CALIFORNIA STATE UNIVERSITY SAN MARCOS

THESIS SIGNATURE PAGE

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE

MASTER OF SCIENCE

IN

COMPUTER SCIENCE

THESIS TITLE: Improving pharmacological research of HIV-1 Integrase Inhibition using Differential Evolution-Binary Particle Swarm Optimization and Non-Linear Adaptive Boosting Random Forest Regression

AUTHOR: Richard Adrian Galvan

DATE OF SUCCESSFUL DEFENSE: 5-5-2015

THE THESIS HAS BEEN ACCEPTED BY THE THESIS COMMITTEE IN
PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF
SCIENCE IN COMPUTER SCIENCE.

Dr. Ahmad R. Hadaegh

THESIS COMMITTEE CHAIR

SIGNATURE

May 5, 2015

DATE

Dr. Xiaoyu Zhang

THESIS COMMITTEE MEMBER

SIGNATURE

5/5/15

DATE

**Improving pharmacological research of HIV-1 Integrase Inhibition
using Differential Evolution-Binary Particle Swarm Optimization
and Non-Linear Adaptive Boosting Random Forest Regression**

By Richard Adrian Galvan

Department of Computer Science

California State University San Marcos

Abstract

In the work, we present results produced from non-linear QSAR models developed and implemented using evolutionary computation and Random Forest Regression to study the effectiveness of dimeric Aryl β -Diketo Acids on HIV-1 Integrase inhibition. Dimeric Aryl β -Diketo Acids have been proven to be effective inhibitors of the biological mechanism of protein transfer known as HIV-integrase.

This report extends an initial investigative study of Aryl β -Diketo Acids for HIV-1 Integrase inhibition that used linear QSAR models implemented using a Multiple Linear Regression (MLR) machine learning strategy and a hybridized Differential Evolution-Binary Particle Swarm Optimization (DE-BPSO) algorithm to select and identify drug descriptors having the greatest inhibitory effect on HIV-1 Integrase, respectively. This comparative study uses a non-linear Random Forest Regression (RFR) strategy with Adaptive Boosting (AdaBoost) to generate QSAR models with greater predictive accuracy in identifying optimal Aryl β -Diketo Acids drug descriptor subsets that can more effectively inhibit HIV protein enzyme activity.

KEYWORDS: Adaptive Boosting (AdaBoost), Acquired Immune deficiency Syndrome (AIDS), Differential Evolution-Binary Particle Swarm Optimization (DE-BPSO), Quantitative Structure-Activity Relationship (QSAR), and Random Forest Regression (RFR)

Acknowledgements

I'd like to thank my family for standing together and being my support and Melanie Lauren Hernandez, my spirit and my love who helped inspire the hopes and goals of this research in seeking to advance the medical community's ability to develop unique and novel medicines to help those most affected by the ravages of HIV-1. You are all my lights in the dark.

I'd also like to thank my advisor Dr. Ahmad R. Hadaegh for his guidance, wisdom, good faith and support with this investigation, as well as my partners Matinehalsadat Kashani Moghaddam and Ian Kane for their contributions and hard work as part of this research effort.

Table of Contents

Chapter 1	Introduction.....	8
1.1	Overview	8
1.2	Objectives.....	9
Chapter 2	Background	12
2.1	Role of Data mining and Machine Learning in Drug Design	12
2.2	Role of QSAR in Bioinformatics and Computation Chemistry	13
2.3	Model Development	13
2.4	Linear and Non-Linear Models.....	14
2.5	Developing a Non-Linear QSAR Model	15
Chapter 3	Methods	16
3.1	Workflow of Non-Linear QSAR Modeling	16
3.2	Implementation Architecture	17
3.3	Validating a QSAR model	19
Chapter 4	Models	20
4.1	Decision Tree Bagging	20
4.2	Tree Bagging to Random Forests	20
4.3	Improving Random Forests with Adaptive Boosting	22
4.4	Comparing to commercial ensemble methods in previous research	22
Chapter 5	Results.....	24
5.1	Summary	24
5.2	Datasets	24
5.3	Experiment Parameters	26
5.4	Comparing results to previous Non-linear and Linear QSAR Models	26
5.5	Discussion of optimal Non-Linear QSAR models	28
Chapter 6	The QSAR Distributed Web Modeling Framework	32
6.1	Motivation.....	32
6.2	Design and Development.....	33
6.3	Components and Architecture.....	33
6.4	Advancing Distributable QSAR Modeling in Drug Design	36

6.5	Further Development.....	38
Chapter 7	Conclusion	39
7.1	Summary and Future work	39
7.2	Research Contribution	40
Chapter 8	References.....	42

List of Figures

Figure 1 HIV-1 retroviral host integration [35]	8
Figure 2. Physiochemical structure of Aryl β -Diketo acids descriptor datasets in this study [1]	10
Figure 3 A high-level perspective of the drug discovery process workflow as illustrated by the Computational Materials Science Initiative [36]	12
Figure 4. Machine learning in QSAR drug design.....	14
Figure 5. Overview of the DE-BPSO and non-Linear Random Forest QSAR architecture ...	16
Figure 6. Random Forest Decision Tree Hierarchy [21].....	21
Figure 7 Distribution of physiochemical descriptors for 37 Aryl β -Diketo acids drug dataset from DE-BPSO MLR linear QSAR Study [1].....	25
Figure 8 Distribution of physiochemical descriptors for 91 Aryl β -Diketo acids used in commercial MLR, RF and TreeNet study [30]	25
Figure 9. Regression plot for DE-BPSO 37 Aryl β -Diketo Acid dataset.....	31
Figure 10. Regression plot for 91 Aryl β -Diketo Acid dataset	31
Figure 11. QSAR Web Modeling (MVC) architecture.....	34
Figure 12. QSAR Web landing page	35
Figure 13. Configuring a model as part of a training session	35
Figure 14. Distribution of QSAR training models into task queues monitored by separate worker processes	37
Figure 15. Application-level message passing between worker processes	38

List of Tables

Table 1 Comparing predictive quality statistics for the best QSAR models from the Sci-Kit Learn Non-linear AdaBoost RFR [9] with DE-BPSO and the DE-BPSO MLR implementation [1] trained on the 37 Aryl β -Diketo Acid dataset.....	26
Table 2 Comparing predictive quality statistics for the best QSAR models from the Sci-Kit Learn Non-linear DE-BPSO AdaBoost RFR [9] and the commercial SPM implementation of MLR and TreeNet [30] trained on the 91 Aryl β -Diketo Acid dataset	27
Table 3 Comparing the root-mean-square-error and standard error of the estimate for the best QSAR models from the Sci-Kit Learn Non-linear DE-BPSO AdaBoost RFR [9] and the custom DE-BPSO MLR implementation [1] trained on the 37 Aryl β -Diketo Acid dataset..	27
Table 4 Comparing the root-mean-square-error and standard error of the estimate for the best QSAR models from the Sci-Kit Learn Non-linear DE-BPSO AdaBoost RFR [9] and the commercial SPM implementation of MLR and TreeNet [30] trained on the 91 Aryl β -Diketo Acid dataset.....	28

Table 5 Feature descriptors and corresponding importance for the best QSAR model trained from the Sci-Kit Learn Non-linear AdaBoost RFR [9] on the non-linear 37 Aryl β -Diketo Acid dataset.....	29
---	----

Table 6 Feature descriptors and corresponding importance for the best QSAR model trained from the Sci-Kit Learn Non-linear AdaBoost RFR [9] on the non-linear 91 Aryl β -Diketo Acid dataset.....	29
---	----

Chapter 1 Introduction

1.1 Overview

The human immunodeficiency virus type 1 (HIV-1) is a retrovirus responsible for triggering the acquired immunodeficiency syndrome (AIDS) disease. Based on estimates from the World Health Organization (WHO), 35.3 million people were living with HIV at the end of 2012. In the same year, some 2.3 million people became newly infected, and 1.6 million died of AIDS-related causes [2]. As of 2015, exactly 25 pharmaceutical drug compounds have been approved for clinical use in the U.S. [3]. Most of the identified compounds affect reverse transcriptase and protease, two enzymes that play a vital role in the replication cycle of HIV-1

Figure 1 HIV-1 retroviral host integration [35]

Different combinations of compounds targeting these enzymes are created, tested and prescribed as a drug cocktail to affected patients. Crystallization of these enzyme's binding

mechanism provide a way of inhibiting additional reproductive activity of the virus. Although inhibitor treatments targeting the reverse transcriptase and protease have been developed, the crystal structure of the integrase enzyme is less understood. Insight into these structures would help give researchers a better understanding of the conformation and binding mechanism of inhibitors and how these are affected by the HIV-1 virus' drug resistance.

