, the mean

squared error versus the OpenCAT results was used as the evaluation metric for the approach. In this case, this value was computed to be 0.0487 and 0.175 for training and testing data sets respectively.

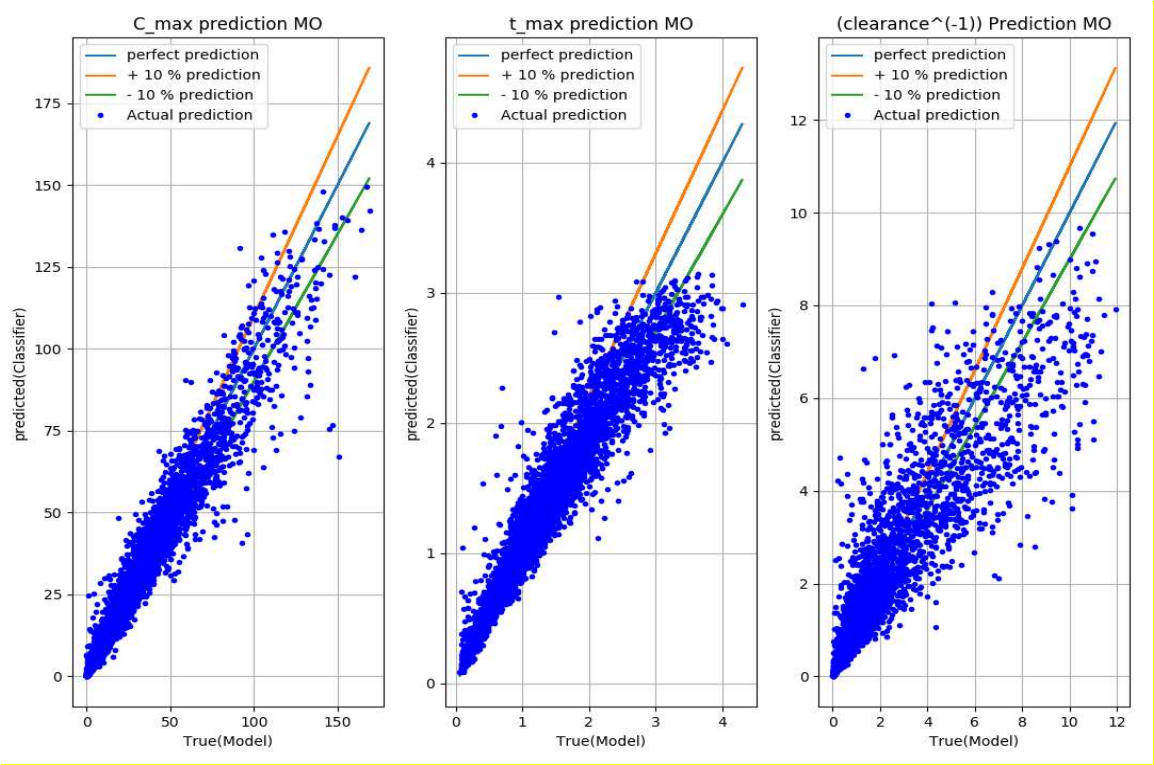


Figure 1-7 Accuracy of the multi-output regressor using full feature training

The three panels of the figure represent the three components of the predicted vector that results from the multi-output classifier. The corresponding probability histograms are shown in Figure 1-8, which depicts the probability of obtaining a predicted value for a PK parameter within a certain error percentage range.

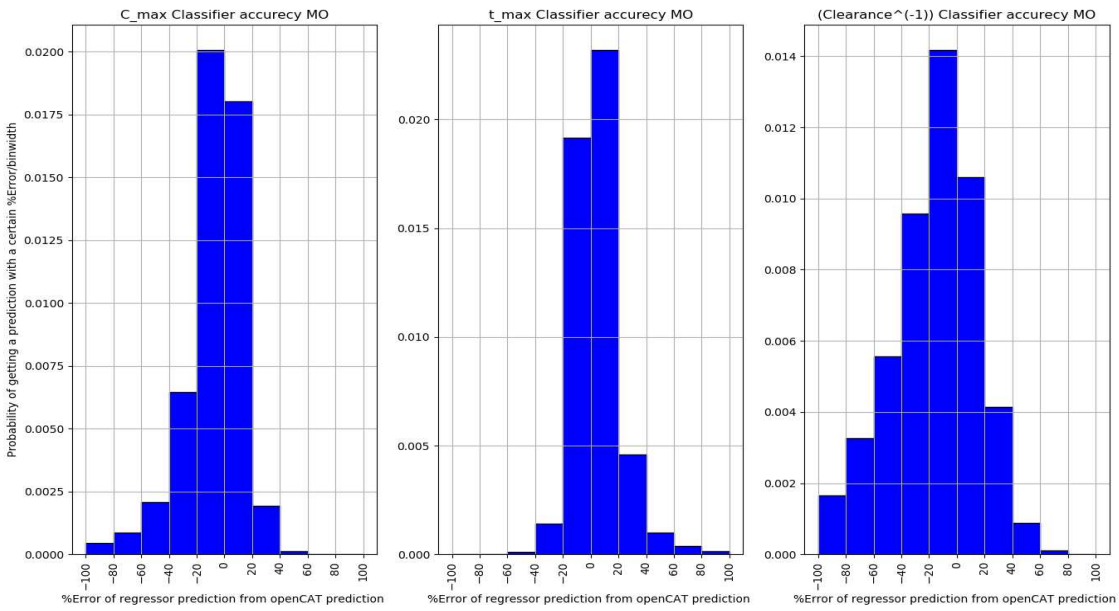


Figure 1-8 Multi-output regressor accuracy Histogram Full features training

In this case, the probabilities of the MO regressor being within 20% of the OpenCAT model predictions are 0.763, 0.844, and 0.493 for C_{max}, t_{max}, and clearance rate, respectively. As with the previous regressors, the histograms show very low probability of predicting a PK parameter value with an error percentage exceeding 40 %.

As with regressors described earlier, it is enlightening to assess the feature importance of the multi-output regressor. Figure 1-9 illustrates the results of this assessment.

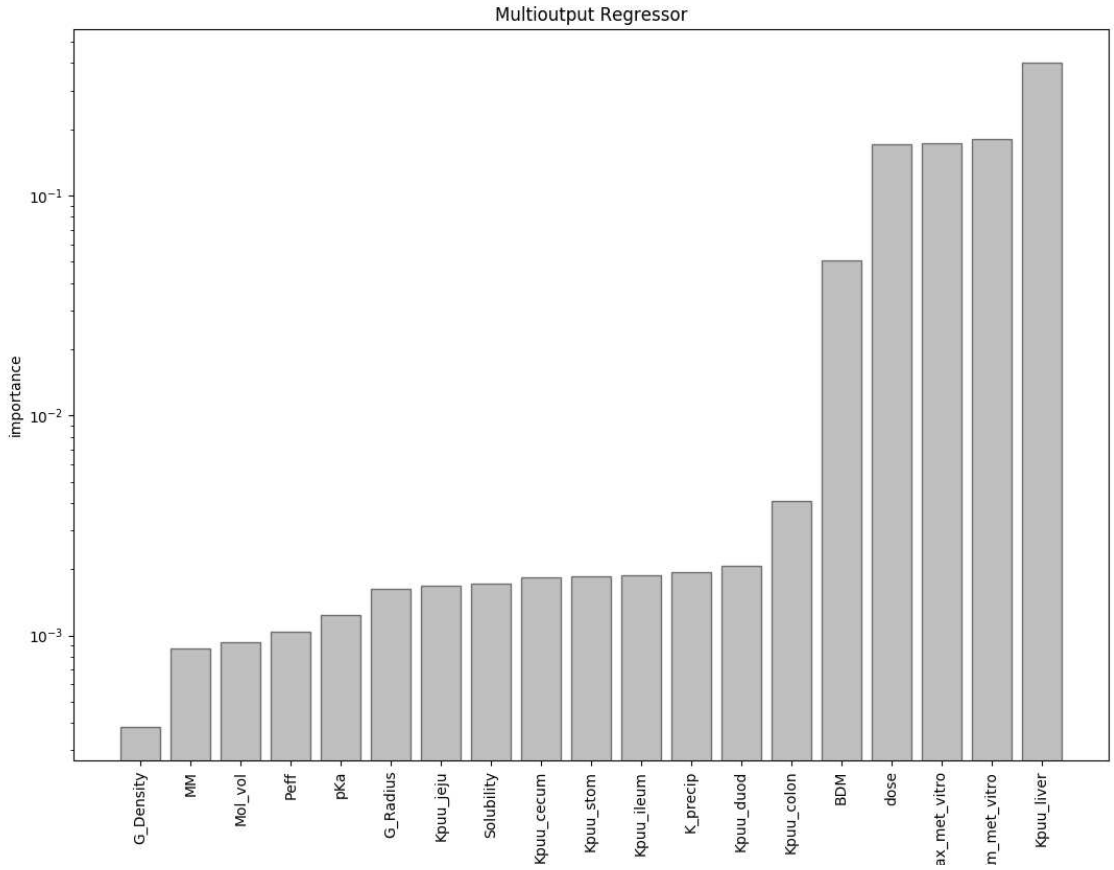


Figure 1-9 Feature importance for the multi-output regressor

This figure illustrates that dose magnitude, patient's body weight, the two metabolic constants, and the ratio of the drug fraction unbound over its partition coefficient in the liver were found to be the features most influential in the decision making process inherent in this regressor.

These influential features were then used to train a reduced feature MO regressor in a similar manner to that described earlier. The accuracy of this regressor was then assessed via probability histograms Figure 1-10.

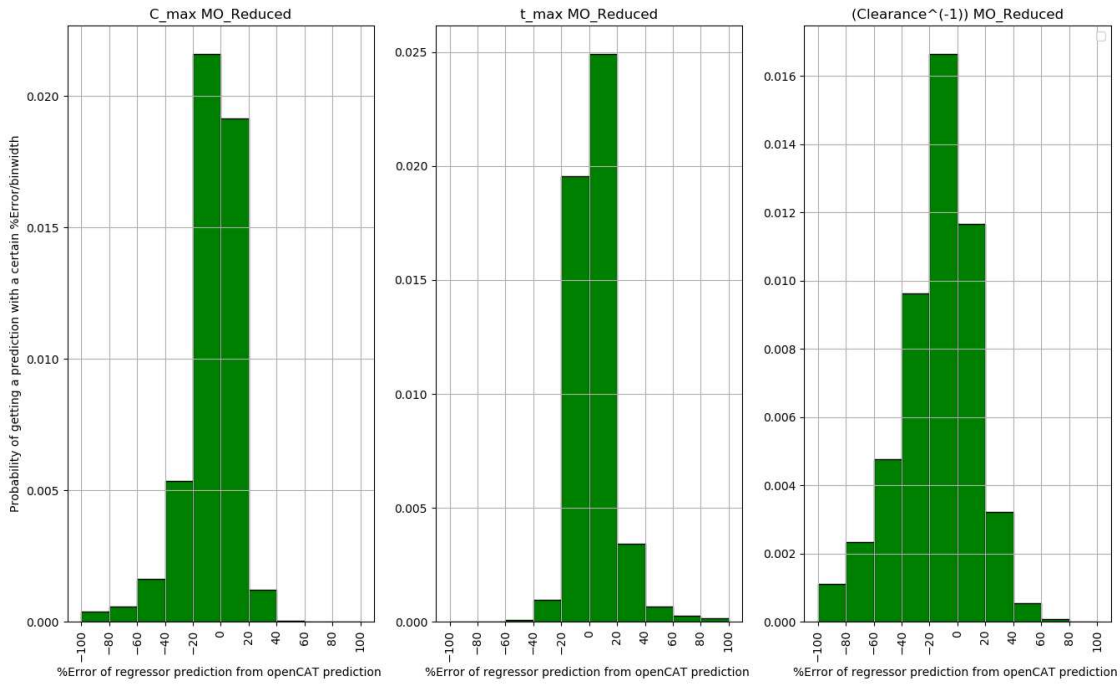


Figure 1-10 Multi-output regressor accuracy using reduced features training

The probability for obtaining a prediction that falls within the 20% error range for the values of the maximum plasma concentration, time at which this maximum value occurs, and for AUC/dose were 0.814, 0.886, and 0.565, respectively.

1.3.3 Comparison of regressor performance

A brief summary of some of the salient results noted above is contained in Table 1-2, which compares the agreement between the regressor output versus that of the OpenCAT model at various probability thresholds.

Table 1-2 Probability of obtaining agreement with the openCAT model

Approach / PK parameter	±20			±40		
	C_{max}	t_{max}	AUC/dose	C_{max}	t_{max}	AUC/dose
Individual regressor with full feature training	0.765	0.968	0.820	0.928	0.990	0.949
Individual regressor with reduced feature training	0.826	0.982	0.9	0.957	0.997	0.977
Multi-output regressor with full feature training	0.763	0.844	0.493	0.927	0.961	0.765
Multi-output regressor with reduced feature training	0.814	0.886	0.565	0.946	0.976	0.820

Table 1-2 shows that the probability of generating a prediction with high agreement with the OpenCAT model increases remarkably when a reduced feature training is utilized in building an *individual regresssor* for each targeted PK parameter. The results in this table suggest that, in the case of the *MO regressor*, the longer the output vector the more accuracy will be lost, with the last components suffering the most. This behavior may be due to the fact that when building a *MO regressor* the machine assumes a relationship among the components of the labels that might not be true for all applications.

1.3.4 Regressor prediction validation

For the sake of validating the predictions of the trained regressors, the PK predictions of individual reduced features trained regressor -which possess the highest probability of generating predictions within the range of 20 % error- was compared with both experimental results from the literature and predictions from the OpenCAT model for a variety of drugs. To critically evaluate the predictions, drugs were chosen across all four classes of the bio-pharmaceutical classification scheme (Charalabidis, Sfouni, Bergström, & Macheras, 2019).The resulting values for the PK parameters are depicted in the Figure 1-11.

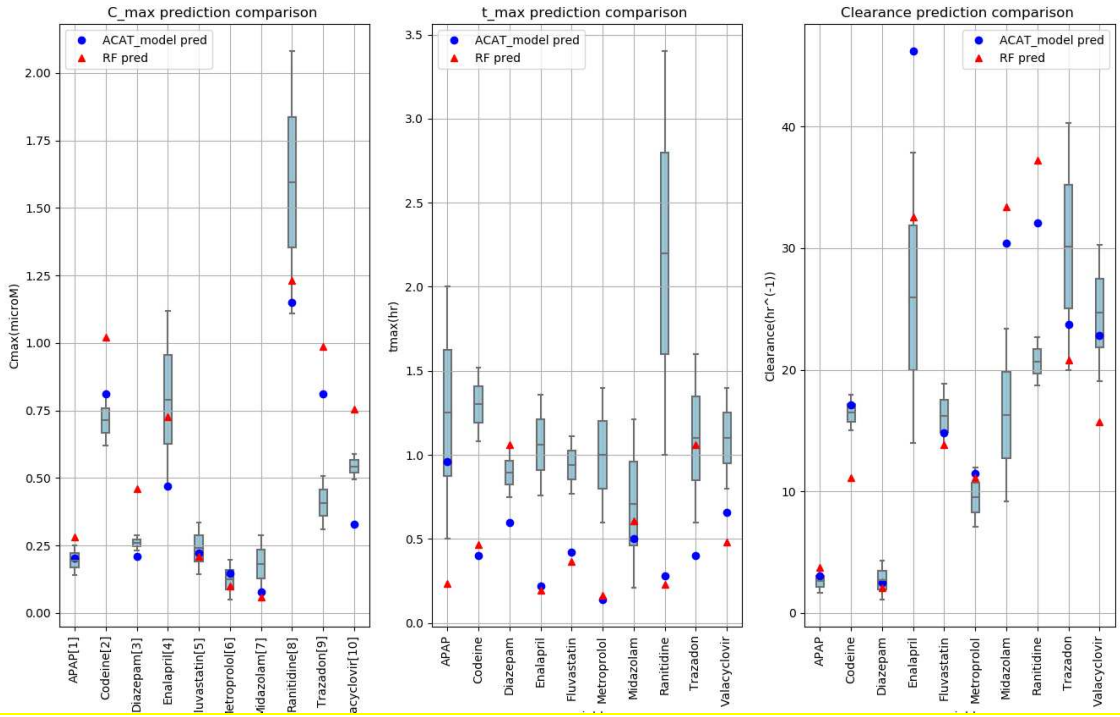


Figure 1-11 Experimental results vs ACAT and RF predictions

Figure 1-11 shows that our regressors generally predict PK parameters in close agreement to those for the full OpenCAT model. As expected, the agreement with respect to data {1. (Critchley, Critchley, Anderson, & Tomlinson, 2005) , 2. (Kim, et al., 2002) , 3. (Friedman, et al., 1992) , 4. (Arafat, et al., 2005) , 5. (Smith, Jokubaitis, Troendle, Hwang,, & Robinson, 1993) , 6. (Stout, et al., 2011) , 7. (Link, et al., 2008) , 8. (Hecken, Tjandramaga, Mullie, Verbesselt, & Schepper, 2015) , 9. (Kale & Agrawal, 2015) , 10. (Lin, Tian, Tian, Zhang, & Mao, 2011) } is similar to that of the full model. In certain cases, the OpenCAT model has relatively poor predictive capabilities for certain drug types.

