Abstract Quantitative Structure-Activity Relationship (QSAR) models have been successfully applied to lead optimisation, virtual screening and other areas of drug discovery over the years. Recent studies, however, have focused on the development of models that are predictive but often not interpretable. In this article, we propose the application of a piecewise linear regression algorithm, OPLRAreg, to develop both predictive and interpretable QSAR models. The algorithm determines a feature to best separate the data into regions and identifies linear equations to predict the outcome variable in each region. A regularisation term is introduced to prevent overfitting problems and implicitly selects the most informative features. As OPLRAreg is based on mathematical programming, a flexible and transparent representation for optimisation problems, the algorithm also permits customised constraints to be easily added to the model. The proposed algorithm is presented as a more interpretable alternative to other commonly used machine learning algorithms and has shown comparable predictive accuracy to Random Forest, Support Vector Machine and Random Generalised Linear Model on tests with five QSAR data sets compiled from the ChEMBL database.

GUIDE

Optimal Piecewise Linear Regression algorithm for QSAR Modelling

Jonathan Cardoso-Silva[1], George Papadatos[2], Lazaros G. Papageorgiou[3], Sophia Tsoka[1,*]

1. Introduction

Quantitative Structure-Activity Relationships (QSAR) are mathematical models that aim to predict biological activity of chemical compounds based on their molecular structure [1]. These models are particularly useful for drug discovery as they can be used to draw hypotheses from the data. QSAR models are often used in virtual screening to help identify new potent compounds for a target of interest [2] or to re-purpose existing medicines to different treatments [3]. The technique can also indicate optimisation strategies to develop potent new drugs from a series of promising compounds [4].

The first QSAR models were built for small series of similar compounds using only a few quantitative features and aimed to discover a transparent relationship, preferably linear, between molecular structure and biological activity [5]. Although this approach is still employed to design new drugs [6, 7], most recent studies propose models that consist of hundreds or thousands of molecular descriptors calculated from the 2D or 3D representations of molecules [8–11] and are often built with non-linear algorithms such as Neural Networks, Support Vector Machines with Gaussian kernels and Random Forests [12]. These techniques usually predict the biological activity of compounds with better accuracy than linear methods, but they are often described as "black box", i.e. the relation between chemical features and biological activity can not be obtained directly from the outcome of the algorithm [13]. Even when it is possible to obtain a ranking or importance of features, as is the case with Random Forest and its out-of-bag feature importance estimation, it is hard to identify a clear relationship between the properties of a molecule and its biological activity. Recent studies have been proposed to reverse engineer the predictions made by "black box" algorithms [14–17] but in this study we argue that it is possible to produce accurate yet interpretable QSAR models directly, without the need of a post-processing step in the form of equations linking features to outcome.

The selection of a subset of features that is most relevant to the prediction problem is an important strategy towards more interpretable QSAR models. A modeller can select

[1] Department of Informatics, Faculty of Natural and Mathematical Sciences, King's College London, Bush House 30 Aldwych, WC2B 4BG, London, United Kingdom. [2] European Molecular Biology Laboratory European Bioinformatics Institute, Wellcome Genome Campus, Hinxton, Cambridgeshire. Current affiliation: GlaxoSmithKline, Gunnels Wood Road, Stevenage, Hertfordshire, SG1 2NY, United Kingdom. [3] Centre for Process Systems Engineering, Department of Chemical Engineering, University College London, Roberts Building, Torrington Place, WC1E 7JE, London, United Kingdom.
[*] Corresponding author: e-mail: sophia.tsoka@kcl.ac.uk

Table 1: Data sets used in this study

Data Set	Biological Endpoint	Source	Samples	Descriptors
hDHFR	human dihydrofolate reductase	CHEMBL202	542	76
rDHFR	rat dihydrofolate reductase	CHEMBL2363	875	80
CHRM3	human muscarinic acetylcholine receptor M3	CHEMBL245	588	87
NPYR1	human neuropeptide Y receptor type 1	CHEMBL4777	354	70
NPYR2	human neuropeptide Y receptor type 2	CHEMBL4018	374	67