The challenge for current pharmacological research lies primarily in developing more effective compounds as the HIV-1 virus genome continues to mutate and evolve to develop resistance to current forms of anti-HIV-1 inhibitory treatments. QSAR models provide a way of understanding each drug molecule's binding mechanism in terms of its physiochemical properties, as well as corresponding inhibitor-virus protein interactions. Dimeric Aryl β -Diketo acids represent a particularly potent class of chemical molecules known to be effective inhibitors of the protein strand transfer mechanism of HIV-1 integrase, combinations of these chemical molecules are prescribed as a drug cocktail called highly active anti-retroviral therapy (HAART). Mutations in the HIV-1 retrovirus, however, trigger a natural growth in resistance to current HIV-I inhibitors, which motivates the need for further research and design of new anti- HIV-1 drug treatments [1][23].

1.2 Objectives

The goal of this research was to use a hybrid evolutionary feature selection strategy called DE-BPSO [1] with the open-source SciKit Learn implementation of non-linear Adaptive Boosting Random Forest Regression [9] to develop non-linear QSAR models to predict optimal drug feature descriptors from two sets of 37 and 91 experimental Aryl β -Diketo acids, respectively, targeted at inhibiting biological processes that play a critical role in the HIV-I retrovirus' reproductive cycle. The target metric used in this study is the drug's pIC50 value, which represents the scaled inhibitory concentration of a drug at which point the HIV-1 protein integrase activity is reduced by 50%. As part of this study train a non-linear QSAR model using open-source SciKit Learn [9] AdaBoost Random Forest Regression with DE-BPSO feature selection and show it as an improvement over optimal linear and non-linear QSAR models produced using custom DE-BPSO MLR and commercial MLR and ensemble strategies. The first study used a custom DE-BPSO MLR strategy to train an optimal linear QSAR model from a set of 37 Aryl β -Diketo acids [1], a second study used commercial

linear MLR and non-linear ensemble strategies to train optimal QSAR models from a set of 91 Aryl β -Diketo acids [1][30]

Figure 2. Physiochemical structure of Aryl β -Diketo acids descriptor datasets in this study [1]

With support of evolutionary computation for feature selection and non-linear machine learning, we show that non-linear QSAR models are effective in predicting drug descriptor compounds with greater inhibitory impact on HIV-1 integrase. In the absence of effective QSAR modeling techniques, pharmacological researchers would need to manually test different combinations of drug feature descriptors. This approach is very time consuming and there's no guarantee that the produced drug treatment will be viable, while at the same time minimizing potential side effects; automating this process allows researchers to more quickly evaluate and assess the potency of these treatments.

In brief, this work makes the following contribution:

- We develop a non-linear model using Random Forest with Adaptive Boosting
- We compare results of this new model with work previously done with linear MLR models on this same data set.

- We also discuss contributions made to a distributable modeling system that uses an open-source relational database platform to store results of QSAR models trained using a standard web browser.

The rest of this paper is organized as follows, we start by describing the background of our work which includes the role of data mining and machine learning in drug design and the main differences between the linear and non-linear data models. In Chapter 2, we provide an a general background of QSAR modeling and its application in supporting computational biochemistry research. In Chapter 3, we will describe the methods used in this research and illustrate the overall architecture of our work. Next we describe details about the Random Forest model in Chapter 4, and present the results in chapter 5. In Chapter 6, we review a QSAR web modeling framework that integrates an open-source relational database system from a previous linear and non-linear QSAR study [30] with a distributable computation architecture and user interface to provide support for a web service front-end for students and researchers to train QSAR models in parallel across several independently allocated, application-level worker process queue. Finally, Chapter 7 provides a summary of the results and the related information. We conclude this paper with presenting the future work and methods that can be adopted to improve future non-linear QSAR modeling studies.

Chapter 2 Background

2.1 Role of Data mining and Machine Learning in Drug Design

Data mining and machine learning are multi-disciplinary fields that combine elements of statistical analysis, distributed computing, artificial intelligence and database systems.

Machine learning is both the practice and study of having a computer achieve a task without having to be explicitly programmed. The goals of machine learning are similar to data mining, i.e. systems in both contexts are typically targeted at probabilistic modeling of sparse problem domains containing a potentially large number of features. Machine learning algorithms attempt to correlate disparate, high-dimensional data and focuses on using this data for continuous self-improvement and understanding within the target domain space it is being trained against; machine learning algorithms are designed to detect patterns in training data and be capable of self-adjusting accordingly.

The biopharmaceutical industry and medical community at large are increasingly employing the use of a variety of machine learning and data-mining methodologies to handle processing and interpretation of medical data from a variety of both public and private research dataset repositories.

Figure 3 A high-level perspective of the drug discovery process workflow as illustrated by the Computational Materials Science Initiative [36]

2.2 Role of QSAR in Bioinformatics and Computation Chemistry

Machine learning and data mining tools are at the heart of the medical research community's efforts at computer aided drug design, of which Quantitative Structure-Activity Relationship (QSAR) modeling plays a critical role [5]. The study of QSAR was pioneered in the early 1960s when it was found that the biological activity of a chemical compound could be defined as a function of its physiochemical properties. These properties are derived from chemical compound structures that are mapped to the target biological activity as part of the QSAR modeling process. The results of this mapping are quantified by a statistically measureable value that can be used as a basis for correlating the physiochemical properties to the desired target biological activity. Examples of physiochemical properties includes solvent accessible areas, total energy, dipole moment and molecular weight, among others. Over the years a variety of standard QSAR properties have been derived for use in modeling and analyzing additional chemicals within the pharmaceutical research community [20].

2.3 Model Development

In this research, a non-linear approach has been implemented and studied with a focus on identifying drug descriptors contributing most to the inhibition of HIV-1 Integrase by training a Random Forest Regressor, enhanced with adaptive boosting [6], to predict a target pIC₅₀ value from drug feature descriptor subsets selected by DE-BPSO, to come up with a more effective model for predicting the molecular structure of potentially more novel and effective inhibitory drug treatments. The results produced from simulations of our non-linear Random Forest model are compared to those of the linear model developed in a previous study involving linear MLR QSAR models [1]. The architectural model of the work is shown in Figure 4.

Figure 4. Machine learning in QSAR drug design

2.4 Linear and Non-Linear Models

When designing supervised non-linear QSAR models, there are two types of machine learning strategies to consider, linear and non-linear. In linear modeling, each feature is first assigned a weight; a target value can then be predicted by calculating the average of these weights. Linear models are structured in such a way that when we increase the value of a feature by a certain amount, the target variable will change by a proportional amount. The proportional amount is determined by the weight of the feature in the model. For example, if the weight of a feature has a value of 1.5 and the feature value changes by 2.0, the target variable prediction will increase by 3.0. With non-linear models, similar changes in a feature's value will have a varying effect on the target pIC50 depending on the feature's initial value. Non-linear models are more resilient at coming up with a best fit from the training data, since they are better capable of generalizing data sets with high-dimensional feature spaces.