1.4 Conclusion

One of the major aims of the work described in this chapter was to investigate the potential of machine learning algorithms to help address a significant problem related to

pharmacology and toxicology by creating an alternative method for PK parameter estimation. In particular, random forest regressors were trained based on a large data set generated through simulation conducted using the OpenCAT differential equation-based model. The training of these RF regressors made them capable of predicting PK parameters associated with a broad range of orally administered drugs. Although the predictions generated by RF regressor generally agree well with those derived from OpenCAT predictions, both set of predictions are occasionally far from the experimentally obtained PK parameters (Figure 1-11). Aside from the OpenCAT model, other oral absorption models have been developed (Ando, Hisaka, & Suzuki, 2015). It may be that a RF regressor could be developed to leverage the strengths of each such model to provide a good level of PK parameter prediction across disparate drug classes.

2 Chapter II: Machine Learning to inform the development of antidotes for nerve agent poisoning

2.1 Introduction

The first nerve agent was synthesized by a German scientist conducting industrial research for the development of pesticides (Tucker, 2007). These agents fall within the general class of organophosphorus compounds (OPs), which include insecticides such as chlorpyrifos and diazinon. OPs interfere with both the central CNS and the peripheral PNS branches of the nervous system by irreversibly inhibiting acetylcholinesterase the enzyme responsible for the breakdown of the neurotransmitter acetylcholine in the synaptic cleft (Costanzi, Machado, & Mitchell, 2018). The accumulation of the neurotransmitter caused by OP exposure can lead to an overstimulation of cholinergic neurotransmission accompanied by various overt adverse health effects. In the unfortunate event of exposure to a nerve agent, a person could experience irritating symptoms such as vomiting; involuntarily muscle twitching; paralysis; and death, primarily resulting from respiratory failure or seizures (King & Aaron, 2015).

Owing to the potent toxicity of this family of compounds, they have been used in numerous military applications as chemical warfare agents. In response, the military and some civilian organizations have sought antidotes to reduce the lethality of these compounds should anyone be exposed through warfare, terrorist attacks, or accidents. The conventional treatment regimen for a poisoning caused by a nerve agent is a combination of three drugs, each of which serves a specific purpose (Chambers J. E., et al., 2016). Specifically, an oxime (e.g., pralidoxime or 2-PAM) will be given to reactivate acetylcholinesterase (AChE), atropine will be injected to antagonize the

muscarinic acetylcholine receptors, and a benzodiazepine (e.g., diazepam) will be administered to alleviate agent-induced seizures.

Unfortunately, the current treatment regimen has a number of limitations. First, the administration of atropine will adequately antagonize the muscarinic acetylcholine receptors, but not the nicotinic receptors. Second, none of the currently known antidotes is a broad-spectrum antidote and all of them lack the ability to reactivate the inhibited acetylcholinesterase once it ages (Moshiri, Darchini-Maragheh, & Balali-Mood, 2012). Finally, the FDA-approved oximes are deficient with respect to their ability to cross the blood–brain barrier (BBB) and thus poorly reactivate AChE in the central nervous system (CNS) (Chambers J. E., et al., 2016).

To improve this critical therapeutic element, a series of oximes candidates were designed, synthesized, and tested by colleagues at Mississippi State University (Chambers et al.). Due to the enhanced lipophilicity of these candidates, they have a better ability to cross the BBB and reactivate inhibited brain cholinesterase (ChE) (Chambers, Chambers, Meek, & Pringle, 2013). Since the hazard associated with nerve agents is significant, only a handful of laboratories can test these compounds. To overcome this issue, Chambers et al. synthesized surrogates for the nerve agents sarin and VX to investigate the ChE reactivation potential of the oximes candidates both in vitro and in vivo (Meek, et al., 2012). The oxime candidates (substituted phenoxyalkyl pyridinium oximes) were designed with the objective of maximizing BBB penetration and ChE reactivation. Their set of candidate chemicals was developed based on the investigators' knowledge of the relevant chemistry and biology, with choices being limited by the feasibility of synthesis of the desired molecules.

Though the domain-specific knowledge of Chambers et al. was critical in establishing a set of candidate molecules, there was no attempt to systematically utilize their experimental results to

make predictions outside of those for the dataset nor to use the information to design new oximes or assess the reactivation potential of existing oximes against untested nerve agents or OP insecticides.

The aim of the work described in this chapter was to bridge this knowledge gap through the development and use of machine learning techniques and to provide a tool that could be used for the rational design and analysis of candidate OP antidotes.

2.2 Methodology

The overall workflow for the model building and analysis is shown in Figure 2-1 It comprised the following essential steps: (i) Converting chemical structure into SMILES, (ii) Converting SMILES into their respective descriptors, (iii) Creating a dataset to fulfill the required goal, and (iv) Building the ML model and initiating the training, testing, and verification procedure. Each of these steps will be detailed in the following sections.

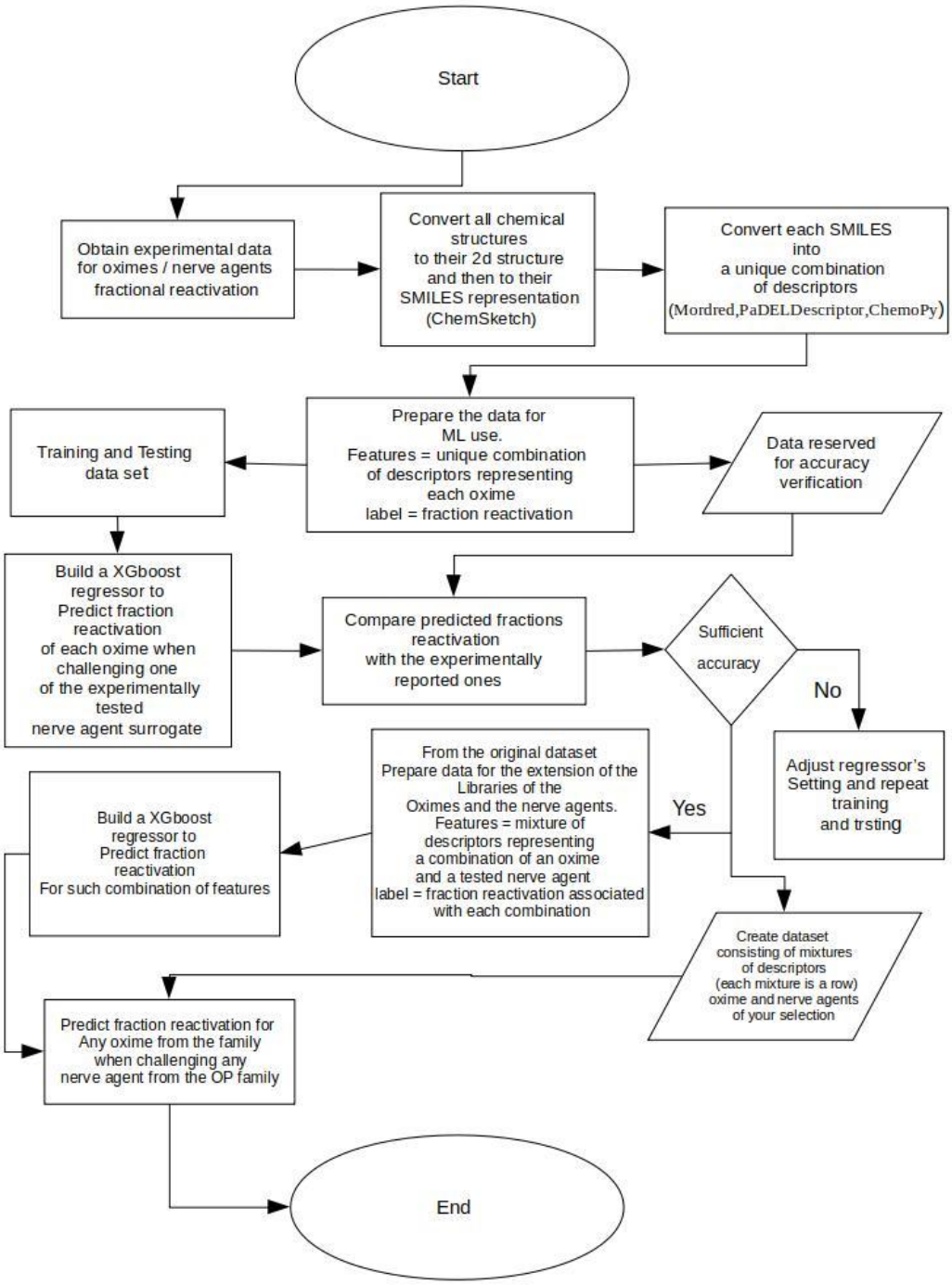


Figure 2-1 Flowchart detailing the construction of the ML model to predict antidotes' reactivation fraction

2.2.1 Data gathering

The data used in the present study were obtained from Chambers et al. These data comprised *in vitro* reactivation values corresponding to each oxime and surrogate nerve agent or OP combination tested.

2.2.2 Determining the model features

The features for use in the model were molecular descriptors of both the oxime antidote and the nerve agent/OP. These features included i. constitutional descriptors; ii. 1D-descriptors (i.e. list of structural fragments, fingerprints); iii. 2D-descriptors (i.e. graph invariants); and iv. 3D-descriptors (e.g., quantum-chemical descriptors).

First, to convert the data to a form suitable for input to descriptor calculators, the chemical structures from Chambers et al. were converted to simplified molecular-input line-entry system notation (SMILES) (SMILES - A Simplified Chemical Language, 2020) which is a specification of a molecular structure using ASCII strings. These SMILES representations were confirmed through the use of the molecular structure drawing program, ACD/ChemSketch (ACD) .

Next, the SMILES strings were used as input to several different open-source or freely-available descriptor calculation codes, namely Mordred (Moriwaki, Tian, Kawashita, & Takagi, 2018) (>1800 descriptors), PaDEL-Descriptor (PaDEL-Descriptor, 2020) (1875 descriptors), and ChemoPy (DS, QS, QN, & YZ, 2013)(1135 descriptors). Owing to differences in nomenclature and descriptor definitions among these software packages, the total unique number of descriptors was infeasible to determine. Each set of descriptors was used separately in the workflow and the set that contained the best predictors for the outputs of interest was selected.

2.2.3 Model building and verification

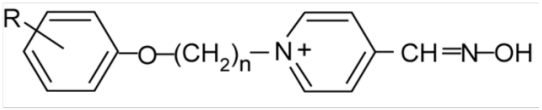
See Appendix I for details about the background and approach to model building.

In essence, using the data from Chambers et al.), a XGboost model - which is another tree based model that train trees in sequence with every tree correcting the mistake made by the previous tree- (A Gentle Introduction to XGBoost for Applied Machine Learning, 2020) was developed to predict the enzyme reactivation (label) based on a set of molecular descriptors (features).

As described in Chapter I, a feature importance analysis was conducted to identify the features with the most influence on the label. From these primary features, a reduced feature set regression model was developed.

2.2.4 Developing and characterizing new candidate antidotes

The oximes designed by Chambers et al., were all of a similar chemical structure:



,

However, the set of chemicals was constructed by varying certain structural features, namely the R group, location of this group in the first aromatic ring, and linker length, n. The specific parameters (see Appendix II for details) were limited to those leading to molecules that could be readily synthesized. Owing to resource constraints, Chambers et al. were not able to synthesize and test all variations of the molecular structures based on these specific parameter sets. However, by examining the set of structures synthesized versus those that are possible, a list of 'missing'

oximes was created. That is, compounds that were feasible, but were not prioritized by Chambers et al. for synthesis and characterization.

2.3 Results and Discussion

2.3.1 Reduced feature training

The set of reduced features used in this analysis consists of features produces by the Mordred package (Moriwaki, Tian, Kawashita, & Takagi, 2018).

Figure 2-2 shows the MSE associated with the reduced features training for four nerve agent surrogates. The endpoint of interest (or regression label) was the reactivation fraction when tested in an *in vitro* rat tissue system. The process of developing the optimal reduced feature set was as follows: The first reduced set was created containing only the most influential feature. For the second set, the two most influential features were incorporated. Subsequent sets included one more feature at the time to form a new set until all of the features had been added. Figure 2-2 shows the MSE associated with variations in the reduced features set. The lowest trough of the curves corresponds to the reduced feature set showing the best predictive ability versus the training data.

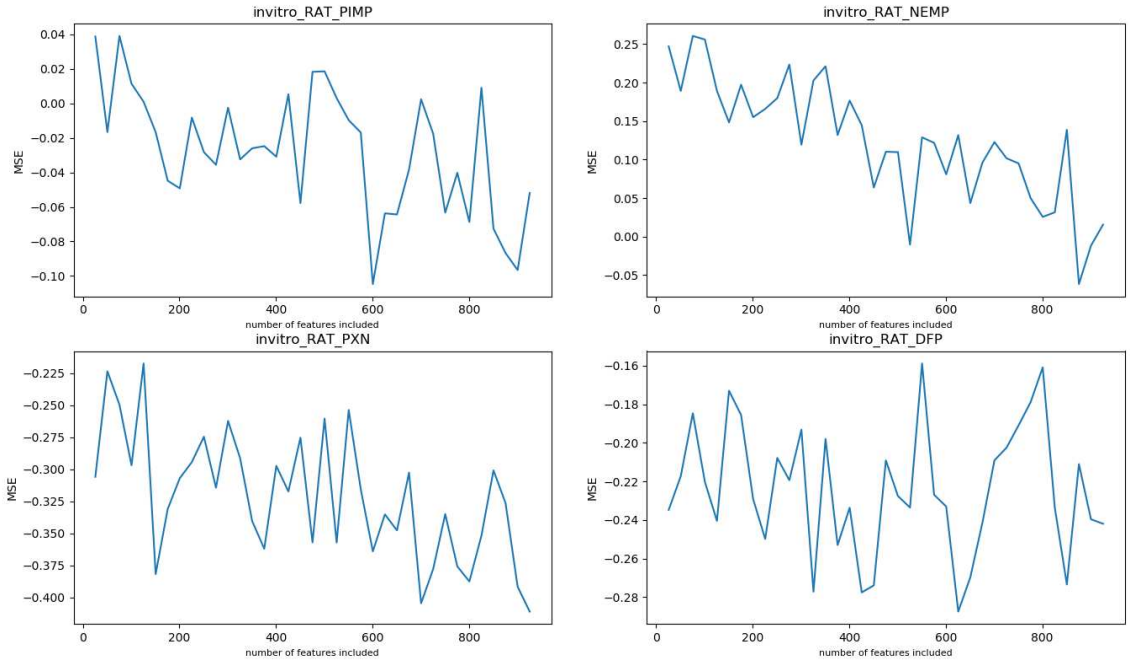


Figure 2-2 Mean squared error associated with various reduced features sets

Figure 2-3 shows the importance associated with each individual feature. The importance of feature according to this figure will eventually reach zero which indicates no contribution from this feature is necessary for the internal decision-making process of the regressor.