these features empirically, according to existing hypotheses or known properties about the compounds in a dataset. However, such an approach does not work well in practice, particularly when mining data in public repositories that have been collected from multiple sources, or where the relevant features might not be known beforehand. In these cases, the selection of most important features is often delegated to the algorithm [13, 18]. A common technique that has been used for feature selection is Principal Component Analysis (PCA) [19–21] however, as dimensionality reduction is achieved through transformation of the original data, a post-processing step is required in order to express the effects of each individual feature to the prediction outcome. Algorithms exist that do not rely on data transformation (e.g. based on genetic algorithms [7], particle swarm optimisation [22] and regularisation strategies [23]) and can perform feature selection prior to QSAR modelling itself. We note that an algorithm performing feature selection simultaneously to QSAR modelling is desirable. Finally, feature selection by itself is not sufficient to otain an interpretable model. It is also important to use an algorithm that can transparently relate the chemical properties of a compound to biological activity, while also being able to account for non-linearities inherent to the data.

In this article, we propose a novel computational strategy for activity prediction, incorporating feature selection and employing a mathematically descriptive basis. Our proposed algorithm, Optimal Piecewise Linear Regression Algorithm with Regularisation (OPLRAreg), identifies different regions in the data and linear equations to describe each of these segments while incorporating an explicit feature selection with regularisation. OPLRAreg models QSAR problems using mathematical programming, a standard representation of optimisation problems that can be solved using exact algorithms and can be easily adjusted by the addition of custom constraints [24–26].

The OPLRAreg algorithm was implemented to predict the inhibitory concentration ($logIC_{50}$) of compounds in data sets compiled from ChEMBL [27]. Best practices in QSAR modelling were followed for data cleaning, preprocessing and rigorous validation [1, 28]. Below, we demonstrate the effect of regularisation in prediction accuracy and dimensionality reduction, illustrate how the proposed algorithm could be easily modified to accommodate custom constraints of a QSAR project and compare the results with other machine learning algorithms in R package *caret* [29] version 6.0-76 (2017).

2. Methods

2.1. Data Sets

We have obtained five data sets from ChEMBL database (version 22_1) [27]. Each data set contains a list of chemical compounds with their respective binding activity to a protein target, measured by $pIC50 = -log_{10}(IC_{50})$. The same data sets were used to benchmark algorithms in [12]; here we obtained an updated list from ChEMBL and performed a preprocessing step to remove invalid and duplicated compounds. First, we selected the entries with IC_{50} measurements and filtered out compounds with dubious measurements, indicated by column DATA_VALIDITY_COMMENT. For groups of duplicated records, if the standard deviation of activity was above 1 log unit, $sd > 1$, these compound samples were removed from the data set; otherwise, a single entry with the median of the activity was kept.

Java Chemistry Development Kit (CDK) (version 1.5.13) [30] was used to calculate 1D and 2D molecular descriptors, totalling more than 200 numerical descriptors for the chemical compounds in each data set. These features were cleaned and normalized following practice described in Tsiliki et al 2015 [28]. Data were normalized and molecular descriptors with near zero variance and highly correlated features were removed using the R package *caret* [29]. Details for data sets after this preprocessing step are given in Table 1.

2.2. New mixed integer programming model

A piecewise linear regression algorithm based on mathematical programming was introduced in [31]. Optimal Piecewise Linear Regression Algorithm (OPLRA) solves Mixed Integer Programming (MIP) models to find partitions in the data where the outcome of samples is predicted by unique linear equations identified for each disjoint region. The algorithm contains a loop defined over all features in the data set where MIPs are solved for two regions ($R = 2$), and the feature leading to the smallest error in prediction across all samples is taken as the partition feature (f^*) for subsequent iterations. The number of regions is then increased at each iteration until the improvement in prediction error is smaller than a user-defined paramater between iterations.

Although OPLRA has been successfully applied to UCI benchmark data sets [31], it did not perform well when applied to QSAR models. The regression coefficients identified by the algorithm fit samples in the training set well, but had poor performance on the test set, indicating the effect of overfitting. To mitigate these problems, the objective function in OPLRA was modified to include two terms; mean absolute error (*MAE*), a well established metric for regression analysis in QSAR [32], and a ℓ_1 regularisation term (*REG*), calculated as the sum of all absolute regression coefficients. The regularisation term reduces the risk of generating linear equations that are too specific to the training set. The new objective function accounts for both accuracy and complexity of the models generated and is shown in Equation 1 below.

$$z = MAE + \lambda \; REG, \qquad (1)$$

where λ is a positive user-defined parameter that controls the influence of regularisation.