2.5 Developing a Non-Linear QSAR Model

The linear implementation in a previous QSAR study used a hybridized Differential Evolution and Binary Particle Swarm Optimization (DE-BPSO) evolutionary computation strategy coupled with a Multi-Linear Regression (MLR) machine learning algorithm as part of the linear QSAR modeling procedure [1].

Through a variety of benchmark optimization problems, DE was found to outperform standalone PSO, while at the same time providing more robust and consistent results over continuous simulations [22]. Binary Particle Swarm Optimization (BPSO) was proposed as an implementation of PSO designed to solve discretized problems, with dimensions of each particle's position being represented by a 0 or 1, [14][29]. A hybridized approach that leverages DE has been shown to effectively improve BPSO, which like PSO, is traditionally more sensitive to its parameters and much more dependent on random initialization of the particles in the swarm. DE uses mutation and cross-over to evolve the velocity vectors which are then used to update the positions of swarm particles in every iteration of the swarm. This causes the search space to be explored by each individual particle with greater efficiency [1]

To help support the design and development of non-linear models, a modular approach was taken to help ensure the DE-BPSO feature selection implementation was independent of the non-linear machine learning implementations used to train the models. This design provided additional maintainability support for our DE-BPSO and non-linear Random Forest implementations since each could be more easily tested and validated independently. Another motivation for this approach was to maintain a decoupled architecture in order to more easily extend the framework to leverage other non-linear models for future studies.

Chapter 3 Methods

3.1 Workflow of Non-Linear QSAR Modeling

Training a non-linear QSAR model involves two core operations, feature selection and feature fitting. Training data is first processed by a DE-BPSO evolutionary computation module, which performs feature selection on the drug descriptor dataset. DE-BPSO performs feature selection at a high-level by iteratively evolving velocities of an initialized binary particle swarm from which a series of rules are used to determine the discrete bit values of each particle's position vector which models the drug descriptor subset selected to be trained by the Random Forest module [1].

Figure 5. Overview of the DE-BPSO and non-Linear Random Forest QSAR architecture

3.2 Implementation Architecture

The development process for non-linear models used in this study included implementing a non-linear Random Forest algorithm [4] using Python and the open-source SciKits-Learn machine learning library [9]. To provide additional computational and test support, the module was integrated into a distributed task framework that used a custom relational database designed to capture configuration parameters used to train non-linear QSAR models [15]; this was done to persist configuration settings of optimal models for use in future studies. The addition of a distributable computation framework allowed us to run multiple training simulations simultaneously to allow the analysis of several models at once [15]. Each distributed task is a unit of work that defines the QSAR model generation process outlined in Figure 5.

The core architecture for drug descriptor feature selection and non-linear QSAR model begins by initializing the binary particle swarm, while the last **Stage V** represents the deterministic terminal state controlling when model training is complete. The first three stages of the workflow use hybridized DE-BPSO algorithm [1] as the feature selection algorithm. This algorithm works by combining the mutation and recombination rules of Differential Evolution (DE) with a discretized version of Particle Swarm Optimization known as (Binary) Particle Swarm Optimization [14]. In this hybridized evolutionary algorithm, DE mutation and recombination is used to evolve the velocity vectors of each particle in the binary swarm population over a set number of iterations, the local position of each particle in the solution space models a selection of feature descriptors as a discrete bit mask of 0s and 1s. The optimality of each solution is determined by the fitness of each particle in each swarm iteration, which is a function of the root-mean-square-error from the predictive model generated from selected features, in this research the model is a non-linear AdaBoost Random Forest Regression. The matrix of local binary position vectors, representing each particle's proposed feature selection, is used to generate the QSAR model as part of training the AdaBoost Random Forest Regression model as part of the state four.

Stage IV: Train the non-linear AdaBoost Random Forest QSAR model

Inputs: P , where P is the row matrix modeling the selected subset of drug descriptors in the training data set.

Outputs: Q , where Q is a single dimension array list of fitness values computed for each binary particle in the swarm.

Step 1: Use Random Forest Regression to fit the selected descriptor subset set to the target pIC50 training set, producing a non-linear model from the fitted training set.

Step 2: Use the fitted non-linear AdaBoost Random Forest Regression model to predict the experimental target training, test and cross-validation pIC50 values. Adaptive Boosting (AdaBoost) is a support algorithm used in our implementation to help reinforce the collective averaging techniques of the standalone Random Forest Regression ensemble strategy [6]; discussion on both approaches is discussed in sections 4.2 and 4.3. In our research Adapting Boosting, combined with random sampling of standalone Random Forest Regression, was shown to produce more optimal QSAR models for the 37 drug dataset while producing a model for the 91 drug dataset meeting the minimum requirements for use in subsequent drug analysis.

Step 3: Evaluate the fitness of each descriptor subset in the row matrix P .

$$f = \sqrt{\frac{(m_t - n - 1) * RMSE_t^2 + m_v * RMSE_v^2}{m_t - \gamma * n - 1 + m_v}} \quad (6)$$

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

Where m_t, m_v are the total samples in the training and validation sets respectively, n is the total number of drug feature descriptors, $RMSE$ is the root-mean-square error of the predicted pIC50 values for the train and validation sets and γ is a penalty value that supports controlling the balance between models that over-fit and under-fit. From the previous linear QSAR model study on this dataset, $\gamma = 3.3$ was determined to be the optimal value to use for evaluating model fitness [1][26].

Step 4: Predicted pIC50 values are recorded to disk for post-analysis

Step 5: Swarm particle fitness array Q is returned.

Stage V (Deterministic Terminal State): Check for training termination

Inputs: P, Q, n, p_g, t, c , where P is the row matrix of binary particle position vectors, Q is the array list of fitness values computed for the swarm at the current iteration, n is the current iteration, p_g is the current global best position vector in the swarm, t is the maximum number of swarm iterations.

Step 1: Check termination criteria, in this study we end training if $\text{best}(Q)$ has not changed for 30 iterations or if the maximum number of iterations (n) has reached 2000.

Step 2: Identify the local best particle position vector, $\text{best}(p_i)$ row index in P that corresponds to the index of $\text{best}(Q)$.

Step 3: Compare the local best particle position vector with the current global best among the swarm, if $\text{best}(p_{id}) > \text{best}(p_{gd})$, update the swarm's global best position.

Step 4: Increment n for the next iteration and go to **Stage II**

3.3 Validating a QSAR model

When developing QSAR models, in general, it's not enough for a generated model to fit to data it's trained on, the model itself must be statistically validated to ensure its predictive reliability. In the absence of any kind of statistical quality measures, a published QSAR model may fail future validation tests and could not be relied upon as a practical drug screening utility. In order to more thoroughly evaluate a QSAR models explanatory power, external test and validation sets consisting of data samples not used during training should be used as the basis for quality analysis. A popular statistical measure used to determine a QSAR model's predictive quality is the coefficient of determination (R^2) calculated on the training set, and two partitioned subsets known as the validation and external test subsets.