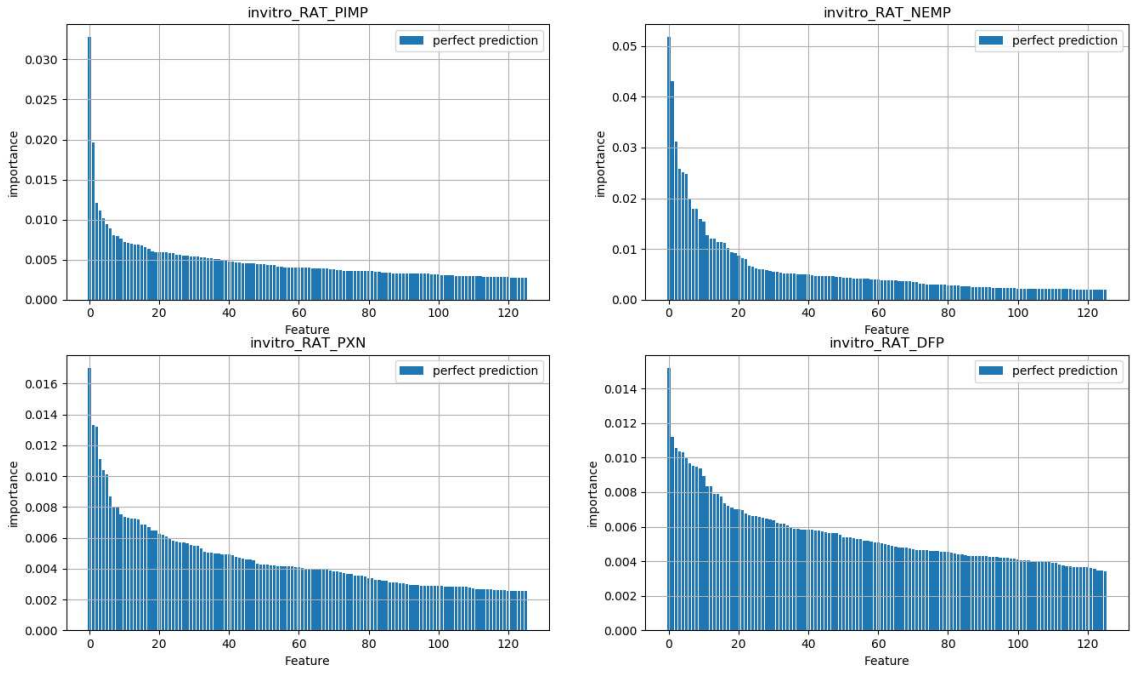


Figure 2-3 Feature (descriptor) importance

After the reduced feature set was found, the regressor was tested against the full (training plus test) data to determine the predictive capability of the regressors. Figure 2-4 shows the result of this comparison, where dots represent the error percentage corresponds to each antidote/surrogate combination and the two solids lines represent a prediction envelope of ±10 %. This figure shows that when using these optimized trained regressors, most of the predictions fall within the envelope.

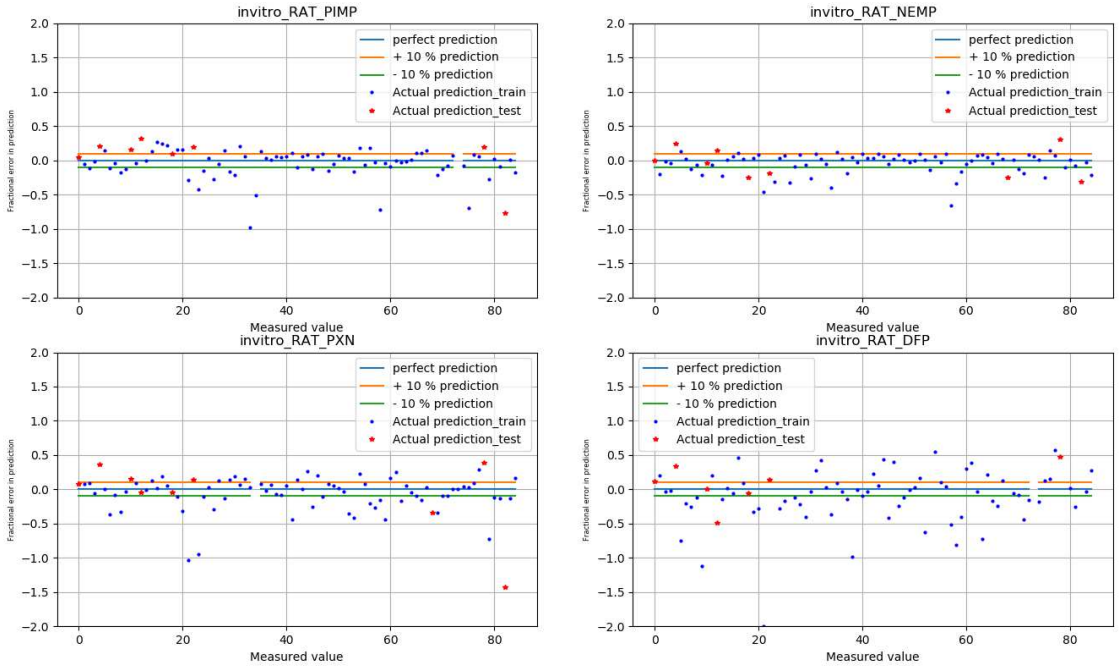


Figure 2-4 Prediction error for reactivation based on various surrogate challenges

The next step in the verification was to compare regressor predictions against data for which no similar training data had been used. In other words, the comparison Table 2-1 was based completely on samples for which the regressors had no prior knowledge. Table 2-1 shows the results of this comparison for all of the nerve agent surrogates (PIMP, NEMP, DFP, and PXN). In general, the predictions are in reasonable agreement with the data. In practice, such predictions are useful to Chambers et al. when they are expected to be within about 25% of the experimentally determined value.

Table 2-1 Measured (M) versus predicted (P) fraction reactivation for tested surrogates

Id	n	R_group	PIMP (M | P)	NEMP (M | P)	DFP (M | P)	PXN (M | P)
MSU 002	3	4-02N-	0.37 | 0.366	Training set	0.69 | 0.507	0.28 | 0.105
MSU 013	3	3-CH=CHCH=CH-4	0.65 | 0.596	0.57 | 0.578	0.45 | 0.633	Training set
MSU 033	3	3,4-Cl2-	0.55 | 0.651	0.62 | 0.657	Training set	0.4 | 0.246
MSU 036	4	2,6-([CH3]2CH)2-	0.53 | 0.621	0.58 | 0.594	Training set	0.23 | 0.299
MSU 066	5	4-Ph-C(CH3)2-	0.48 | 0.453	Training set	0.44 | 0.517	0.15 | 0.117
MSU 073	5	4-CH3CH2C(CH3)2-	0.34 | 0.311	Training set	0.42 | 0.393	0.13 | 0.105

2.3.2 Potential new antidotes

After verifying the functionality of the optimized regressor, a new study commenced to look predicting the reactivation of the 'missing' oximes described earlier (section 2.2.4). Appendix II contains structural information for the list of tested oximes (denoted by 'MSU 0XX') and the 'missing' oximes (denoted by 'CSU 0XX'). Following the procedure outlined earlier, the descriptors for these candidate oximes were computed and used as features in the regressor to predict the label fraction reactivation. Table 2-2 shows the regressor predictions for several

candidate and tested oximes against two surrogate nerve agents: PIMP and NEMP. These results suggest that some of the untested antidotes would outperform many of the tested ones.

Table 2-2 Predicted reactivation fraction for promising oxime compounds

PIMP				NEMP			
Id	n	R_group	Reactivation	Id	n	R_group	Reactivation
MSU 016	3	4-Ph-C(:0)-	0.78	MSU 081	3	2-CH=CHCH=CH-3	0.80
MSU 021	4	2,5-Cl2-	0.72	MSU 021	4	2,5-Cl2-	0.76
CSU_023	5	4-O2N-	0.331	CSU_004	3	2,5-Cl2-	0.622
MSU 009	5	4-CH3-0	0.30	MSU_044	3	4-Br-	0.62
MSU_031	4	3-0-C(:0)-CH=C(CH3)-4	0.27	MSU_037	4	3-CH3-4-Cl-	0.61

2.3.3 Expanding the library of nerve agents and insecticides

One of the issues with nerve agent antidotes is that they may not be effective against exposure to agents other than those one for which they were designed. Thus, the ability to evaluate the enzyme reactivation for oximes against a range of agents and other OPs is critically important. To fill this need, a regressor design approach was developed to create a tool to make such predictions. A key element of this approach was to include the full set of descriptors for both the oxime and the OP as features and to be able to assess the contribution of descriptors from both molecule types.

Like the methods employed previously, the regressor was trained with a dataset separate from that used for verification. Again, the agreement in reactivation fraction between the predictions and data was assessed. This comparison is shown in Figure 2-5, where points represent the error percentage corresponds to each sample and the two solids lines represent a error envelope of ±10%.

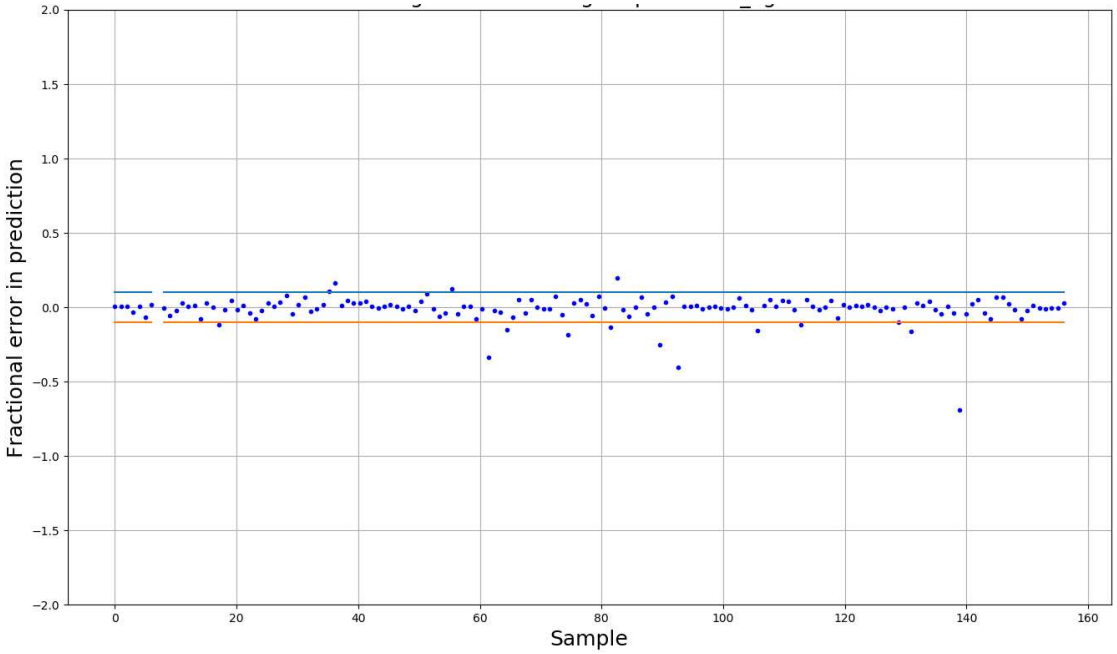


Figure 2-5 Prediction error for the extended library

The histograms shown in Figure 2-6 was generated for the purpose of quantifying the accuracy of our regressor. The plot indicates that the probability of getting a predicted value for a reactivation fraction associated with antidote-nerve agent combination with an error percentage of ± 10% is 0.88 with extremely low probability of obtaining predictions with error percentage greater than 50%.

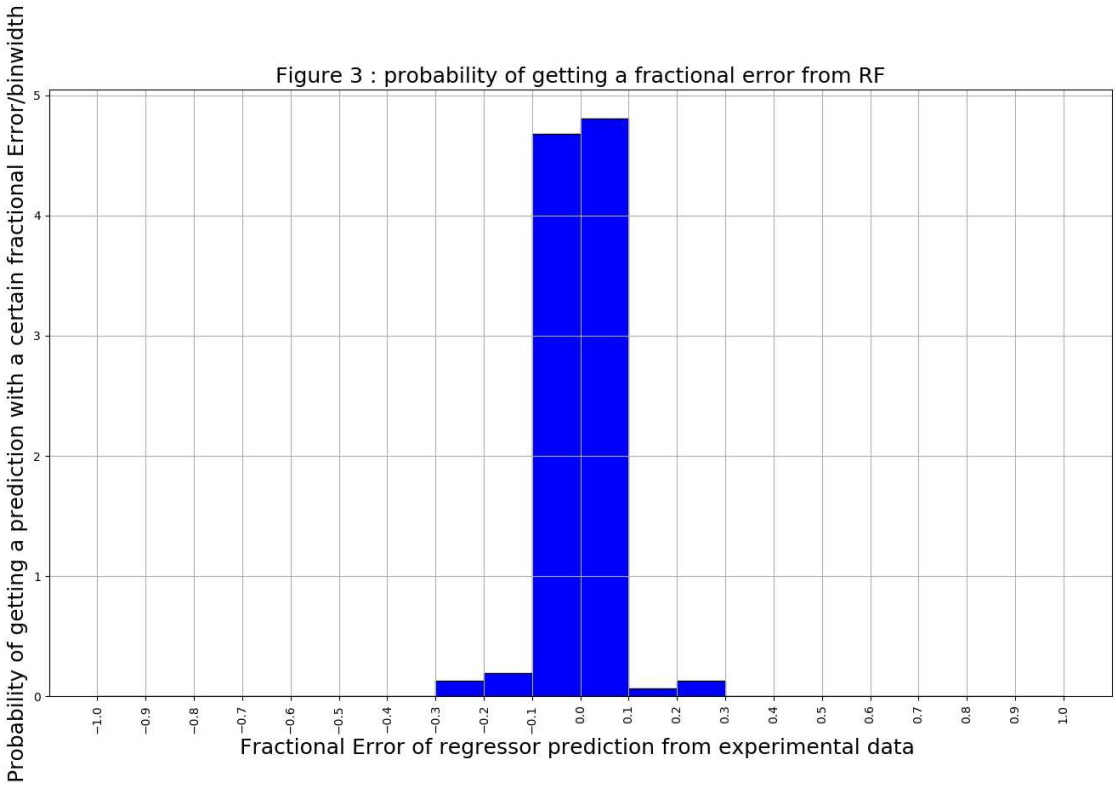


Figure 2-6 Histogram view of the prediction error for the extended library

After assessing the accuracy of the regressor, the library of OPs was expanded by adding seven new compounds to the original library. The chemicals in the updated library were broken into three groups:

i. nerve agent surrogates: nitrophenyl isopropyl methylphosphonate (NIMP), 4-nitrophenyl ethyl dimethylphosphoramidate (NEDPA), and phorate oxon.
ii. insecticides: diazoxon and chlorpyrifos oxon.
iii. nerve agents: [2-(Diisopropylamino) ethyl]-*O*-ethyl methylphosphonothioate ethyl (a.k.a VX) and (*RS*)-Propan-2-yl methylphosphonofluoridate (a.k.a. sarin).

Based on this updated library, predictions of enzyme reactivation were made using the regressor Table 2-3 .

Table 2-3 Predicted fraction reactivation for some untested agents and OPs

Id	surrogates			insecticides		nerve agent	
	Phorate_oxon	NIMP	NEDPA	Diazoxon	Chlorpyrifos_oxon	Sarin	VX
MSU_001	0.526	0.5097	0.6138	0.4324	0.5221	0.2598	0.3559
MSU_002	0.312	0.3222	0.4457	0.2735	0.319	0.1422	0.1473
MSU_007	0.404	0.3802	0.4455	0.3037	0.4115	0.2304	0.2254
MSU_014	0.452	0.4618	0.6064	0.4486	0.4795	0.2152	0.2877
MSU_045	0.592	0.6724	0.6846	0.4792	0.5569	0.3647	0.4624
MSU_052	0.381	0.4795	0.5085	0.3653	0.392	0.152	0.2669
MSU_056	0.452	0.5329	0.5128	0.3169	0.4568	0.2272	0.3348

This table was shared with Chambers et al. who then generated a set of verification reactivation data for the surrogate phorate oxon. As shown in Table 2-4 the predictions are in reasonable agreement with the data for most of the oximes. According to Chambers et al., because this type of tool would likely be used, not as a quantitative predictor, but as a screening tool, this level of accuracy is probably sufficient.