Variables *MAE* and *REG* are defined by the set of equations below:

$$MAE = \frac{\sum_s E_s}{|s|}, \qquad (2)$$

$$REG = \sum_f W^+_{rf}, \qquad (3)$$

$$W^+_{rf} \geq \quad W_{rf} \qquad \forall r, f \qquad (4)$$

$$W^+_{rf} \geq -W_{rf} \qquad \forall r, f \qquad (5)$$

where E_s indicates the absolute error for each sample s and $|s|$ is the number of samples in the training set. Positive variables W^+_{rf} are introduced to indicate the absolute value of regression coefficients W_{rf} and are defined by the two auxiliary constraints above.

At every iteration, the number of regions R and the partition feature f^* used to identify breakpoints are fixed. The allocation of sample s to regions $r \in \{1, 2, \ldots, R\}$ is modelled with binary variables F_{sr} while the breakpoints are represented by the free variables $X_{r,f}$, where f always corresponds to the partition feature f^* of the current iteration.

Equation 6 guarantees that a sample can belong to only one region:

$$\sum_r F_{sr} = 1 \quad \forall s, \qquad (6)$$

while Equation 7 below ensures that breakpoints are consistent:

$$X_{r,f^*} \geq X_{r-1,f^*}, \quad \forall r = 2, 3, \ldots, R-1. \qquad (7)$$

Equations 8 and 9 assign samples to the correct regions according to the breakpoints.

$$A_{sf^*} \geq X_{r-1,f^*} - U\;(1 - F_{sr}) \quad \forall s, r = 2, 3, \ldots, R, \qquad (8)$$

$$A_{sf^*} \leq X_{r,f^*} - U\;(1 - F_{sr}) \quad \forall s, r = 1, 2, \ldots, R-1, \qquad (9)$$

The predicted value P_{sr} for sample s in region r is given by Equation 10, according to regression coefficients W_{rf} and the intercept B_r for each region. Equations 11 and 12 compute the absolute error in prediction E_s for each sample. O_s are the observed values for sample s and U is a large number that will force these constraints to consider only the predicted values P_{sr}, where sample s belongs to region r, $F_{sr} = 1$.

$$P_{sr} = \left(\sum_f W_{rf} A_{sf}\right) + B_r \qquad \forall s, r, \qquad (10)$$

$$E_s \geq O_s - P_{sr} - U\;(1 - F_{sr}), \qquad \forall s, r, \qquad (11)$$

$$E_s \geq P_{sr} - O_s - U(1 - F_{sr}), \qquad \forall s, r, \qquad (12)$$

The full MIP model, OPLRAreg, is given by:

$$\text{minimise} \quad z$$

$$\text{subject to}$$

$$\text{Equations} \qquad (1) - (12)$$

2.2.1. Regularisation and implicit feature selection

Besides reducing the risk of overfitting the data, regularisation has an important role in the selection of features for the model. Without regularisation ($\lambda = 0.00$), regression coefficients can assume any numerical value, so in cases where regression coefficients are large, even minor deviations from the data seen during training can lead to large prediction error. This effect creates models that are too specialised to the training data and can predict samples in the external validation set poorly. On the other hand, when regularisation is enforced ($\lambda > 0.00$), regression coefficients are forced to assume smaller values and deviations from training will not have a large impact on the accuracy of predictions.

As an additional effect of regularisation, the coefficients of many features are set to zero, indicating that these descriptors are not important to prediction. This implicit feature selection step also reduces the number of loops to identify the partition feature of OPLRAareg and reduces the size of MIP models in the remaining iterations. The effect of regularisation in the accuracy of OPLRAreg is also discussed on Section 3.1 and illustrated on results shown in Table 2.