The coefficient of determination (R^2) of each dataset form part of a statistical framework that can be used for performing quality analysis on a generated QSAR model. An additional

statistical measure known as the cross-validated correlation coefficient (Q^2) is used in conjunction with R^2 across each subset to give insight into the statistical reliance of predictions made by a generated QSAR model. A high value of cross-validated correlation coefficient ($Q^2 > 0.5$) and the coefficient of determination for training ($R^2_{train} > 0.6$), test ($R^2_{test} > 0.5$) and validation ($R^2_{validation} > 0.5$) data subsets are generally considered to be much more robust indicators of a QSAR model's predictive reliability [7]**Error! Reference source not found.**

Chapter 4 Models

4.1 Decision Tree Bagging

Decision trees are a popular method of data mining because they remain invariant when operating on data that has undergone scaling and other pre-process normalization and are capable of producing inspect-able, but seldom accurate, models. Decision trees that are grown very deep, for example, tend to learn highly irregular patterns making them more prone to over-fitting a training set by having low bias and high variance [24]. Tree bagging applies to multiple decision trees that are constructed as an ensemble that uses repeated resampling of the training data with out-of-bag replacement estimate [4] being used as a metric for gauging the quality of the constructed decision tree models.

4.2 Tree Bagging to Random Forests

Random forests are a popular ensemble-based machine learning strategy for regression tasks [4] that use multiple decision tree learners (i.e. an ensemble) to collectively predict the value of a target observation with the output being the average of the prediction of each individual tree within the forest. This behavior allows Random Forests to compensate for each individual decision tree's tendency to overfit the training set. Each decision tree in our model takes input in the form of (X, Y) , where Y is the target vector and X is the input vector of feature descriptors, selected by DE-BPSO, to be used for prediction. Using this information, each decision tree regression model acts as an individual meta-estimator that casts a vote for the predicted target observation, the forest then averages the error of the predicted value from the votes cast by each tree within the ensemble [4].

Figure 6. Random Forest Decision Tree Hierarchy [21].

Each tree is grown by considering a random sample from variables of the given input vector X at each nodal split. Sample data at each node is represented as Q , an impurity measure is used as the criterion to determine the variable to be selected at the nodal split. Our non-linear Random Forest Regression model uses the Random Forest algorithm defined by Breiman and Cutler [4] and implemented as part of the Sci-Kit Learn Python library [10], here each regression tree considers all descriptors selected by DE-BPSO and uses the mean square error $H(x_i) = (\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2)$ as the impurity criteria $H(x_i)$. The selected feature x_i at each nodal split is the descriptor that minimizes this impurity. To predict the continuous pIC50 target value, the problem space is partitioned so samples that minimize this impurity are grouped together. The variable (feature) importance is determined by aggregating the total number of occurrences of the feature variable used at each split point within each tree and taking the average of the aggregates among all decision tree members of the ensemble.

Previous research on these datasets [1][30] used a Multiple Linear Regression (MLR) strategy to derive a linear QSAR model which is generally limited to data that is linearly separable. To better handle the higher-dimensional data and potentially less linearly correlated features of our Aryl β -Diketo Acid drug datasets, our approach uses a non-linear Random Forest Regression strategy to obtain a QSAR model that can better generalize to data

that may not be linearly separable. The Random Forest Regression strategy is combined with an Adaptive Boosting support algorithm to reinforce the predictions made by the random sampling averaging strategy of the overall ensemble.

4.3 Improving Random Forests with Adaptive Boosting

A meta-estimator strategy known as Adaptive Boosting (AdaBoost) was applied to support to the non-linear Random Forest Regression model over each training iteration [6]. The concept of boosting is an alternative to the Random Forest Regression model's random sampling and averaging strategy and was combined as a way of hybridizing the machine learning strategy to enhance the predictive quality of generated non-linear QSAR models

The concept behind AdaBoost, in the context of our research, is to sequentially train copies of the Random Forest Regression model on each of the input drug data sets. In each sequential boosting stage a Random Forest copy is dynamically adjusted to focus fitting on values that were incorrectly predicted by a copy in the previous boosting stage [10]. As opposed to Random Forest Regression modeling strategy, which averages prediction errors over all decision trees in the forest, boosting works as a sequence of steps where in each step a set of weights on copies of the base Random Forest Regression estimator are adjusted according a loss function which determines the likelihood the model will be retrained in subsequent boosting. This overall process has the intuitive effect of forcing each subsequent Random Forest copy to focus on more difficult cases, i.e. those that previous copies of the model in the boost sequence failed to accurately predict [6].

The combination of boosting and Random Forest ensemble training through random sampling in the training stage was a departure from previous and collaborative works on optimal QSAR model training using custom MLR [1] and commercial MLR and RFR ensemble strategies [30]. Our approach uses the open-source SciKit Learn implementation AdaBoost RFR which supports mitigating the effects of over-fitting on the Aryl β -Diketo acid drug dataset and was able to converge toward a more optimal non-linear QSAR model with much higher predictive statistical qualities than comparable commercial-based ensemble strategies, see section 4.4.

4.4 Comparing to commercial ensemble methods in previous research

A previous investigation of machine learning used two proprietary ensemble implementations called TreeNet and Random Forests from the commercial SPM data mining suite

[30][33][34] to train non-linear QSAR models on the 91 Aryl β -Diketo acid drug dataset used in this study, an discussion comparing the results of between these two implementations is discussed in section 5.4. TreeNet is a proprietary implementation of the random forest algorithm [4] from Salford Systems [34] that provides a greater degree of predictive accuracy over conventional ensemble bagging and boosting techniques and includes an interactive detection system that helps to improve model performance and discovery of segments of previously unrecognized patterns in training data. Salford Systems proprietary Random Forests [33] implementation is based on the works of Breiman and Cutler [4] that uses ensemble bagging, random sampling and feature importance ranking strategies to build non-linear predictive models.

The SciKit-Learn machine learning library [9] provides free and open-source implementations of the Random Forest ensemble and Adaptive Boosting methods based on the works of Breiman and Cutler [4], and Drucker [6], respectively. The SciKit-Learn implementation of Random Forests is a collective learning strategy that fits an ensemble of independent decision trees to training data using random sampling and feature importance ranking to compose non-linear predictive models. The SciKit-Learn implementation of Adaptive Boosting [6] is a meta estimator that introduces a reinforcement learning component to the underlying non-linear base Random Forest Regressor model by introducing a boost sequence, see section 4.3, which trains subsequent copies of the model on dynamically weighted instances of the feature descriptor training data, weights get are adjusted in each boost step based on the error of the most current predictive model. This reweighting strategy forces subsequent models in each boost step to focus on training features from previous boost cases that resulted in higher predictive error based on a computed average loss for each feature prediction; features exhibiting the most minimal average loss are less likely to be selected by the model in the next boost stage [6].

Chapter 5 Results

5.1 Summary

In this section we focus on optimal models and corresponding results obtained from using hybridized DE-BPSO [1] as a feature eliminator combined with an implementation of non-linear Adaptive Boosting Random Forest Regression model using the open-source machine learning library SciKit-Learn [9]. We show that the SciKit-Learn implementation of AdaBoost Random Forest Regression was able to produce a better QSAR model for the 37 Aryl β -Diketo Acid dataset than a custom DE-BPSO MLR implementation [1]. We also show how the best QSAR models from the open-source implementation of AdaBoost RFR with DE-BPSO compares against QSAR models produced from the commercial SPM TreeNet and MLR [30].

5.2 Datasets

The drug datasets of 37 and 91 dimeric Aryl β -Diketo Acids under analysis (see Figure 7 and Figure 8) were derived from experimental inhibitory activities conducted in a study of QSAR modeling using DE-BPSO MLR [1] that were found to be capable of effectively inhibiting the HIV-1 integrase enzyme controlling the protein strand transfer function crucial to the virus' reproductive cycle. Each dataset was computed using CODESSA [27][28] and consists of constitutional, geometrical, topological, electrostatic and quantum-chemical feature descriptors. The Kennard-Stone algorithm [12] was used to partition the 37 drug dataset into training, validation and test subsets $n_{training} = 23$, $n_{validation} = 7$ and $n_{test} = 7$. The 91 drug descriptor set was partitioned into subsets $n_{training} = 63$, $n_{validation} = 14$ and $n_{test} = 14$. Using this method provides a uniform distribution of descriptors across each subset to help ensure all biological activities are properly tested. All descriptors were subsequently rescaled to be at a mean of zero and a standard deviation of one [1].