Table 2-4 Comparison of the fraction reactivation for Phorate_oxon

Identifier	Fractional reactivation (measured | predicted)
MSU_001	0.579 | 0.526
MSU_002	0.479 | 0.312
MSU_007	0.446 | 0.404
MSU_014	0.435 | 0.452
MSU_045	0.502 | 0.592
MSU_052	0.62 | 0.381
MSU_056	0.249 | 0.452

2.4 Conclusion

One of the major aims of the work described in this chapter was to investigate the potential of machine learning algorithms to help address a significant problem related to pharmacology and toxicology. Specifically, a major objective was to develop and assess a machine learning tool to help in the design of phenoxyalkyl pyridinium oximes (Chambers & Meek, 2020) as nerve agent antidotes. Though no mechanistic insights were gained by examining the most influential features for the XGboost regressor, the tool was still useful in making predictions in two important cases: (i) new oximes that could be used to reactivate AChE following exposure to OPs and (ii) assessing the reactivation potential for a given oxime against new (untested) OP challenges. It is anticipated that a tool such as this will be useful in the future design of antidote compounds to treat nerve agent and OP insecticide poisoning.

Bibliography

(n.d.). Retrieved from ACD/ChemSketch for Academic and Personal Use

(2020, 03 29). Retrieved from GNU MCsim: https://www.gnu.org/software/mcsim/

A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning. (2020, 5 12). Retrieved from Machine learning Mastery: A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning

A Gentle Introduction to XGBoost for Applied Machine Learning. (2020, 04 30). Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/gentle-introduction-xgboost-applied-machine-learning/

ACD/ChemSketch for Academic and Personal Use. (2020, 01 17). Retrieved from https://www.acdlabs.com/resources/freeware/chemsketch/

Agoram, B. W. (2001). Predicting the impact of physiological and biochemical processes on oral drug bioavailability. *Advanced Drug Delivery Reviews*, 50.

Amrane, M., Oukid, S., Gagaoua, I., & Ensari, T. (2018). Breast Cancer Classification using Machine learning. *IEEE Xplore*.

An, S., Malhotra, K., Dilley, C., Han-Burgess, E., Valdez, J. N., Robertson, J., & Sun, J. (2018). Predicting drug-resistant epilepsy — A machine learning approach based on administrative claims data. *Epilepsy Behavior*, 118-125.

Ando, H., Hisaka, A., & Suzuki, H. (2015). A New Physiologically Based Pharmacokinetic Model for the Prediction of Gastrointestinal Drug Absorption: Translocation Model. *Drug Metabolism and Disposition*, 590-602.

Arafat, T., Awad, R., Hamad, M., Azzam, R., Al-Nasan, A., Jehanli, A., & Matalka, K. (2005). Pharmacokinetics and pharmacodynamics profiles of enalapril maleate in healthy volunteers following determination of enalapril and enalaprilat by two specific enzyme immunoassays. . *Journal of Clinical Pharmacy and Therapeutics*, 319-328.

Bois, F. (2018). ACAT model documentation. *INERIS Technical Report*.

Chambers, J. E., & Meek, E. C. (2020). Novel centrally active oxime reactivators of acetylcholinesterase inhibited by surrogates of sarin and VX. *Neurobiology of Disease*, 133.

Chambers, J. E., Chambers, H. W., Meek, E. C., & Pringle, R. B. (2013). Testing of novel brain-penetrating oxime reactivators of acetylcholinesterase inhibited by nerve agent surrogates. *Chemico-Biological Interactions*, 135–138.

Chambers, J. E., Meek, E. C., Bennett, J. P., Bennett, W., Chambers, H. W., Leach, C. A., & Wills, R. W. (2016). Novel substituted phenoxyalkyl pyridinium oximes enhance survival and attenuate seizure-like behavior of rats receiving lethal levels of nerve agent surrogates. *Toxicology*, 51–57.

Charalabidis, A., Sfouni, M., Bergström, C., & Macheras, P. (2019). The Biopharmaceutics Classification System (BCS) and the Biopharmaceutics Drug Disposition Classification System (BDDCS): Beyond guidelines. *International Journal of Pharmaceutics*, 264–281.

Costanzi, S., Machado, J.-H., & Mitchell, M. (2018). Nerve Agents: What They Are, How They Work, How to Counter Them. *ACS Chemical Neuroscience*, 873–885.

Critchley, J. A., Critchley, L. A., Anderson, P. J., & Tomlinson, B. (2005). Differences in the single-oral-dose pharmacokinetics and urinary excretion of paracetamol and its conjugates between Hong Kong Chinese and Caucasian subjects. *Journal of Clinical Pharmacy and Therapeutics*, 179-184.

Djuris, J. (2013). *Computer aided applications in pharmaceutical technology.*

DS, C., QS, X., QN, H., & YZ, L. (2013). ChemoPy: freely available python package for computational biology and chemoinformatics. *Bioinformatics.*

Friedman, H., Greenblatt, D. J., Peters, G. R., Metzler, C. M., Charlton, M. D., Harmatz, J. S., & Francom, S. F. (1992). Pharmacokinetics and pharmacodynamics of oral diazepam: Effect of dose, plasma concentration, and time. *e. Clinical Pharmacology and Therapeutics*, 139-150.

Gobeau, N., Stringer, R., B. S., Tuntland, T., & Faller, B. (2016). Evaluation of the GastroPlusTM Advanced Compartmental and Transit (ACAT) Model in Early Discovery. *Pharmaceutical Research*, 2126-2139.

Hartmanshenn, C., Scherholz, M., & Androulakis, I. P. (2016). Physiologically based pharmacokinetic models: Approaches for enabling personalized medicine. *Journal of Pharmacokinetics and Pharmacodynamics*, 481-504.

Hecken, A., Tjandramaga, T., Mullie, A., Verbesselt, R., & Schepper, P. (2015). Ranitidine: Single dose pharmacokinetics and absolute bioavailability in man. *British Journal of Clinical Pharmacology* , 195-200.

Kale, P., & Agrawal, Y. K. (2015). Pharmacokinetics of single oral dose trazodone:A randomized, two-period, cross-over trial in healthy, adult, human volunteers under fed condition. *Frontiers in Pharmacology*.

Kim, I., Barnes, A. J., Oyler, J. M., Schepers, R., Joseph, R. E., Cone, E. J., & Huestis, M. A. (2002). Plasma and Oral Fluid Pharmacokinetics and Pharmacodynamics after Oral Codeine Administration. *Clinical Chemistry* .

King, A. M., & Aaron, C. K. (2015). Organophosphate and Carbamate Poisoning. *Emergency Medicine Clinics of North America*, 133-151.

Komura, D., & Ishikawa, S. (2018). Machine Learning Methods for Histopathoogical Image Analysis. *Computational and Structural Biotechnology Journal,* , 34-42.

Lalka, D., Griffith, R. K., & Cronenberger, C. L. (1993). The Hepatic First-Pass Metabolism of Problematic Drugs. *The Journal of Clinical Pharmacology*, 657-669.

Lin, H., Tian, Y., Tian, J., Zhang, Z., & Mao, G. (2011). Pharmacokinetics and bioequivalence study of valacyclovir hydrochloride capsules after single dose administration in healthy Chinese male volunteers. *Arzneimittelforschung*, 162-167.

Link, B., Haschke, M., Grignaschi, N., Bodmer, M., Aschmann, Y. Z., Wenk, M., & Krähenbühl, S. (2008). Pharmacokinetics of intravenous and oral midazolam in plasma and saliva in humans: Usefulness of saliva as matrix for CYP3A phenotyping. . *British Journal of Clinical Pharmacology*, , 473-484.

Machine Learning For Beginners. (2020, 05 02). Retrieved from Toward data science: Machine Learning For Beginners

Meek, E. C., Chambers, H. W., Coban, A., Funck, K. E., Pringle, R. B., Ross, M. K., & Chambers, J. E. (2012). Synthesis and In Vitro and In Vivo Inhibition Potencies of Highly Relevant Nerve Agent Surrogates. *Toxicological Sciences*, 525–533.

Moriwaki, H., Tian, Y.-S., Kawashita, N., & Takagi, T. (2018). Mordred: a molecular descriptor calculator. *Journal of Cheminformatics*.

Moshiri, M., Darchini-Maragheh, E., & Balali-Mood, M. (2012). Advances in toxicology and medical treatment of chemical warfare nerveagents. *Daru*.

PaDEL-Descriptor. (2020, 04 30). Retrieved from http://www.yapcwsoft.com/dd/padeldescriptor/

Python. (2020, 5 12). Retrieved from Python 3.7.0: https://www.python.org/downloads/release/python-370/

Random Forest Regression. (2020, 04 29). Retrieved from toward data science: https://towardsdatascience.com/random-forest-and-its-implementation-71824ced454f

Sachan, N., Bhattacharya, A., Pushkar, S., & Mishra, A. (2009). Biopharmaceutical classification system: A strategic tool for oral drug delivery technology. *Asian Journal of Pharmaceutics*, 76.

sci-kit learn machine learning in python. (2020, 5 12). Retrieved from scikit learn: https://scikit-learn.org/stable/

SMILES - A Simplified Chemical Language. (2020, 04 30). Retrieved from DAYLIGHT: https://www.daylight.com/dayhtml/doc/theory/theory.smiles.html

Smith, H. T., Jokubaitis, L. A., Troendle, A. J., H. D., & Robinson, W. T. (1993). Pharmacokinetics of Fluvastatin and Specific Drug Interactions. . *American Journal of Hypertension*.

Stout, S. M., Nielsen, J., Welage, L. S., Shea, M., Brook, R., Kerber, K., & Bleske, B. E. (2011). . Influence of Metoprolol Dosage Release Formulation on the Pharmacokinetic Drug Interaction With Paroxetine. *The Journal of Clinical Pharmacology*, 389-396.

Suzuki, T., Morita, H., Ono, K., Maekawa, K., Nagai, R., & Yazaki, Y. (1995). Sarin poisoning in Tokyo subway. *Lancet*, 980.

Tucker, J. B. (2007). *War of nerves: chemical warfare from World WarI to Al-Qaeda.* Anchor, New York.

Wolpert, D., & Macready, W. (1997). No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 67-82.

Yu, L. X., & Amidon, G. L. (1999). A compartmental absorption and transit model. *International Journal of Pharmaceutics*, 119-125.

Appendices

Appendix I: Relevant details of machine learning approach

Introduction

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence (AI) based on the idea that systems can 'learn' from data, identify patterns, and make decisions with minimal human intervention. Machine learning can be used to solve regression, classification, and clustering problems (Machine Learning For Beginners, 2020) In the case of the OpenCAT analysis, predicting pharmacokinetics parameters is a regression problem since the outputs are continuous. Various machine learning techniques are known to be useful when confronted with such problem. For such problems, a variety of techniques are available, including decision trees, random forests regressions, and neural networks.

Each technique possess its unique way of approaching the final decision and which plays the most important role in one technique outperforming the others when dealing with a certain problem However, the ultimate goal of all techniques is to build an understanding of the relationships among variables and their corresponding outputs and determine the levels of influence among variables and their effects on reaching a certain decision. The "No Free Lunch" theorem (Wolpert & Macready, 1997), which basically states that "no one machine learning technique is the best for all problems", necessitates and exploration of various techniques for each problem of interest. Moreover, even for the same problem of interest, the performance of a given machine learning technique is altered based on the size and the structure of the data set.

As the name implies, a random forest (RF) is a collection of decision trees working in concert. One can think of a decision tree as a series of yes/no questions and subsequent branches asked about the data that eventually lead to a predicted class (or continuous value in the case of

regression). In the instance of the random forest regression, the decisions made by different trees will be averaged in an appropriate way, as oppose to a 'voting' procedure that takes place when a classification problem is to be solved. Two concepts distinguish random forest from a simple collection of decision trees. The first concept is the random sampling of training dataset when building a tree and the second is the random subsets of features considered prior to nodes' splitting. Random forests can used in cases of complex, non-linear relationships, and in contrast to a neural network, a RF is relatively transparent in terms of the decisions made to arrive at a decision. Despite this advantage, one must take care to avoid model over-fitting and/or under-fitting (OF/UF). In this work, Python (v3.7.0) (Python, 2020)was used to help to automate the workflow. The Python package scikit-learn(v 0.22.2) (sci-kit learn machine learning in python, 2020) was used to conduct the RF analyses.

An alternative to RF models is gradient boosting (GB) (A Gentle Introduction to the Gradient Boosting Algorithm for Machine Learning, 2020) The key difference between these approaches is that GB trains models in succession, with each new model being trained to correct the errors made by the previous ones. Models are added sequentially until no further improvements can be achieved. Another crucial distinction between RF and GB is the ability of the GB to handle data sets that contain missing data. In this work, XGBoost (A Gentle Introduction to XGBoost for Applied Machine Learning, 2020) was used to perform the GB calculations.

Model building

Optimizing a regressors' hyperparameters

Much of machine learning focused on balancing computational expense against prediction accuracy. To achieve this balance, several model parameters can be adjusted and optimized.

Optimization of so-called hyperparameters is essential in many instances, especially when dealing with large data sets or data set that involves a large number of features. A hyperparameter is a parameter whose value is set before the learning process begins. In the present analyses, the scikit-learn method RandomizedSearchCV was used to automatically try different combination of regressor hyperparameters to optimize the accuracy of the model. Using this method as core functionality, a Python script was developed to facilitate an efficient workflow. The first part of the script split the data set into training and testing subsets. To eliminate any bias in the training-testing split process, the code created multiple training-testing sets, keeping the size of the testing set fixed among all permutations. Next the randomized search method was employed to determine the optimal hyperparameters for a given permutation. Finally, each fitted regressor was tested by calculating the mean squared error of its predictions versus the data and the regressor was saved for future utilization.

Optimizing the size of the testing set

The size of the training and testing data sets plays an important role in determining the quality of a regressor and whether the regressor has been over-fitted or under-fitted. To assess the effects of dataset size, another Python script was built to vary size of the testing set as a fraction of size of the original data set. Smaller training data sets could be important if the optimization is computationally expensive. The second purpose of this step was to detect and eliminate regressor OF/UF. The script was designed to try various fractional sizes and for each size, optimize hyperparameters, train and test the regressor, and compute the mean squared errors. By examining the difference between the MSE related to the training set and that associated with the test set, the extent of OF/UF could be assessed.

Characterizing feature importance

In this step the importance of each feature in influencing the decision that to be made by each regressor was determined. Aside from helping to establish models with tractable numbers of features, the feature importance analysis has the potential to assist in the design of experiments. The scikit-learn method "feature_importance" was used as the core element of this analysis. Again, a Python script was created to iterate through training-testing splits, optimize hyperparameters, and fit and test regressors. In addition, a metric associated with the influence of each feature on the regression decision was computed for each permutation.

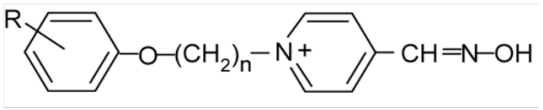
Assessing model performance

A crucial step in the development process for each regressor was an assessment of its prediction accuracy for a label's value compared to a specified subset of the available data. As noted earlier, the MSE was used as a metric; however, the coefficient of determination was used in other cases, depending on the purpose of the trial.

In order to visualize the accuracy of the regressor two types of plots were generated. The first type shows the relationship between the actual data and the predicted values. Dashed lines were often added as the boundaries of the prediction envelope. The second illustrative plot type was a histogram that shows the probability of obtaining values from the regressor within a certain percentage of the data.