2.3. Proposed algorithm

Algorithm 1 summarises the iterative process of the proposed OPLRAreg method with the modifications described above. First, a simple linear regression is fit to the training data (number of regions $R = 1$) and z is recorded. The regularisation will ensure that the coefficient of less relevant features are set to zero and only descriptors that have been effectively used in the linear equation are kept in the next iterations. Note that constraints related to breakpoints and assignment of samples to regions (Equations 7, 8, 9) are not

Algorithm 1 OPLRA with proposed modifications

```
Solve OPLRAreg for R = 1                                   ▷ Simple linear regression
$\text{ERROR}_{\text{current}} \leftarrow z$
$\text{ERROR}_{\text{old}} \leftarrow \infty$
$\text{ERROR}_{\text{tmp}} \leftarrow \infty$
$f_{\text{best}} \leftarrow \{\}$
$F \leftarrow \{f \in \mathbf{f} \mid W_{r_1,f} \neq 0\}$   ▷ Implicit feature selection
$R \leftarrow 2$
for $i \leftarrow 1; i \leftarrow i+1; i \leq F$ do          ▷ Selects best partition feature in 2 regions
    Solve OPLRAreg with 2 regions and partition feature $f_i$
    if $z < \text{ERROR}_{\text{tmp}}$ then
        $\text{ERROR}_{\text{tmp}} \leftarrow z$
        $f_{\text{best}} \leftarrow f_i$
    end if
end for
$\text{ERROR}_{\text{old}} \leftarrow \text{ERROR}_{\text{current}}$
$\text{ERROR}_{\text{current}} \leftarrow \text{ERROR}_{\text{tmp}}$
$f^* \leftarrow f_{\text{best}}$
while $\text{ERROR}_{\text{current}} < (1-\beta)\text{ERROR}_{\text{old}}$ do   ▷ Number of regions increases
    $R \leftarrow R+1$
    Solve OPLRAreg with $R$ regions and partition feature $f^*$
    $\text{ERROR}_{\text{old}} \leftarrow \text{ERROR}_{\text{current}}$
    $\text{ERROR}_{\text{current}} \leftarrow z$
end while
return partition feature $f^*$, breakpoints $X_{rf}$, regression coefficients for each region $W_{rf}$
```

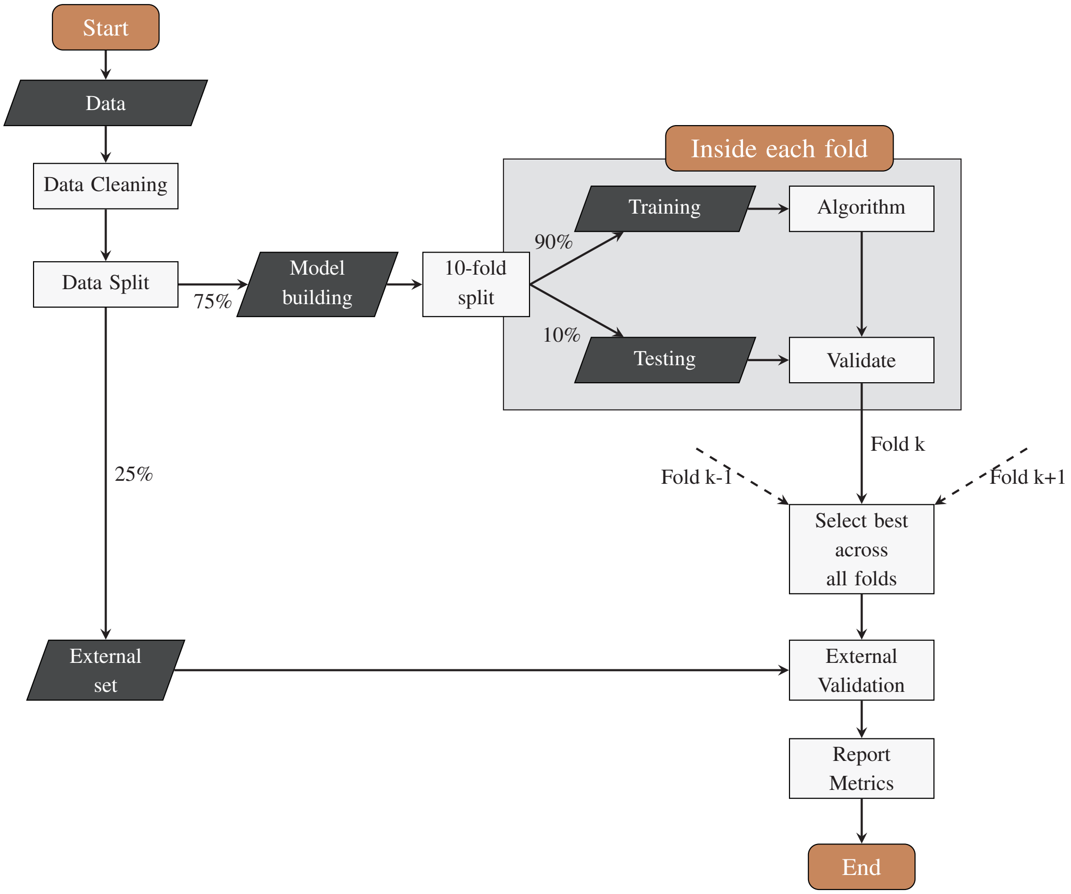


Figure 1: Validation scheme adopted in this study.