Figure 7 Distribution of physiochemical descriptors for 37 Aryl β -Diketo acids drug dataset from DE-BPSO MLR linear QSAR Study [1]

Figure 8 Distribution of physiochemical descriptors for 91 Aryl β -Diketo acids used in commercial MLR, RF and TreeNet study [30]

The regression targets for each dataset represent a sample's $pIC_{50} = -\log(IC_{50})$ value which is a measure of the Aryl β -Diketo Acid compound concentration required to achieve 50% biological activity inhibition. Note that the pIC_{50} is taken as the negated log unit of the IC_{50} value, so higher values will typically indicate exponentially greater potency. A common cut-off for defining potent compounds is a $pIC_{50} \geq 6$ [13].

5.3 Experiment Parameters

Two separate experiments were conducted to gauge the performance of the Sci-Kit Learn implementation of non-linear AdaBoost Random Forest Regression [9] using DE-BPSO as the feature selector. Simulations for these experiments completed 2000 iterations of training using features selected by 50 particles making up the DE-BPSO swarm population; the number of decision trees in the AdaBoost RFR ensemble and total number of boost sequence steps were set to 50 to be equal to the size of the DE-BPSO swarm population. The parameters for our experiment were tuned to be consistent with the configuration of the experimental environment used to obtain an optimal linear QSAR model with DE-BPSO MLR [1], in this study parameters $\beta = 0.004$, a scaling factor $F = 0.7$ and a cross-over rate of $CR = 0.7$ were found to produce the model with the best predictive quality statistics and are used in our experiment as a basis for training the Sci-Kit Learn DE-BPSO AdaBoost Random Forest Regression implementation on the set of 37 and 91 dimeric Aryl β -Diketo Acid datasets.

5.4 Comparing results to previous Non-linear and Linear QSAR Models

Initial results after training a non-linear AdaBoost Random Forest Regression model with features selected by DE-BPSO were very promising. The model with the best predictive and correlative qualities generated by Sci-Kit Learn Non-linear AdaBoost RFR [9] implementation with DE-BPSO is compared to the linear DE-BPSO model and shows a significant improvement for the 37 drug dataset over the previous DE-BPSO MLR implementation [1]. Predictive scores for the larger sample of 91 drug descriptors were shown to be better than those from the commercial SPM TreeNet implementation [30]. Although they were not as optimal as the SPM MLR results, the non-linear models produced from training using AdaBoost RFR with DE-BPSO against this dataset were still able to meet the quality metrics to be considered a reliable model for subsequent analysis [8][7][30].

Table 1 Comparing predictive quality statistics for the best QSAR models from the Sci-Kit Learn Non-linear AdaBoost RFR [9] with DE-BPSO and the DE-BPSO MLR implementation [1] trained on the 37 Aryl β -Diketo Acid dataset

QSAR Model	Q^2	$R^2_{training}$	$R^2_{validation}$	R^2_{test}
Sci-Kit Learn AdaBoost RFR with DE-BPSO	0.808	0.941	0.871	0.754
Custom DE-BPSO MLR	0.783	0.886	0.765	0.722

Table 2 Comparing predictive quality statistics for the best QSAR models from the Sci-Kit Learn Non-linear DE-BPSO AdaBoost RFR [9] and the commercial SPM implementation of MLR and TreeNet [30] trained on the 91 Aryl β -Diketo Acid dataset

QSAR Model	Q^2	$R^2_{training}$	$R^2_{validation}$	R^2_{test}
SPM MLR with DE-BPSO	0.661	0.896	0.842	0.809
Sci-Kit Learn AdaBoost RFR with DE-BPSO	0.604	0.942	0.540	0.583
SPM TreeNet	Not reported	0.648	Not reported	0.475
SPM RFR	Not reported	Not reported	Not reported	0.430

Table's 1 and 2 show the predictive quality statistics of the best QSAR model generated by the SciKit-Learn implementation of non-linear AdaBoost Random Forest Regression to the optimal QSAR models produced from a custom implementation of DE-BPSO MLR, the commercial SPM MLR with DE-BPSO and TreeNet, respectively [1][9][30]. The cross-validated correlation coefficient (Q^2) and correlation coefficient (R^2) statistics together summarize the overall predictive quality of the SciKit-Learn non-linear AdaBoost RFR model against our experimental 37 and 91 drug datasets. For both (R^2) and (Q^2), a value closer to 1 is typically indicative of a predictive QSAR model with very reliable statistical qualities.

Table 3 Comparing the root-mean-square-error and standard error of the estimate for the best QSAR models from the Sci-Kit Learn Non-linear DE-BPSO AdaBoost RFR [9] and the custom DE-BPSO MLR implementation [1] trained on the 37 Aryl β -Diketo Acid dataset

QSAR Model	SEE	$RMSE_{training}$	$RMSE_{validation}$	$RMSE_{test}$
Sci-Kit Learn AdaBoost RFR with DE-BPSO	0.228	0.208	0.314	0.421
Custom DE-BPSO MLR	Not reported	Not reported	Not reported	Not reported

Table 4 Comparing the root-mean-square-error and standard error of the estimate for the best QSAR models from the Sci-Kit Learn Non-linear DE-BPSO AdaBoost RFR [9] and the commercial SPM implementation of MLR and TreeNet [30] trained on the 91 Aryl β -Diketo Acid dataset

QSAR Model	SEE	RMSE _{training}	RMSE _{validation}	RMSE _{test}
Sci-Kit Learn DE-BPSO AdaBoost RFR with DE-BPSO	0.177	0.171	0.516	0.446
SPM TreeNet	Not reported	MSE = .345 RMSE = 0.587	Not reported	MSE = .345 RMSE = 0.587
SPM RFR	Not reported	MSE = .412 RMSE = 0.766	Not reported	MSE = .412 RMSE = 0.766
SPM MLR with DE-BPSO	Not reported	0.3729	0.1210	.0953

Table's 3 and 4 show the standard error of estimate (SEE) and root-mean-square error (RMSE) of the best QSAR model generated by the SciKit-Learn implementation of non-linear AdaBoost Random Forest Regression to the optimal QSAR models produced from a custom implementation of DE-BPSO MLR and SPM TreeNet, respectively [1][9][30]. Both the SEE and RMSE are general purpose statistical quality measures, the standard error of estimation is a measure that helps us determine how far spread experimental pIC50 target values are from the underlying regression model. The RMSE serves as a way of assessing the errors between a model's predicted pIC50 values and the observed experimental pIC50 target regression values. For both statistics, the goal is to obtain a value that's as close to zero as possible, indicating the error between predicted and actual target values is minimal.

5.5 Discussion of optimal Non-Linear QSAR models

In tables 5 and 6 below, we outline the feature importance reported by the optimal QSAR model trained with the Sci-Kit Learn Non-linear AdaBoost RFR [9] with DE-BPSO [1] on both the 37 and 91 dimeric Aryl β -Diketo Acid drug datasets, respectively. Because Random Forest Regression is a non-linear modeling strategy, we cannot express the optimal model as a linear combination of features with corresponding coefficients like the linear models such those produced by the custom DE-BPSO MLR [1] and commercial SPM MLR [30] implementations. Instead, like the non-linear SPM TreeNet model [30], the output from the

optimal Sci-Kit Learn AdaBoost RFR is expressed as the feature variable's importance determined to have greatest impact on the experimental target pIC₅₀ values.