Appendix II: Details of candidate antidotes

Generic oxime 2D structure:



Identifier	Linker length	R_group	Identifier	Linker length	R group	Identifier	Linker length	R group
CSU_001	3	4-Cl-	MSU 001	4	4-Cl-	MSU 044	3	4-Br-
CSU_002	3	4-CH3CH2C(:0)-	MSU 002	3	4-02N-	MSU 045	4	4-Br-
CSU_003	3	4-CH3-0-	MSU 003	4	4-CH3C(:0)-	MSU 046	5	4-Br-
CSU_004	3	2,5-Cl2-	MSU 004	5	4-CH3C(:0)-	MSU 047	5	2,3,5-(CH3)3-
CSU_005	3	4-Ph	MSU 005	4	4-CH3-0	MSU 048	5	3-CH=CHCH=CH-4
CSU_006	3	4-CH3-	MSU 006	4	3-CH=CHCH=CH-4	MSU 049	4	4-Ph-0-
CSU_007	3	3-Cl-	MSU 007	3	H-	MSU 050	4	4-Ph-CH2-
CSU_008	3	2-CH3-4-02N-	MSU 008	4	H-	MSU 051	4	4-Ph-CH2C(:0)-

CSU_009	3	2,6-([CH3]2CH)2-	MSU009	5	4-CH3-0	MSU052	3	2,3,5-(CH3)3-
CSU_010	3	4-Ph-0-	MSU010	5	4-Cl-	MSU053	3	4-Ph-CH2-O-
CSU_011	3	4-Ph-CH2-	MSU011	6	4-CH3C(:0)-	MSU054	5	4-Ph-CH2-O-
CSU_012	3	4-(CH3)3CCH2C(CH3)2-	MSU012	4	4-CH3CH2C(:0)-	MSU055	5	4-(CH3)3CCH2C(CH3)2-
CSU_013	3	2-CH3-4-(CH3)3C-	MSU013	3	3-CH=CHCH=CH-4	MSU056	4	2,4,5-Cl3)-
CSU_014	3	2,4-[(CH3)3C-]2-	MSU014	3	4-CH3C(:0)-	MSU057	4	2,3,5-(CH3)3-
CSU_015	3	4-CH3(CH2)6-0-	MSU015	5	4-CH3CH2C(:0)-	MSU058	4	2-CH3-4-(CH3)3C-
CSU_016	3	4-Br-3,5-(CH3)2-	MSU016	3	4-Ph-C(:0)-	MSU059	4	2,4-[(CH3)3C-]2-
CSU_017	3	4-Ph-C(CH3)2-	MSU017	4	4-Ph-C(:0)-	MSU060	4	4-CH3CH2C(CH3)2-
CSU_018	3	2-Br-4-Cl-	MSU018	5	4-Ph-C(:0)-	MSU061	4	4-CH3(CH2)6-0-
CSU_019	3	3-02N-4-Cl-	MSU019	6	4-CH3-0-	MSU062	4	4-(CH3)3CCH2C(CH3)2-

CSU_020	3	2-Ph-CH2-	MSU_020	4	4-Ph-CH2-O-	MSU_063	4	4-Br-3,5-(CH3)2-
CSU_021	3	2,6-Br2-4-CH3-	MSU_021	4	2,5-Cl2-	MSU_064	3	4-CH3CH2C(CH3)2-
CSU_022	4	4-CH3-0-	MSU_022	6	3-CH=CHCH=CH-4	MSU_065	5	2,4,5-Cl3)-
CSU_023	5	4-02N-	MSU_023	4	4-02N-	MSU_066	5	4-Ph-C(CH3)2-
CSU_024	5	H-	MSU_024	4	4-Ph	MSU_067	4	4-Ph-C(CH3)2-
CSU_025	5	4-CH3-0-	MSU_025	4	2-CH=CHCH=CH-3	MSU_069	4	2-Br-4-Cl-
CSU_026	5	4-Ph	MSU_026	4	4-CH3-	MSU_070	3	2,4,5-Cl3)-
CSU_027	5	4-CH3-	MSU_027	6	4-Ph	MSU_071	4	2,4-Cl2-
CSU_028	5	3-Cl-	MSU_028	4	3-Cl-	MSU_072	5	4-Ph-CH2-
CSU_029	5	3,4-Cl2-	MSU_029	4	3,4-Cl2-	MSU_073	5	4-CH3CH2C(CH3)2-
CSU_030	5	2,4,6-Cl3-	MSU_030	3	2,4,6-Cl3-	MSU_074	5	2,5-Cl2-

CSU_031	5	2,6-([CH3]2CH)2-	MSU 031	4	3-0-C(:0)-CH=C(CH3)-4	MSU 075	5	4-Ph-CH2C(:0)-
CSU_032	5	4-Cl-3,5-(CH3)2-	MSU 032	5	3-0-C(:0)-CH=C(CH3)-4	MSU 076	4	4-(CH3)3C-
CSU_033	5	4-Ph-0-	MSU 033	3	3,4-Cl2-	MSU 077	4	3-02N-4-Cl-
CSU_034	5	2-CH3-4-(CH3)3C-	MSU 034	4	2-CH3-4-02N-	MSU 078	4	2-Ph-CH2-
CSU_035	5	2,4-[(CH3)3C-]2-	MSU 035	3	3-0-C(:0)-CH=C(CH3)-4	MSU 080	5	2-CH=CHCH=CH-3
CSU_036	5	4-CH3(CH2)6-0-	MSU 036	4	2,6-([CH3]2CH)2-	MSU 081	3	2-CH=CHCH=CH-3
CSU_037	5	4-Br-3,5-(CH3)2-	MSU 037	4	3-CH3-4-Cl-	MSU 083	4	2,6-Br2-4-CH3-
CSU_038	5	2-Br-4-Cl-	MSU 038	4	2,6-Cl2-4-02N-	MSU 085	4	2,4-Br2-
CSU_039	5	2,4-Cl2-	MSU 039	4	2,4,6-Cl3-	MSU 086	3	4-(CH3)3C-
CSU_040	5	4-(CH3)3C-	MSU 040	3	4-Ph-CH2C(:0)-	MSU 090	3	2,4-Br2-
CSU_041	5	3-02N-4-Cl-	MSU 041	3	4-Cl-3,5-(CH3)2-	MSU 091	5	2-CH3-4-02N-

CSU_042	5	2-Ph-CH2-	MSU042	4	4-Cl-3,5-(CH3)2-	MSU092	5	2,4-Br2-
CSU_043	5	2,6-Br2-4-CH3-	MSU043	3	4-CH3-0			

Appendix III: antidotes' reactivation fraction (original data)

All data represent measurements from *in vitro* systems using brain cells from rats.

PIMP: phthalimidyl isopropyl methylphosphonate

NIMP: nitrophenylethylmethylphosphonate

PXN: paraoxon

DFP: diisopropylphosphofluoridate

Identifier	PIMP	NEMP	PXN	DFP
MSU 001	0.54	0.52	0.56	0.23
MSU 002	0.37	0.32	0.69	0.28
MSU 003	0.34	0.36	0.73	0.2
MSU 004	0.38	0.34	0.55	0.2
MSU 005	0.53	0.53	0.76	0.3
MSU 006	0.65	0.67	0.49	0.09
MSU 007	0.37	0.41	0.32	0.17

MSU 008	0.41	0.33	0.48	0.16
MSU 009	0.3	0.35	0.25	0.16
MSU 010	0.39	0.34	0.52	0.08
MSU 011	0.48	0.36	0.67	0.21
MSU 012	0.34	0.33	0.76	0.27
MSU 013	0.65	0.57	0.45	0.12
MSU 014	0.4	0.25	0.6	0.18
MSU 015	0.48	0.38	0.81	0.22
MSU 016	0.78	0.62	0.54	0.18
MSU 017	0.72	0.69	0.87	0.37
MSU 018	0.71	0.56	0.66	0.2

MSU 019	0.42	0.31	0.41	0.17
MSU 020	0.53	0.47	0.32	0.12
MSU 021	0.72	0.76	0.32	0.15
MSU 022	0.28	0.26	0.16	0.05
MSU 023	0.46	0.37	0.73	0.25
MSU 024	0.24	0.23	0.16	0.02
MSU 025	0.43	0.62	0.35	0.13
MSU 026	0.49	0.53	0.52	0.17
MSU 027	0.25	0.24	0.27	0.03
MSU 028	0.47	0.43	0.74	0.18
MSU 029	0.71	0.74	0.42	0.16

MSU 030	0.38	0.48	0.77	0.14
MSU 031	0.27	0.31	0.72	0.19
MSU 032	0.52	0.58	0.62	0.29
MSU 033	0.55	0.62	0.78	0.4
MSU 034	0.14	0.38	0.68	0.22
MSU 035	0.2	0.25	0	0.14
MSU 036	0.53	0.58	0.65	0.23
MSU 037	0.54	0.61	0.48	0.19
MSU 038	0.38	0.37	0.72	0.17
MSU 039	0.56	0.67	0.47	0.09
MSU 040	0.56	0.44	0.47	0.17

MSU 041	0.54	0.7	0.51	0.17
MSU 042	0.65	0.65	0.25	0.18
MSU 043	0.35	0.45	0.72	0.27
MSU 044	0.51	0.62	0.61	0.23
MSU 045	0.57	0.67	0.86	0.39
MSU 046	0.38	0.45	0.36	0.12
MSU 047	0.49	0.56	0.84	0.37
MSU 048	0.59	0.67	0.42	0.13
MSU 049	0.31	0.42	0.51	0.14
MSU 050	0.34	0.35	0.44	0.15
MSU 051	0.53	0.48	0.59	0.19

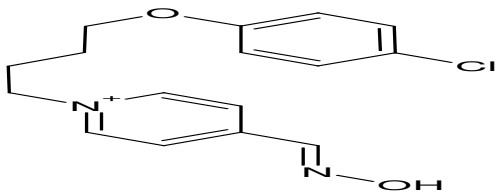
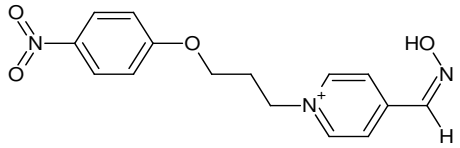
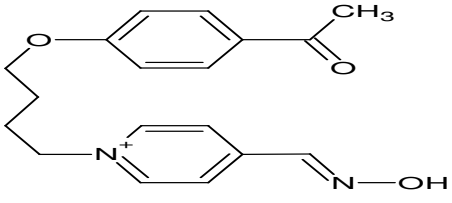
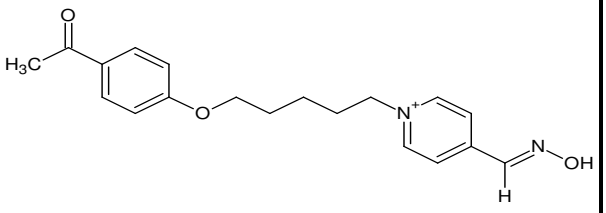
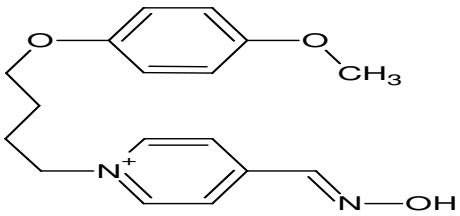
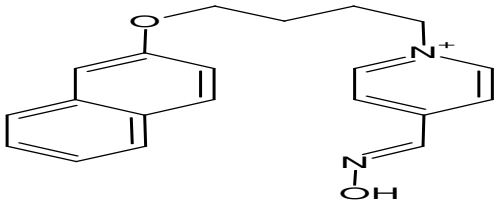
MSU 052	0.47	0.55	0.47	0.23
MSU 053	0.43	0.46	0.24	0.09
MSU 054	0.29	0.36	0.24	0.04
MSU 055	0.51	0.56	0.93	0.5
MSU 056	0.51	0.55	0.56	0.22
MSU 057	0.67	0.66	0.32	0.2
MSU 058	0.39	0.19	0.25	0.12
MSU 059	0.18	0.25	0.4	0.1
MSU 060	0.36	0.33	0.22	0.12
MSU 061	0.31	0.36	0.7	0.32
MSU 062	0.41	0.45	0.94	0.36

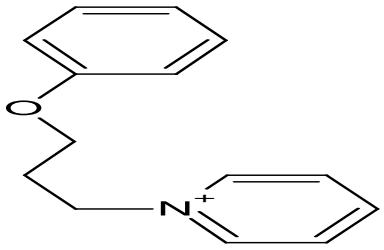
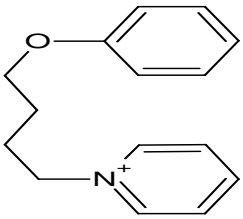
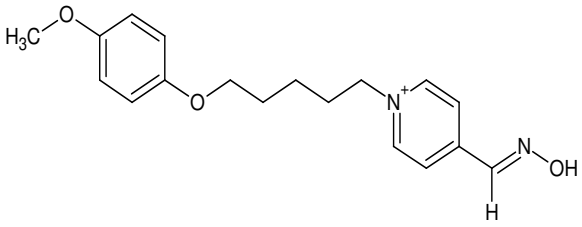
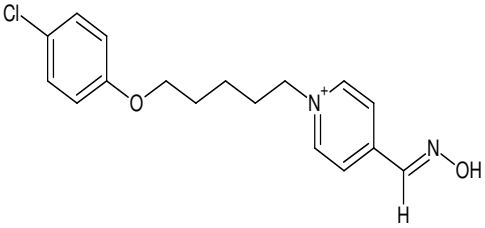
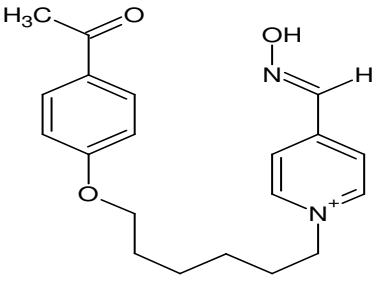
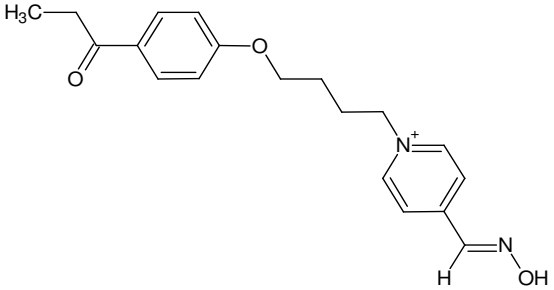
MSU 063	0.46	0.69	0.31	0.17
MSU 064	0.38	0.53	0.46	0.1
MSU 065	0.51	0.6	0.63	0.27
MSU 066	0.48	0.46	0.44	0.15
MSU 067	0.46	0.57	0.38	0.14
MSU 069	0.65	0.59	0.47	0.23
MSU 070	0.16	0.46	0.43	0.06
MSU 071	0.44	0.6	0.31	0.18
MSU 072	0.3	0.32	0.42	0.15
MSU 073	0.34	0.31	0.42	0.13
MSU 074	0.58	0.7	0.67	0.17

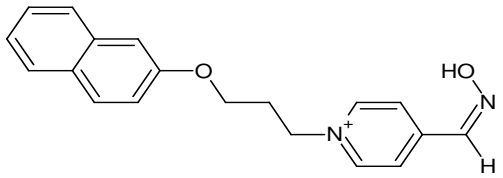
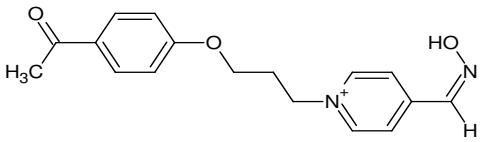
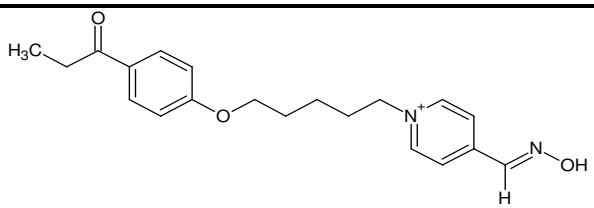
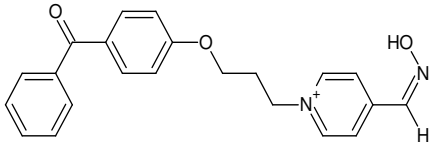
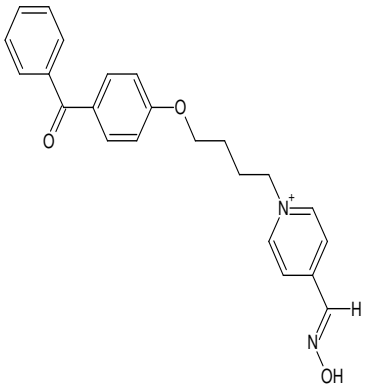
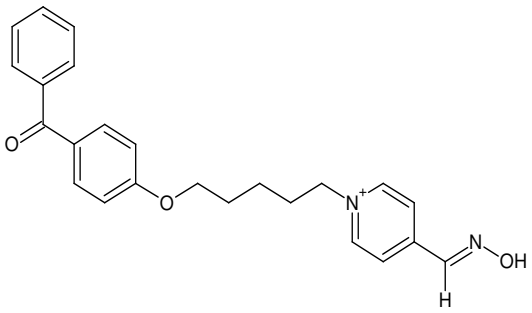
MSU 075	0	0.55	0.6	0
MSU 076	0.36	0.42	0.4	0.14
MSU 077	0.21	0.34	0.61	0.24
MSU 078	0.47	0.5	0.63	0.2
MSU 080	0.58	0.68	0.86	0.49
MSU 081	0.58	0.8	0.8	0.35
MSU 083	0.31	0.44	0.23	0.01
MSU 085	0.51	0.55	0.4	0.2
MSU 086	0.34	0.38	0.35	0.14
MSU 090	0.27	0.41	0.23	0.01
MSU 091	0.31	0.42	0.52	0.2

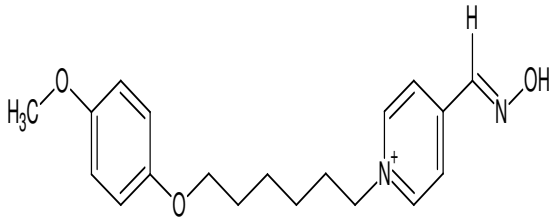
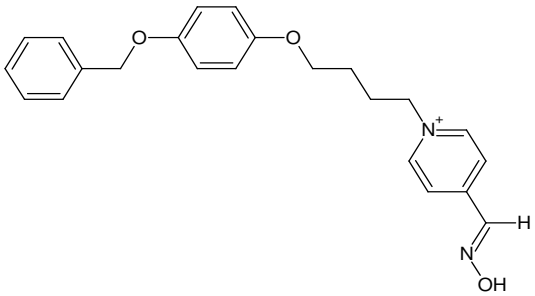
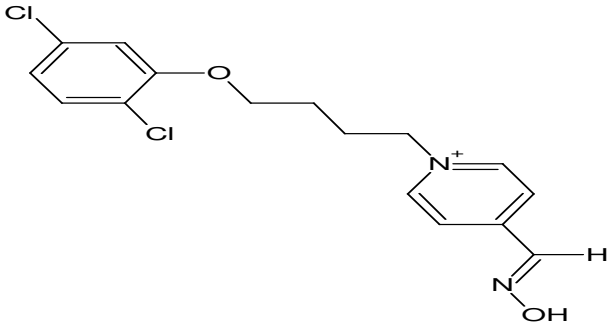
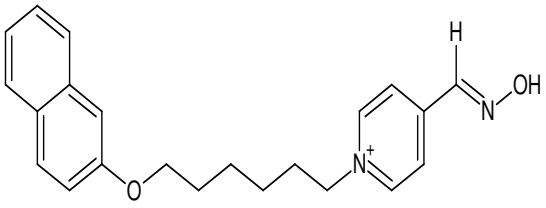
MSU 092	0.36	0.37	0.81	0.28

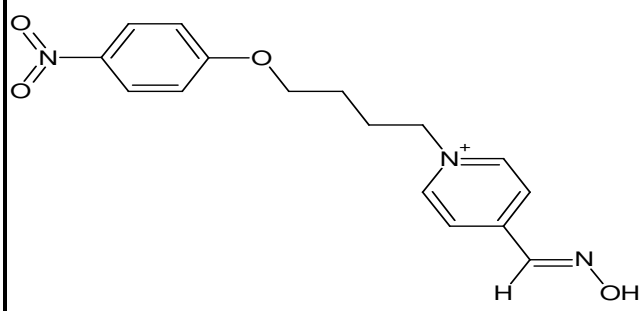
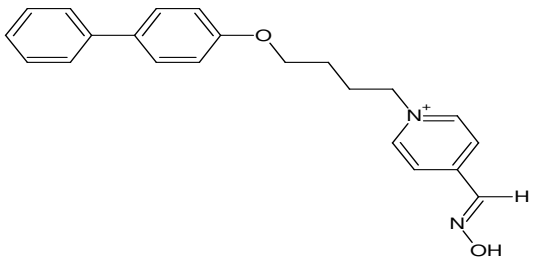
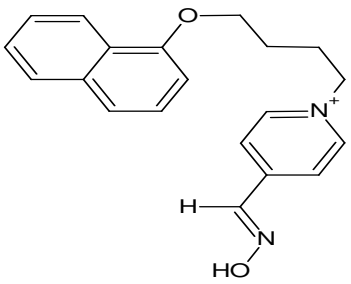
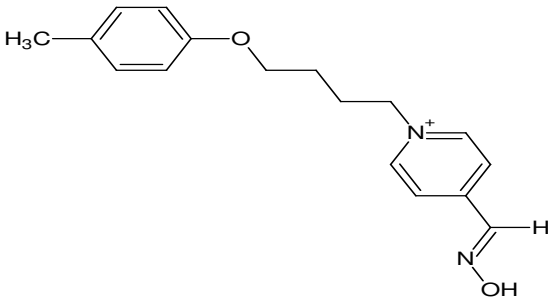
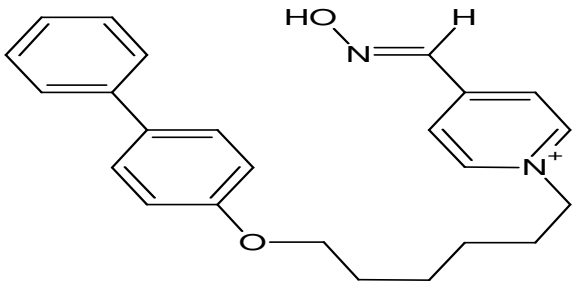
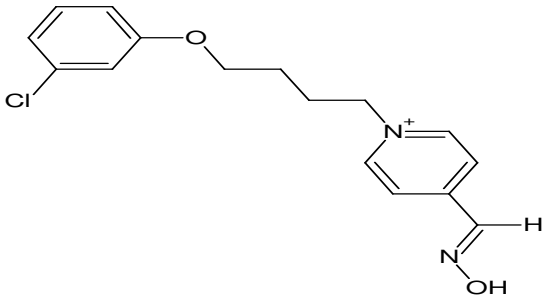
Appendix IV: SMILES representation of candidate antidotes

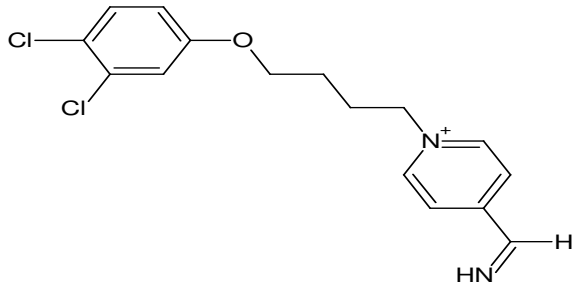
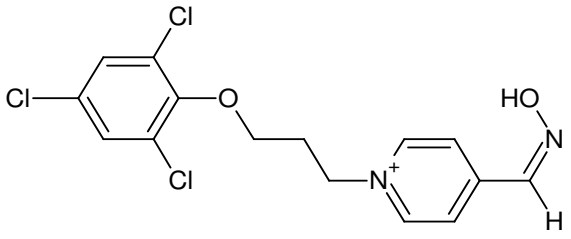
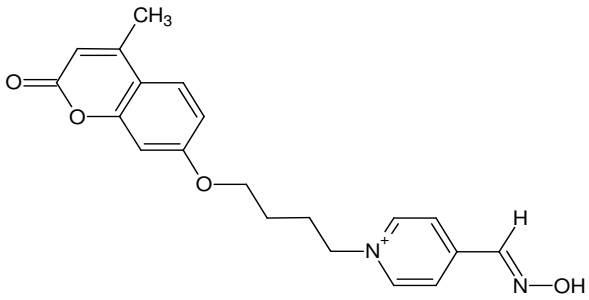
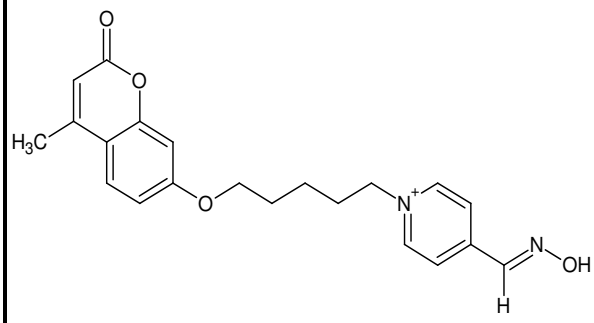
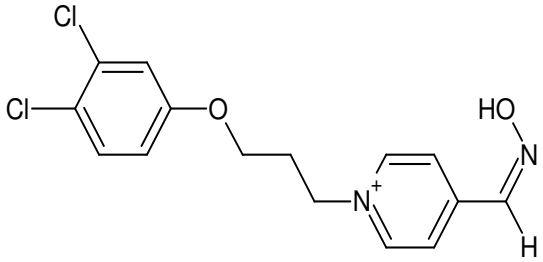
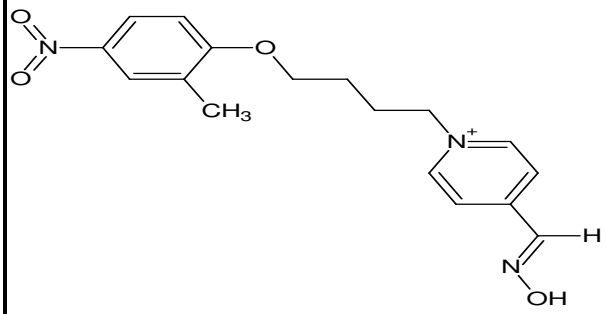
MSU 001	MSU 002
	
c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(Cl)cc2	O=N(=O)-c1ccc(OCCCc2ccc(\C=N\O)cc2)cc1
MSU 003	**MSU 004**
	
c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(C(=O)C)cc2	c1cc(C=NO)cc[n+]1-CCCCCO-c2ccc(C(=O)C)cc2
MSU 005	**MSU 006**
	
c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(OC)cc2	c1cc2cc(OCCCC-[n+]3ccc(C=NO)cc3)ccc2cc1
MSU 007`	**MSU 008**

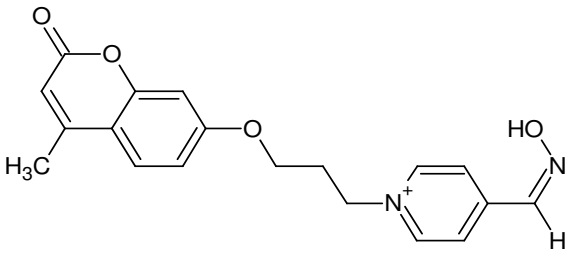
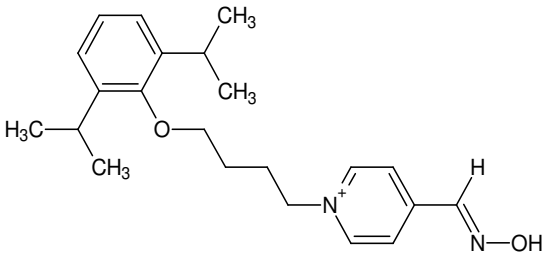
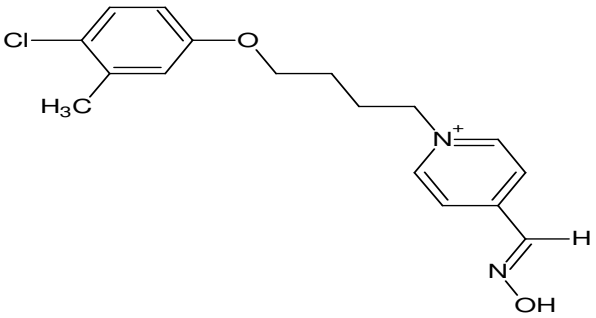
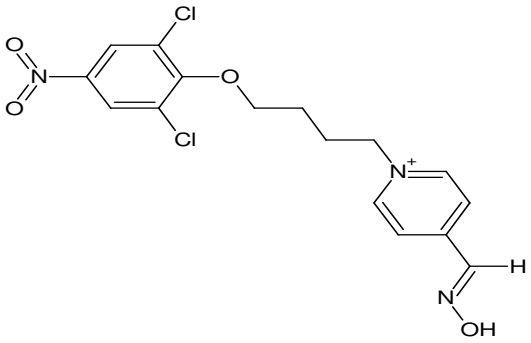
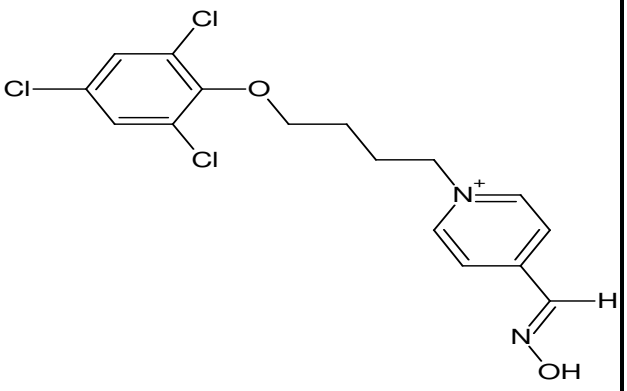
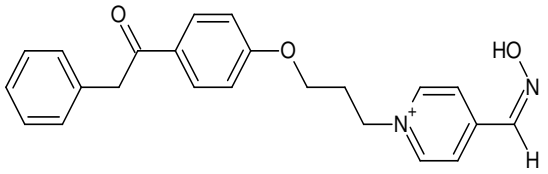
 c1cccc[n+]1-CCCO-c2ccccc2	 c1cccc[n+]1-CCCCO-c2ccccc2
MSU 009	MSU 010
 \c1cc(C=NO)cc[n+]1-CCCCCO-c2ccc(OC)cc2	 c1cc(C=NO)cc[n+]1-CCCCCO-c2ccc(Cl)cc2
MSU 011	MSU 012
 c1cc(C=NO)cc[n+]1-CCCCCCO-c2ccc(C(=O)C)cc2	 c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(C(=O)CC)cc2
MSU 013	MSU 014

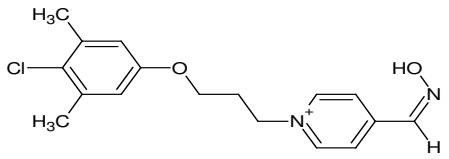
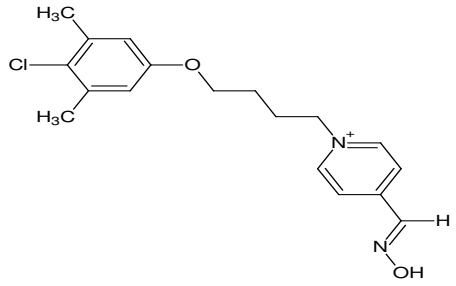
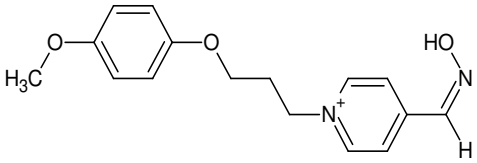
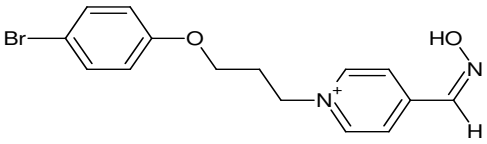
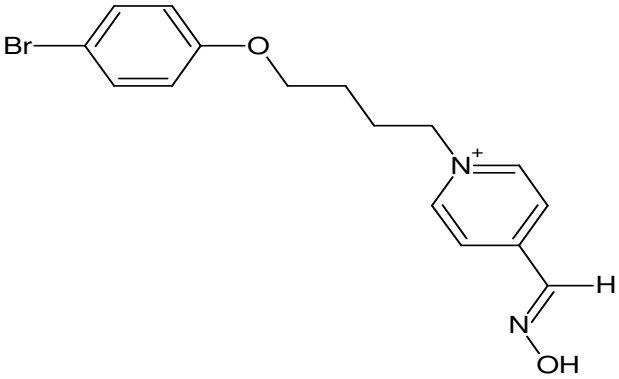
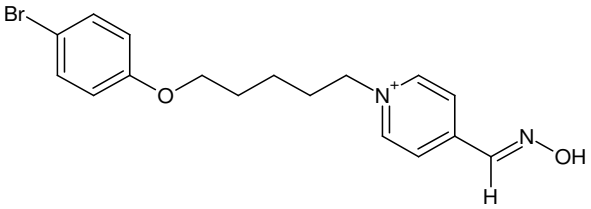
 c1cc2cc(OCCC-[n+]3ccc(C=NO)cc3)ccc2cc1	 c1cc(C=NO)cc[n+]1-CCCO-c2ccc(C(=O)C)cc2
MSU 015	MSU 016
 c1cc(C=NO)cc[n+]1-CCCCCO-c2ccc(C(=O)CC)cc2	 c1cc(C=NO)cc[n+]1-CCCO-c2ccc(C(=O)-c3ccccc3)cc2
MSU 017	MSU 018
 c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(C(=O)-c3ccccc3)cc2	 c1cc(C=NO)cc[n+]1-CCCCCO-c2ccc(C(=O)-c3ccccc3)cc2
MSU 019	MSU 020