Table 5 Feature descriptors and corresponding importance for the best QSAR model trained from the Sci-Kit Learn Non-linear AdaBoost RFR [9] on the non-linear 37 Aryl β -Diketo Acid dataset

Descriptor	Feature Importance
Number of aromatic bonds	0.432
Average electrophilic reaction index for a O atom	0.367
Minimum electrophilic reaction index for a C atom	0.201

Table 6 Feature descriptors and corresponding importance for the best QSAR model trained from the Sci-Kit Learn Non-linear AdaBoost RFR [9] on the non-linear 91 Aryl β -Diketo Acid dataset

Descriptor	Feature Importance
HA dependent HDCA-2/SQRT(TMSA) [Zefirov's PC]	0.411
Kier and Hall index (order 2)	0.314
Relative number of benzene rings	0.275

For each of the 37 and 91 drug datasets, the optimal models from the Sci-Kit Learn AdaBoost RFR [9] with DE-BPSO consisted of 50 decision tree estimators. The mean square error prediction metric was used as the Gini impurity criterion [4] for gauging the quality of a split within each decision tree in the Random Forest Regression ensemble. Each of our optimal non-linear QSAR models, using SciKit Learn AdaBoost RFR and DE-BPSO, considers all features selected by the DE-BPSO algorithm [1] when looking for the best split based on this impurity metric. The maximum depth of each tree was expanded until all leaves contained less than the minimum number of samples required to split internal nodes of each ensemble member during fitting. For our study, optimal QSAR models from SciKit Learn's AdaBoost RFR determined the minimum number of samples required to split internal nodes to be 2, while a minimum of 1 samples was required in newly created leaves, any leaf nodes containing 0 samples are discarded by the models.

Our studies optimal non-linear QSAR models use the Sci-Kit Learn AdaBoost RFR implementation to determine feature importance rankings from fitted data by averaging the

occurrences of each Aryl β -Diketo Acid feature descriptor selected by DE-BPSO and used most within split points of each individual decision tree member of the RFR ensemble when fitting to the training data. Members of the RFR decision tree ensemble naturally perform feature selection as part of model training by taking the average of the aggregate of feature occurrences used at each nodal split within each decision tree in the ensemble, see section 4.2.

The best average scores for each independent feature descriptor sample are a measure of the importance of the selected feature descriptor in Random Forest Regression with larger ranking scores indicating features determined by the Non-linear AdaBoost RFR [9] model to be more significant toward affecting inhibition of HIV-1 integrase.

The AdaBoost [9] component of the model optimal model uses a 50 step boosting sequence consistent with the population size of the DE-BPSO algorithm and Random Forest Regression model ensemble. In each step, boosting is applied by fitting an instance of the non-linear Random Forest Regression model to a weighted copy of the input training data, an average loss is computed to determine the level of predictive confidence for the copy of the Random Forest Regression model instance on the weighted training data. This loss represents the model instance's prediction error and is used to adjust the weights of the training data for subsequent copies of the Random Forest Regression model in each step of the boost, the smaller the predictive error the less likely the training sample will be included in subsequent boosting steps. The AdaBoost strategy forces the Random Forest Regression model to retrain on samples that were incorrectly predicted by previous instances in the boost. The boost sequence terminates after the maximum number of steps has been reached or if the average prediction error drops below a default .5 threshold [6].

Figure 9. Regression plot for DE-BPSO 37 Aryl β -Diketo Acid dataset

Figure 10. Regression plot for 91 Aryl β -Diketo Acid dataset

In figures 5 and 6, the regression plots for each demonstrates the statistical quality of the best non-linear QSAR model trained using AdaBoost RFR with DE-BPSO by showing the regression line between predicted and observed pIC₅₀ values across the training, validation and test subsets; the regression line was plotted using an online charting tool called Plotly

[32]. Points closer to the regression line across the training, validation and test subsets show the predictive quality of our best models.

Based on the results of comparing the linear and non-linear QSAR model's generated by SciKit Learn AdaBoost RFR [9] with DE-BPSO [1], respectively, the non-linear model trained with AdaBoostRFR with DE-BPSO presents much better predictive qualities on the 37 drug data set than linear MLR with DE-BPSO[1]. This suggests a non-linear AdaBoost Random Forest QSAR model could find a more accurate correlation of features between selected descriptors and the target inhibitory response quantified by the experimental pIC50 target values.

From results reported by our optimal non-linear models using SciKit Learn AdaBoost RFR [9], we see the number of aromatic bonds, a minimum electrophilic reaction index for C atoms and an average electrophilic reaction index for O atoms are molecular descriptors with the most influence on the biological inhibitory activities of Aryl β -Diketo Acids. As previously described, linear models from the commercial SPM MLR used in previous research [30] had better predictive quality statistics on the 91 drug data set, however non-linear models from its proprietary Random Forests and TreeNet ensemble strategies [33][34] did not perform as well as our non-linear models based on the open-source SciKit Learn AdaBoost and RFR strategy using DE-BPSO. Our optimal non-linear model determined the relative number of benzene rings, a Kier and Hall valance connectivity index and the HA dependent HDCA-2/SQRT (TMSA) [Zefirov's PC] or topographic index as being the most relevant descriptors. In each of the 37 and 91 drug sample simulations, the total number of individual decision tree estimators in the non-linear AdaBoost Random Forest QSAR model was set to be equal to the number of particles in the binary swarm population, since the non-linear RF model was able to outperform the linear MLR model on the smaller 37 drug dataset, an AdaBoost Random Forest tuned with a larger ensemble of estimators may be able to perform better on larger datasets than the linear MLR model.

Chapter 6 The QSAR Distributed Web Modeling Framework

6.1 Motivation

The CSUSM QSAR Web Modeling System is a prototype system designed to be a scalable web application built on a popular web framework [18] to support training quantitative

structure-activity relationship (QSAR) models using a distributed task framework built on an application-level message-passing architecture. The goal was to design a prototype framework that could be used to perform QSAR training simulations in parallel to allow analysis of multiple models at once. The system was designed to be extended to provide support for future drug discovery research.

6.2 Design and Development

The middle-ware distributable framework and front-end user interface were developed and implemented independently in collaboration with a separate study of optimal QSAR modeling in drug design [30]. This was done to help support the more rapid development of each component of the system. This research contributed the implementation of the middle-ware application program interface (API) and web front-end user interface components that interact with a database backend for persisting QSAR model training sessions and results for retrieval by one or multiple client sessions running from a standard web browser. The decoupled nature of each component helped ensure ease of maintainability and expandability to support future research efforts.

6.3 Components and Architecture

The CSUSM QSAR Web Modeling System is a Model-View-Controller (MVC) web-based application framework that offers a decoupled and scalable architecture. This separation of concerns allowed different components of the system to be developed and tested independently to make it easier to incorporate different evolutionary algorithm feature selection implementations and QSAR machine learning models.

QSAR Web Modelling System Architecture

Figure 11. QSAR Web Modeling (MVC) architecture

A. *Model*

Components contained at the Model-layer provide core logic and persistence support implemented as part of a separate collaborative research effort [30] that used an open-source relational database [16] for storing model configuration, result and session data for reference by users executing QSAR model training tasks. Model configuration data is used to instantiate multiple instances of worker processes capable of training many QSAR models in an asynchronous-parallel manner. The database schema was implemented by research collaborators to provide a relational-data persistence layer to store QSAR modeling information and results produced from models trained by clients using the middle layer API. The ability to decouple middle layer logic from the backend data model made it easy to persist and retrieve results and configuration information allowing our research to run simulations in parallel.