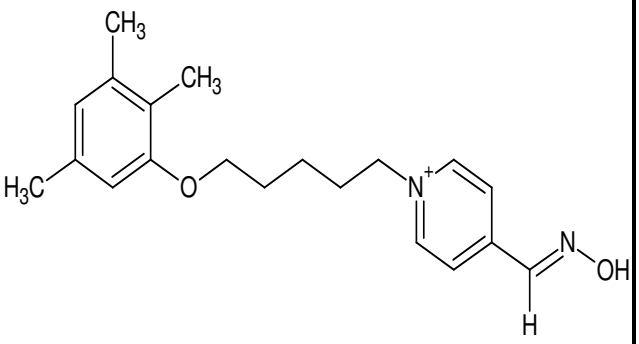
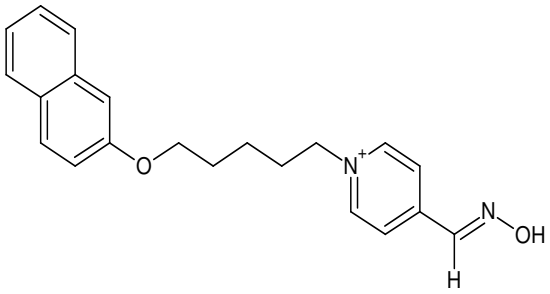
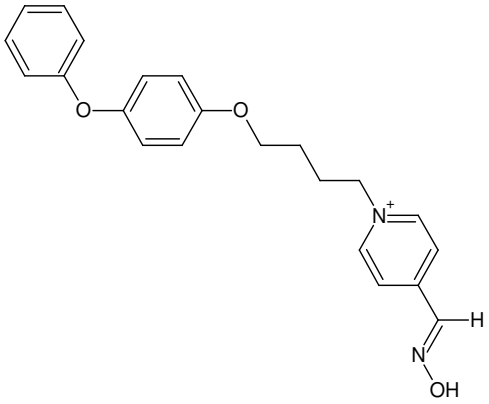
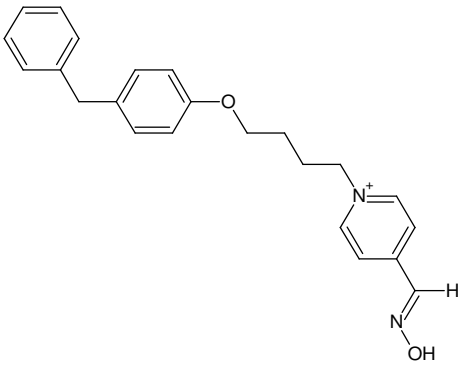
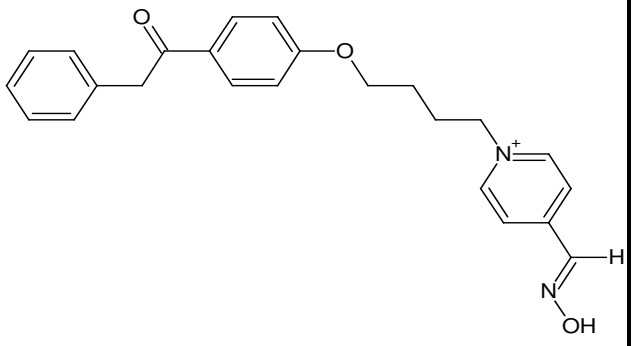
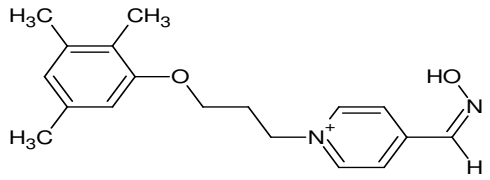
 c1cc(C=NO)cc[n+]1-CCCCCCO-c2ccc(OC)cc2	 c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(OC-c3ccccc3)cc2
MSU 021	MSU 022
 c1cc(C=NO)cc[n+]1-CCCCO-c2c(Cl)ccc(Cl)c2	 c1cc2cc(OCCCCCC-[n+]3ccc(C=NO)cc3)ccc2cc1
MSU 023	MSU 024

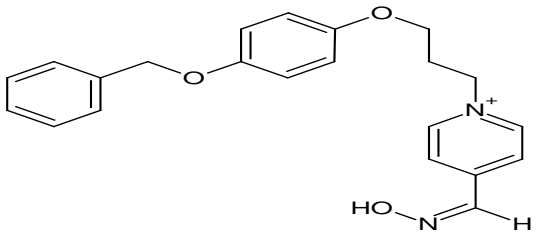
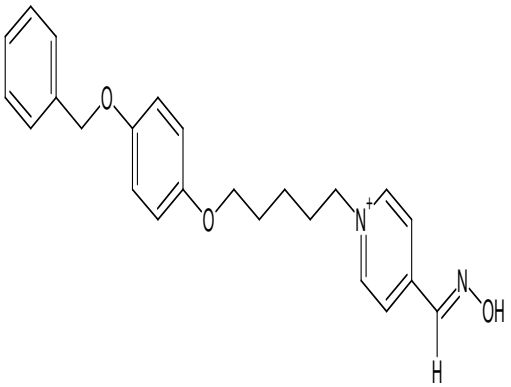
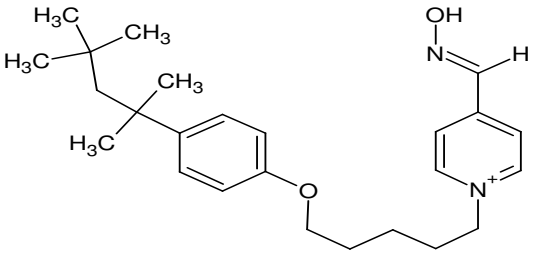
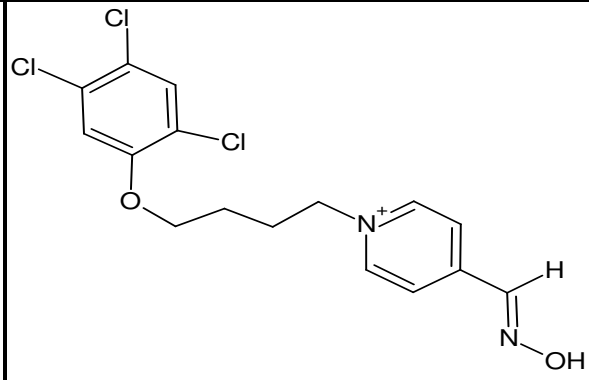
 c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(N(=O)=O)cc2	 c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(c3ccccc3)cc2
MSU 025	MSU 026
 c1cc2c(OCCCC-[n+]3ccc(C=NO)cc3)cccc2cc1	 c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(C)cc2
MSU 027	MSU 028
 c1cc(C=NO)cc[n+]1-CCCCCCO-c2ccc(c3ccccc3)cc2	 c1cc(C=NO)cc[n+]1-CCCCO-c2cc(Cl)ccc2
MSU 029	MSU 030

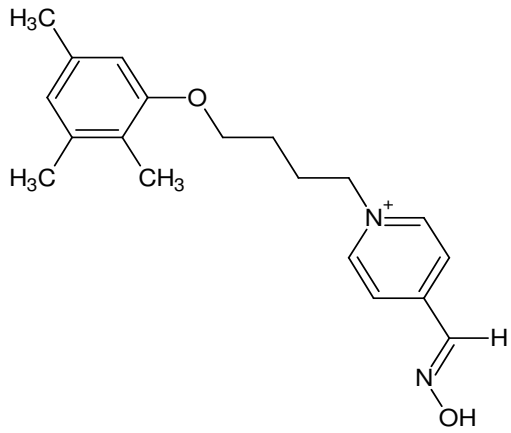
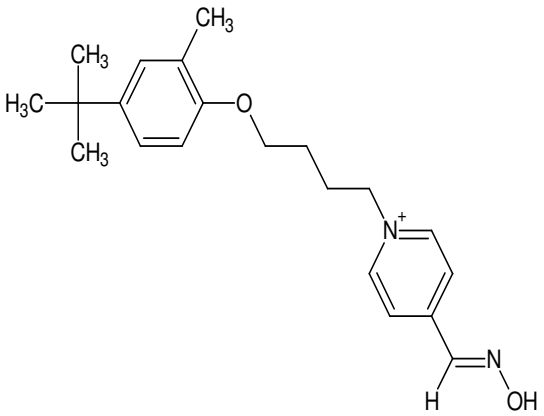
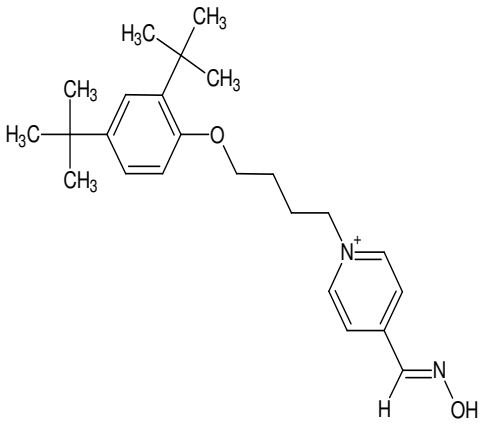
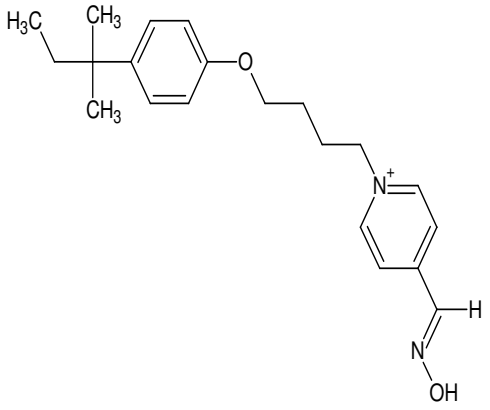
 c1cc(C=NO)cc[n+]1-CCCCO-c2cc(Cl)c(Cl)cc2	 c1cc(C=NO)cc[n+]1-CCCO-c2c(Cl)cc(Cl)cc2(Cl)
MSU 031	MSU 032
 c1(=O)oc2cc(OCCCC-[n+]3ccc(C=NO)cc3)ccc2c(C)c1	 c1(=O)oc2cc(OCCCCC-[n+]3ccc(C=NO)cc3)ccc2c(C)c1
MSU 033	MSU 034
 c1(=O)oc2cc(OCCC-[n+]3ccc(C=NO)cc3)ccc2c(C)c1	 c1cc(C=NO)cc[n+]1-CCCCO-c2c(C)cc(N(=O)=O)cc2
MSU 035	MSU 036

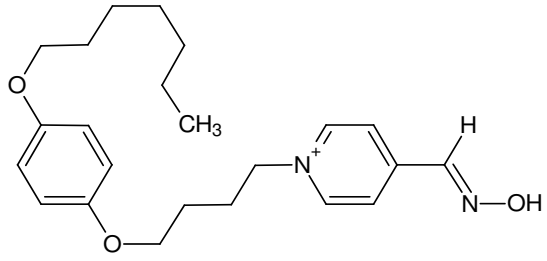
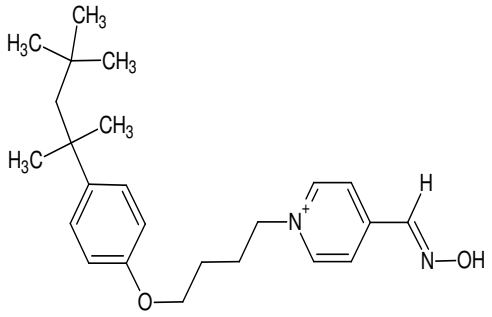
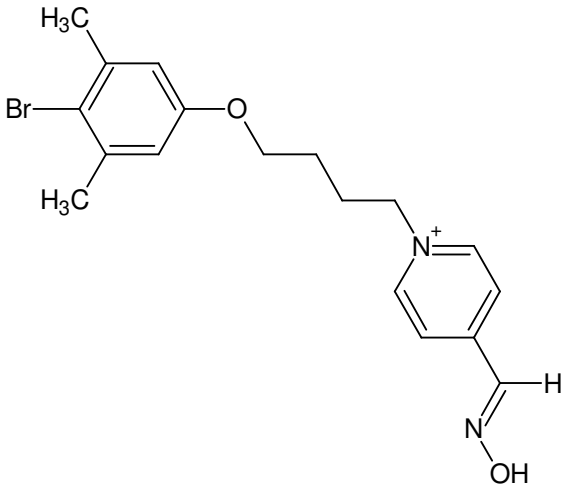
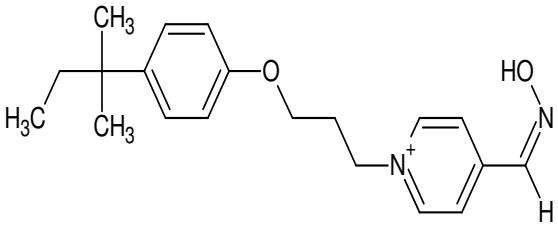
 c1(=O)oc2cc(OCCC-[n+]3ccc(C=NO)cc3)ccc2c(C)c1	 c1cc(C=NO)cc[n+]1-CCCCO-c2c(C(C)C)cccc2(C(C)C)
MSU 037	MSU 038
 c1cc(C=NO)cc[n+]1-CCCCO-c2cc(C)c(Cl)cc2	 c1cc(C=NO)cc[n+]1-CCCCO-c2c(Cl)cc(N(=O)=O)cc2(Cl)
MSU 039	MSU 040
 c1cc(C=NO)cc[n+]1-CCCCO-c2c(Cl)cc(Cl)cc2(Cl)	 c1cc(C=NO)cc[n+]1-CCCO-c2ccc(C(=O)C-c3ccccc3)cc2
MSU 041	MSU 042

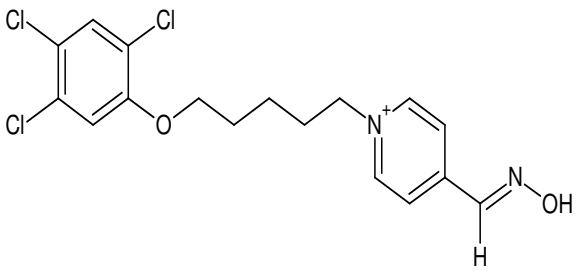
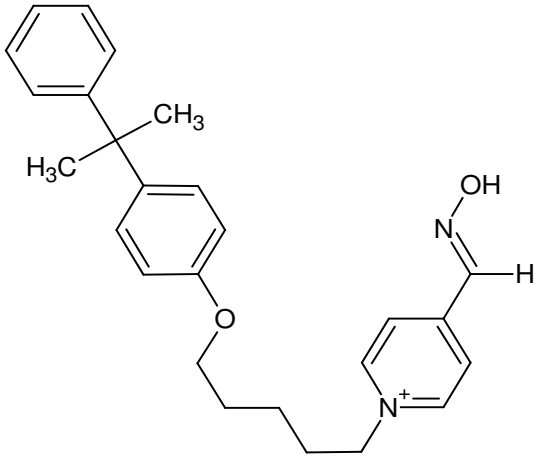
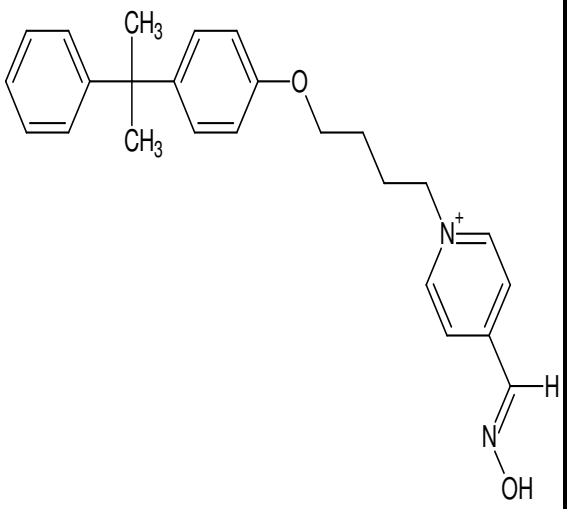
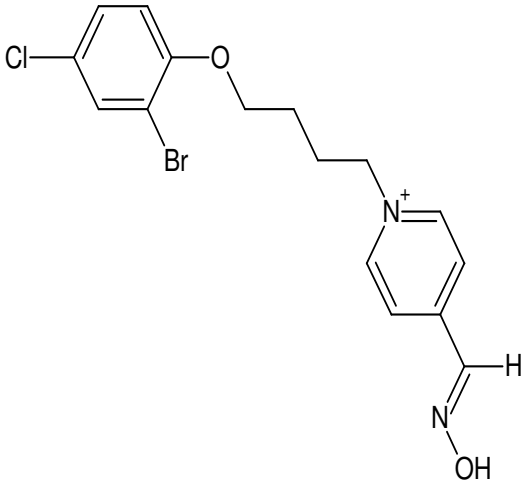
 c1cc(C=NO)cc[n+]1-CCCO-c2cc(C)c(Cl)c(C)c2	 c1cc(C=NO)cc[n+]1-CCCCO-c2cc(C)c(Cl)c(C)c2
MSU 043	MSU 044
 c1cc(C=NO)cc[n+]1-CCCO-c2ccc(OC)cc2	 c1cc(C=NO)cc[n+]1-CCCO-c2ccc(Br)cc2
MSU 045	MSU 046
 c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(Br)cc2	 c1cc(C=NO)cc[n+]1-CCCCCO-c2ccc(Br)cc2
MSU 047	MSU 048