B. *View*

View-layer components provide user interface controls and the ability to interactively track the status of a QSAR model within the context of a user-defined training session. The view layer is responsible for presenting

responsive and dynamic web browser-based user interface that's designed to be executed from any system with an internet connection and a standard web browser. The flexibility of the web browser also opens up the possibility of easily expanding the user interface to include additional reporting and analysis features to support future research. This separation of concerns between the model logic and user interface presentation layers allowed each component to be tested and extended independently. The web browser as a general user interface also provides clients the ability to retrieve results archived in the database stored in the system architectures model layer.

Figure 12. QSAR Web landing page

Figure 13. Configuring a model as part of a training session

C. Controller

The Controller-layer contains middle layer components that delegate communication between the user and QSAR modeling tasks at the model layer. At its core, the middle layer is an application program interface (API) that incorporates a scalable message broker that controls distribution of training tasks among worker processes using application-level message passing [17]. Distributable computing is achieved using queue data structures to store QSAR model training as individual units of work known as tasks. This helps avoid blocking a single web server's process which would normally require it to full-full both client requests and process QSAR training tasks in parallel, this would reduce a traditional web client-server architecture's response time significantly. Instead, training tasks are encapsulated as worker units which are queued and distributed by a message broker [17] controlling independent worker processes on the same server as the middle layer API controller process, reducing the overhead of the main middle layer process and allowing the system more easily scale to multiple clients. The distributable architecture of the middle layer controller components are also extendible to include multiple server systems working collaboratively as part of a distributed cluster.

6.4 Advancing Distributable QSAR Modeling in Drug Design

The QSAR Web modeling system's primary goal is to help advance the drug design researcher's ability to generate and analyze multiple QSAR models in a distributed and easily scalable environment. Leveraging the QSAR web modeling system allowed our research to better compare predictive models generated from several training sessions running on independent, distributable worker processes. This system provides a common web browser-based user interface for researchers to conduct QSAR model training tasks as one or more user sessions run on a scalable, backend processing architecture. Each session is persisted to the supported database model [30] and can be easily restored through the web browser by referencing a unique session identifier in the QSAR web modeling URL. This URL identifier encapsulates the context of a QSAR model training task along with corresponding results associated with models generated from running the tasks on a distributable background worker process.

QSAR model training tasks are processed into worker queues on one more dedicated backend servers that work separate from the web server controller to allow it to continue to process multiple client requests in parallel with one or more training tasks running independently in the background. This method of distributing units of work among multiple processes provides a more responsive interface to allow, for example, our researchers to run several independent models through different web browser instances. The unique QSAR training session identifier also makes it more convenient to correlate generated results with specific feature selection and machine learning strategies which are encapsulated as user session configuration metadata.

Figure 14. Distribution of QSAR training models into task queues monitored by separate worker processes

In figure 10 above, we show a high-level illustration of the distributable architecture of the QSAR web modeling system. Using the concept of a task queue and distributed worker process pool [31] allows multiple QSAR models to be trained in parallel by many different users without interfering with the primary web user interface or middle layer API.

The QSAR web modeling system uses an application-level messaging protocol [17] to provide the facilities for receiving status from QSAR training task workers in real-time. Each training session's status and results are retrieved from the database model, implemented in a separate collaborative research effort [30][16], using a queue-based publisher-subscriber message passing system which was incorporated to facilitate communication between users and dedicated worker processes performing user session training tasks in the background. This provides researchers using the QSAR web modeling system the ability to determine the status of any running models at any time and also allows each worker process to report the result of a QSAR model back to the front-end user for retrieval and post-analysis.

Figure 15. Application-level message passing between worker processes

6.5 Further Development

Although the QSAR Web Modeling system was in an early prototype phase at the time this study was conducted, the system was made available as an open-source project [15] with the intention of helping promote open collaboration and to streamline the development to help support current and future research. The primary goal of the project is to build support for additional scaling capability to support current non-linear AdaBoost Random Forest QSAR model implementations, as well as provide a basis for incorporating new models in the future.

The distributed aspect of the framework will provide greater QSAR modeling capabilities beyond the means of a standalone supercomputer. This architecture provides an open framework to extend and incorporate more computationally intense models as part of future QSAR modeling research efforts and helps avoid the need for reliance on more expensive computer systems by harnessing the power of less expensive commodity hardware.

Chapter 7 Conclusion

7.1 Summary and Future work

In this study, we used hybridized DE-BPSO [1] as a feature selection method combined with an implementation of non-linear Adaptive Boosting Random Forest Regression model using the open-source machine learning library SciKit-Learn [9] to develop an optimal non-linear QSAR model for the analysis of Aryl β -Diketo Acids targeting the inhibition of HIV-1 integrase protein enzyme by identifying the physiochemical molecular descriptors that exhibit the greatest influence on the crystallization of the integrase enzyme's binding mechanism.

An experiment was run on two sets of 37 and 91 dimeric Aryl β -Diketo Acids each partitioned into a training, validation and test sets using the Kennard-Stone algorithm [12]. Each partitioned data set represented a collection of geometrical, topological, constitutional, electrostatic, and quantum-chemical descriptor compounds.

We then compared linear models from two independent investigations of the 37 and 91 drug datasets, respectively. For the first independent study on the 37 drug dataset, which used a custom implementation of DE-BPSO MLR [1], the Sci-Kit Learn AdaBoost RFR with DE-BPSO was shown to be more effective in training a non-linear model with better predictive qualities than the DE-BPSO MLR QSAR model. A second independent study used the commercial data mining tool SPM MLR with DE-BPSO and TreeNet [30] to generate optimal linear and non-linear models from the 91 drug datasets. In this dataset, AdaBoost RFR with DE-BPSO showed lower predictive qualities over the commercial SPM MLR with DE-BPSO, but exhibited better statistical qualities than the commercial TreeNet and RFR implementations and still meets the requirements for subsequent analysis [1][7]**Error!**

Reference source not found.

From results produced by our non-linear AdaBoost Random Forest QSAR model implementation, the descriptors with the greatest influence on the biological activity of β -

diketo acids are those which are related to molecular volume, topology, and electrostatic effects, with large molecular volumes having the greatest impact.

The AdaBoost RFR DE-BPSO model found the three most important features of the 37 drug data set to be the number of aromatic bonds, a minimum electrophoresis reaction index for C atoms and an average electrophoresis reaction index for O atoms to be the most significant descriptors for indication of inhibition of the HIV-1 integrase enzyme. The total number of aromatic bonds refers to the double bonds in flat rings structure of molecular atoms, which remain highly stable during a chemical reaction. The average and minimum electrophilic reaction indexes for oxygen and carbon atoms, respectively, refer to the quantum mechanical interactions between these molecules. The results produced are very encouraging since the HIV-1 integrase active site is considered to be hydrophobic in nature with a single Mg^{2+} ion available for chelation with a drug compound for enzymatic inhibition [25].

For the 91 drug data set, the relative number of benzene rings, Kier and Hall index (order 2) and HA dependent HDCA-2/SQRT (TMSA) [Zefirov's PC] were considered the most significant descriptors. The Kier and Hall index is a topological feature referring to the valence connectivity between atoms. The HA dependent HDCA-2/SQRT (TMSA) [Zefirov's PC] is an electrostatic descriptor used to quantify the electric charge distribution and electronic density of a molecule. Although predictive scores were lower compared to the linear MLR QSAR model, the resulting non-linear AdaBoost RFR was still able to produce a model fitting the minimum requirements to be considered predictive. Since the non-linear AdaBoost Random Forest QSAR model with an ensemble population of 50 decision trees was able to outperform the linear MLR QSAR model on the 37 drug dataset, future works will focus on running simulations of non-linear AdaBoost Random Forests with a larger ensemble population on a larger distributed web modeling framework cluster to determine if adding more boosted decision tree learners will allow the non-linear Random Forest to more effectively scale to even larger datasets.