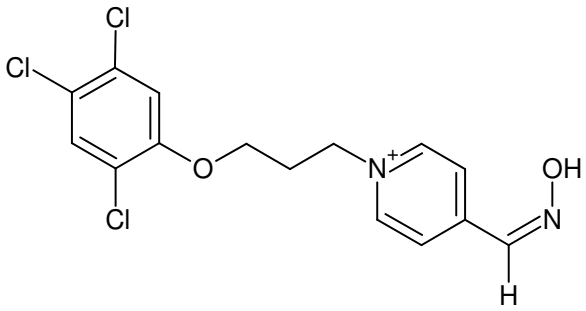
 c1cc(C=NO)cc[n+]1-CCCCCO-c2c(C)c(C)cc(C)c2	 c1cc2cc(OCCCCC-[n+]3ccc(C=NO)cc3)ccc2cc1
MSU 049	MSU 050
 c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(O-c3ccccc3)cc2	 c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(C-c3ccccc3)cc2
MSU 051	MSU 052
c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(C(=O)C-c3ccccc3)cc2 	 c1cc(C=NO)cc[n+]1-CCCO-c2c(C)c(C)cc(C)c2
MSU 053	MSU 054

 c1cc(C=NO)cc[n+]1-CCCO-c2ccc(OC-c3ccccc3)cc2	 c1cc(C=NO)cc[n+]1-CCCCCO-c2ccc(OC-c3ccccc3)cc2
MSU 055	MSU 056
 c1cc(C=NO)cc[n+]1-CCCCCO-c2ccc(C(C)(C)CC(C)(C)C)cc2	 c1cc(C=NO)cc[n+]1-CCCCO-c2c(Cl)cc(Cl)c(Cl)c2
MSU 057	MSU 058

 c1cc(C=NO)cc[n+]1-CCCCO-c2c(C)c(C)cc(C)c2	 c1cc(C=NO)cc[n+]1-CCCCO-c2c(C)cc(C(C)(C)C)cc2
MSU 059	MSU 060
 c1cc(C=NO)cc[n+]1-CCCCO-c2c(C(C)(C)C)cc(C(C)(C)C)cc2	 c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(C(C)(C)CC)cc2

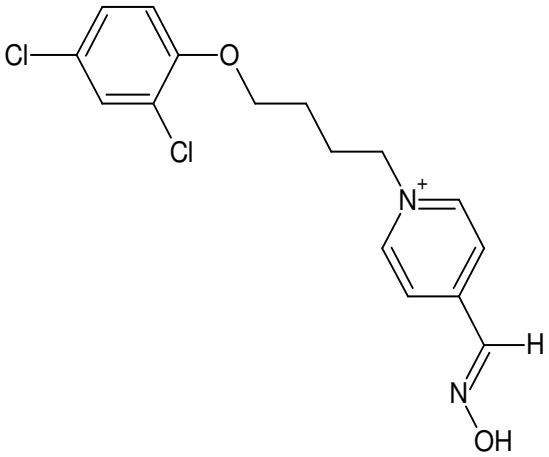
MSU 061	MSU 062
 c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(OCCCCCCC)cc2	 c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(C(C)(C)CC(C)(C)C)cc2
MSU 063	MSU 064
	 c1cc(C=NO)cc[n+]1-CCCO-c2ccc(C(C)(C)CC)cc2

c1cc(C=NO)cc[n+]1-CCCCO-c2cc(C)c(Br)c(C)c2	
MSU 065	MSU 066
 c1cc(C=NO)cc[n+]1-CCCCCO-c2c(Cl)cc(Cl)c(Cl)c2	 c1cc(C=NO)cc[n+]1-CCCCCO-c2ccc(C(C)(C)-c3ccccc3)cc2
MSU 067	MSU 069
 c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(C(C)(C)-c3ccccc3)cc2	 c1cc(C=NO)cc[n+]1-CCCCO-c2c(Br)cc(Cl)cc2
MSU 070	MSU 071



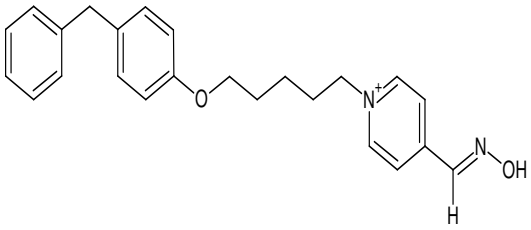
c1cc(C=NO)cc[n+]1-CCCO-c2c(Cl)cc(Cl)c(Cl)c2

MSU 072

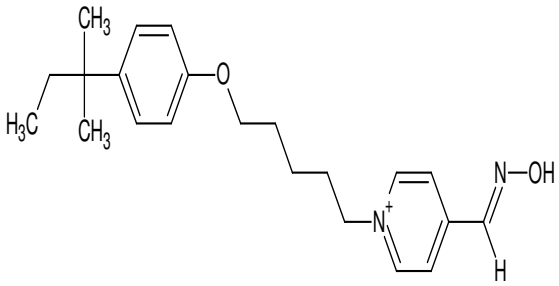


c1cc(C=NO)cc[n+]1-CCCCO-c2c(Cl)cc(Cl)cc2

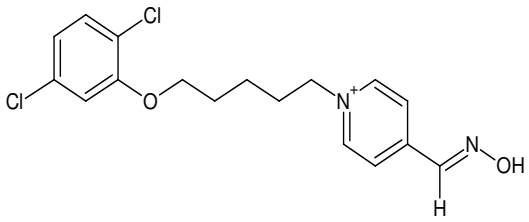
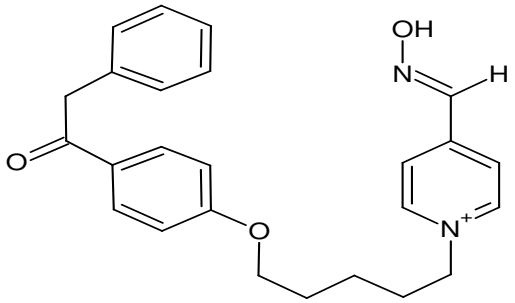
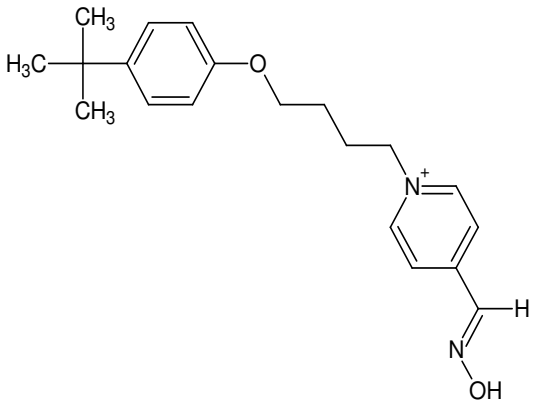
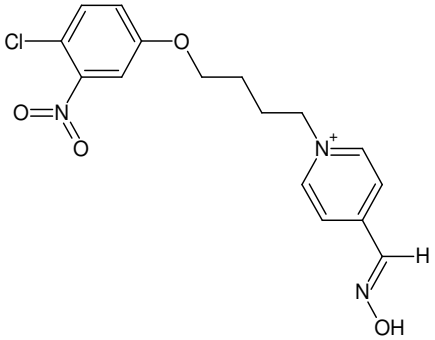
MSU 073

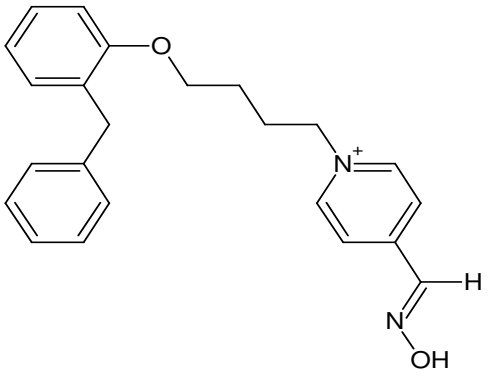


c1cc(C=NO)cc[n+]1-CCCCCO-c2ccc(C-c3ccccc3)cc2



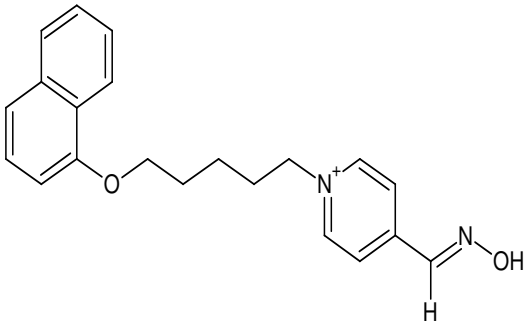
c1cc(C=NO)cc[n+]1-CCCCCO-c2ccc(C(C)(C)CC)cc2

MSU 074	MSU 075
 c1cc(C=NO)cc[n+]1-CCCCCO-c2c(Cl)ccc(Cl)c2	 c1cc(C=NO)cc[n+]1-CCCCCO-c2ccc(C(=O)C-c3ccccc3)cc2
MSU 076	MSU 077
 c1cc(C=NO)cc[n+]1-CCCCO-c2ccc(C(C)(C)C)cc2	 c1cc(C=NO)cc[n+]1-CCCCO-c2cc(N(=O)=O)c(Cl)cc2
MSU 078	MSU 080



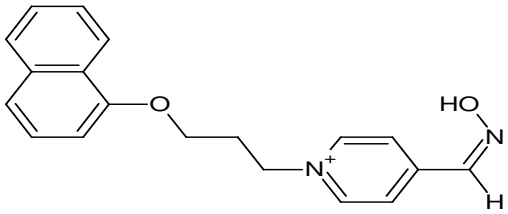
c1cc(C=NO)cc[n+]1-CCCCO-c2c(C-c3ccccc3)cccc2

MSU 081

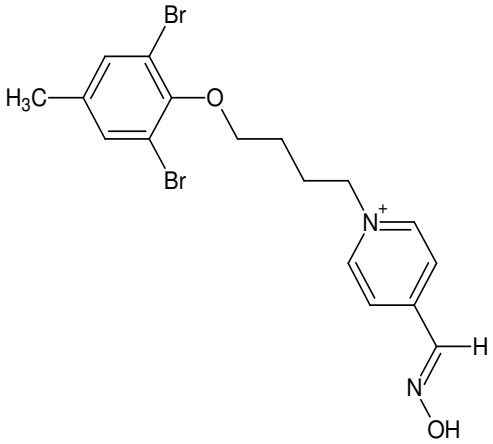


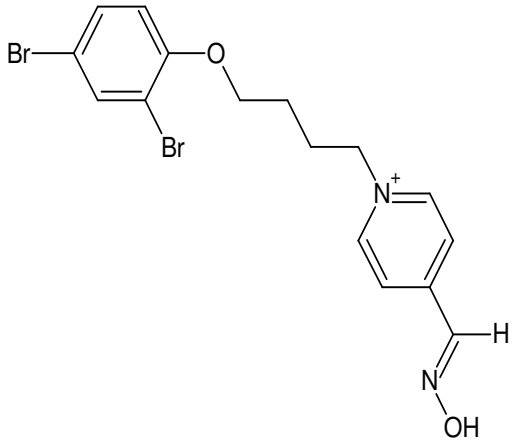
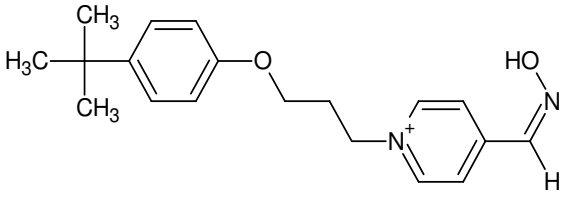
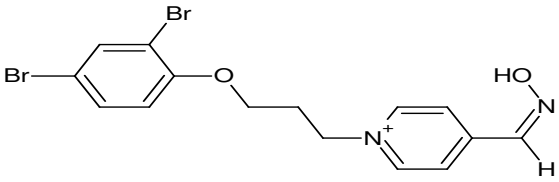
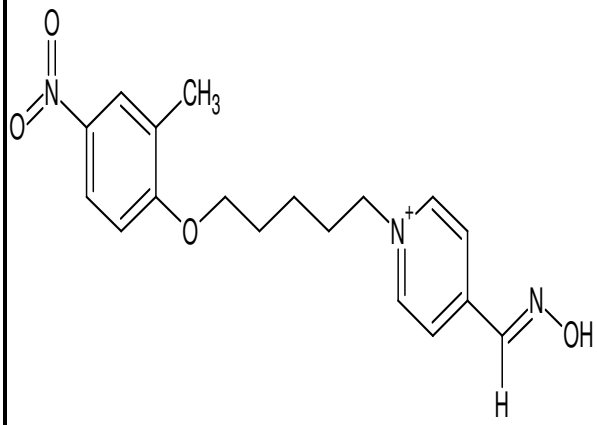
c1cc2c(OCCCCC-[n+]3ccc(C=NO)cc3)cccc2cc1

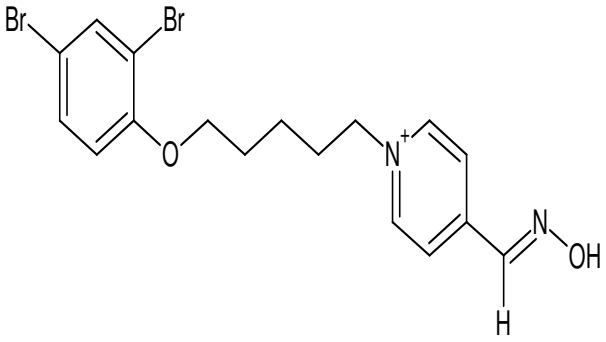
MSU 083



c1cc2c(OCCC-[n+]3ccc(C=NO)cc3)cccc2cc1



	c1cc(C=NO)cc[n+]1-CCCCO-c2c(Br)cc(C)cc2(Br)
MSU 085	MSU 086
 c1cc(C=NO)cc[n+]1-CCCCO-c2c(Br)cc(Br)cc2	 c1cc(C=NO)cc[n+]1-CCCO-c2ccc(C(C)(C)C)cc2
MSU 090	MSU 091
 c1cc(C=NO)cc[n+]1-CCCO-c2c(Br)cc(Br)cc2	 c1cc(C=NO)cc[n+]1-CCCCCO-c2c(C)cc(N(=O)=O)cc2

MSU092	
	c1cc(C=NO)cc[n+]1-CCCCCO-c2c(Br)cc(Br)cc2

Appendix V: Python codes archives

The codes used to achieve this works is archived in the following OSF repositories:

Codes and other essential materials for chapter I could be found in:

ACAT_RandomForest ; Identifier : DOI 10.17605/OSF.IO/93RXS

Codes and other essential materials for chapter II could be found in:

antidote_XGboost ; Identifier : DOI 10.17605/OSF.IO/C9264