7.2 Research Contribution

For this research, a middle layer distribution framework and web browser user interface component for a prototype distributed web modeling framework were contributed as part of an effort to support running multiple QSAR model training simulations in parallel [15]. In future research, we plan to expand this distributable system by integrating additional feature

selection and machine learning models that can be trained in parallel to be able to compare qualities of multiple QSAR models trained on larger datasets to provide support for the study of even more effective HIV-1 inhibitors and their interactions with the biological activity of their HIV-1 retrovirus targets.

To help support storing results and QSAR model configurations for a middle layer interface, a database developed by research collaborators was used to provide support for extending the distributable QSAR web modeling system as part of support for future research efforts of different configurations of DE-BPSO AdaBoost Random Forest Regression as well as extending support for other implementations of non-linear machine learning.

The database schema was designed as a separate component of the middle layer which made the system easier to maintain, test and improve by avoiding tight coupling of the middle layer and model layer components. Middle layer components provide an easily extendible API for clients interfacing from a standard web browser and allow computationally intensive QSAR modeling tasks to be distributed and run in parallel when executed using the training API from the web browser user interface. Results from one or more training tasks are stored in the database to make them easily accessible for download through the web browser user interface. For this research, the database for the web modeling system provided support to store and persist results produced from both the 37 and 91 Aryl β -Diketo Acids datasets for future analysis and also offers a platform for executing multiple QSAR model training tasks under a distributable computation framework that can be extended to support research and investigation of drugs targeting the inhibition of HIV-1 integrase enzymes.

Chapter 8 References

- [1] Ko, G., & Reddy, A. (2012). Differential Evolution-Binary Particle Swarm Optimization Algorithm for the Analysis of Aryl /3-Diketo Acids for HIV-1 Integrase Inhibition. IEEE Congress on Evolutionary Computation.
- [2] Global report: UNAIDS report on the global AIDS epidemic 2013. (2013, November 1). Retrieved from <http://www.unaids.org/en/resources/campaigns/globalreport2013/globalreport>
- [3] De Clercq, E. (n.d.). Anti-HIV Drugs: 25 Compounds Approved Within 25 Years After The Discovery Of HIV. International Journal of Antimicrobial Agents, 307-320.
- [4] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.
- [5] Cordeiro, N., & Luan, F. (2009). Overview of QSAR Modeling in Rational Drug Design. International Journal of Antimicrobial Agents, 33(4), 307-320.
- [6] Drucker, H. (1997). Improving Regressors Using Boosting Techniques. Proceedings of the Fourteenth International Conference on Machine Learning, 107-115.
- [7] Tropsha, A., Gramatica, P., & Gombar, V. (n.d.). The Importance of Being Earnest: Validation is the Absolute Essential for Successful Application and Interpretation of QSPR Models. QSAR & Combinatorial Science, 69-77.
- [8] Golbraikh, A., & Tropsha, A. (n.d.). Beware of q²! Journal of Molecular Graphics and Modeling, 20(4), 269-276.
- [9] Pedregosa, F. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 2825-2830.
- [10] Freund, Y. (n.d.). A Decision-Theoretic Generalization Of On-Line Learning And An Application To Boosting,. Journal of Computer and System Sciences, 55(1), 119-139.
- [11] Grömping, U. (n.d.). Variable Importance Assessment in Regression: Linear Regression versus Random Forest. The American Statistician, 63(4), 308-319.
- [12] Kennard, R., & Stone, L. (n.d.). Computer Aided Design of Experiments. Technometrics, 11(1), 137-148.

- [13] What Is PIC50 or PAC50? Center for Chemical Genomics. University of Michigan Life Science Institute, 29 Oct. 2010. Web. 04 Mar. 2015.
http://mscreen.lsi.umich.edu/mscreenwiki/index.php/What_is_pIC50_or_pAC50%3F
- [14] Shen, Q., Jiang, J., Jiao, C., Shen, G., & Yu, R. (n.d.). Modified particle swarm optimization algorithm for variable selection in MLR and PLS modeling: QSAR studies of antagonism of angiotensin II antagonists. *European Journal of Pharmaceutical Sciences*, 145-152.
- [15] Galvan, R. (2014, November 1). Retrieved from
<https://github.com/kingrichard2005/qsarweb-public>
- [16] PostgreSQL: The world's most advanced open source database. (1995, May 1). Retrieved March 11, 2015, from <http://www.postgresql.org/>
- [17] RabbitMQ - Messaging that just works. (n.d.). Retrieved from
<http://www.rabbitmq.com/>
- [18] Django The web framework for perfectionists with deadlines. (2005, July 1). Retrieved from <https://djangoproject.com/>
- [19] Storn, Rainer, and Kenneth Price. Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization* 11 (1997). Print.
- [20] Wang, J. (2005). *Data mining in Bioinformatics*. London: Springer.
- [21] De Freitas, N. (2013, January 1). CPSC 540: Machine learning. Retrieved from <http://www.cs.ubc.ca/~nando/540-2013/lectures.html>
- [22] Zhang, C., Ning, J., Lu, S., Ouyang, D., & Ding, T. (n.d.). A novel hybrid differential evolution and particle swarm optimization algorithm for unconstrained optimization. *Operations Research Letters*, 117-122.
- [23] Shafer, R., & Shapiro, J. (2009). HIV-1 drug resistance mutations: An updated framework for the second decade of HAART. *US National Library of Medicine National Institutes of Health*, 10(2), 67-84.
- [24] Hastie, T., & Tibshirani, R. (2009). *The Elements of Statistical Learning*. Dordrecht: Springer.
- [25] Chen, X., Tsiang, M., Yu, F., Hung, M., Jones, G., Zeynalzadegan, A., Chen, J. (n.d.). Modeling, Analysis, and Validation of a Novel HIV Integrase Structure Provide Insights into the Binding Modes of Potent Integrase Inhibitors. *Journal of Molecular Biology*, 504-519.

- [26] Depczynski, U., Frost, V., & Molt, K. (n.d.). Genetic algorithms applied to the selection of factors in principal component regression. *Analytica Chimica Acta*, 217-227.
- [27] CODESSA version 2.7.16. Shawnee, Kansas: Semichem, Inc., 2009
- [28] The CODESSA PRO project. (n.d.). Retrieved from <http://www.codessa-pro.com/>
- [29] Agrafiotis, D., & Cedeño, W. (n.d.). Feature Selection for Structure–Activity Correlation Using Binary Particle Swarms. *Journal of Medicinal Chemistry*, 1098-1107.
- [30] Moghaddam, M. (2015). Development and comparison of Linear and Non-linear (QSAR) Models To predict HIV-1 Integrase Inhibitors using Hybrid Differential Evolution-Binary Particle Swarm Optimization Algorithm as a feature selection method.
- [31] Celery: Distributed Task Queue. (n.d.). Retrieved from <http://www.celeryproject.org/>
- [32] Collaborative data science. (n.d.). Retrieved from <https://plot.ly/>
- [33] Random Forests® - Salford Systems - Data Mining and Predictive Analysis Software. (n.d.). Retrieved from <http://www.salford-systems.com/products/randomforests>
- [34] TreeNet® - Salford Systems - Data Mining and Predictive Analysis Software. (n.d.). Retrieved from <http://www.salford-systems.com/products/treenet>
- [35] The mechanism of retroviral DNA integration through crystal structures of its key intermediates. (n.d.). from <http://www.diamond.ac.uk/Science/Research/Highlights/casestudy55.html>
- [36] On the Front Lines of Drug Discovery Using the Computational Chemistry Approach. (n.d.). from <http://torrent.cms-initiative.jp/torrent-no.8/on-the-front-lines-of-drug-discovery-using-the-computational-chemistry-approach>