Table 2: Comparison of OPLRA performance for different regularisation parameters

	rDHFR	hDHFR	CHRM3	NPYR1	NPYR2
MAE					
$\lambda = 0.000$	54.76 ± 70.31	28.56 ± 29.51	103.15 ± 118.52	124.28 ± 105.66	153.32 ± 147.09
$\lambda = 0.005$	0.74 ± 0.07	0.79 ± 0.06	0.78 ± 0.06	0.70 ± 0.09	0.57 ± 0.12
$\lambda = 0.010$	0.84 ± 0.06	0.84 ± 0.08	0.78 ± 0.07	0.76 ± 0.09	0.63 ± 0.09
$\lambda = 0.020$	0.90 ± 0.07	0.85 ± 0.05	0.83 ± 0.10	0.73 ± 0.11	0.61 ± 0.08
Time (min)					
$\lambda = 0.000$	90.84 ± 3.98	44.93 ± 3.23	60.44 ± 1.57	24.82 ± 3.84	24.59 ± 4.30
$\lambda = 0.005$	9.73 ± 0.83	4.64 ± 0.73	7.40 ± 0.60	4.70 ± 0.78	5.38 ± 0.62
$\lambda = 0.010$	5.41 ± 0.93	1.86 ± 0.36	5.15 ± 0.71	3.69 ± 1.59	2.82 ± 0.39
$\lambda = 0.020$	2.17 ± 0.78	0.62 ± 0.14	2.48 ± 0.21	2.31 ± 0.23	1.55 ± 1.59
Features					
$\lambda = 0.000$	80.0 ± 0.00	75.9 ± 0.32	86.2 ± 0.42	69.2 ± 0.42	67.0 ± 0.00
$\lambda = 0.005$	21.9 ± 1.60	19.9 ± 1.80	23.7 ± 1.57	22.4 ± 2.80	25.1 ± 3.41
$\lambda = 0.010$	13.4 ± 1.43	8.9 ± 2.69	16.8 ± 1.81	16.4 ± 2.12	14.7 ± 1.83
$\lambda = 0.020$	5.0 ± 0.67	2.6 ± 0.52	12.0 ± 2.26	9.4 ± 0.97	7.3 ± 0.48
Regions					
$\lambda = 0.000$	4.3 ± 0.82	4.4 ± 0.97	4.0 ± 0.47	4.8 ± 1.03	4.8 ± 1.87
$\lambda = 0.005$	2.0 ± 0.00	2.0 ± 0.00	2.0 ± 0.00	2.3 ± 0.48	2.0 ± 0.00
$\lambda = 0.010$	2.1 ± 0.32	2.0 ± 0.00	2.0 ± 0.00	2.3 ± 0.95	2.0 ± 0.00
$\lambda = 0.020$	2.1 ± 0.32	2.0 ± 0.00	2.0 ± 0.00	2.0 ± 0.00	2.3 ± 0.95

used while solving the first MIP model and all samples are assigned to a single region, $F_{sr_1} = 1$ according to Equation 6. Then, an MIP with two regions ($R = 2$) is solved for each selected feature and the feature that corresponds to the best model in this iteration is determined as the partition feature f^* for the remaining iterations.

The number of regions increases until the improvement of absolute error in consecutive iterations is no more than a user-defined parameter β. In this study, the value of β did not affect results significantly (see sensitivity analysis in Section 3.1 and supplementary data), therefore a small, non-zero value is suggested.

2.4. Implementation and Validation scheme

The validation scheme used in this study is illustrated in Figure 1 and is aligned with state-of-the art QSAR model validation procedures [1, 28]. Data sets are initially split at random, 75% of samples were used for model building and 25% for the external validation set. Samples in the model building set were further split in internal training and testing sets using stratified sampling techniques available in *caret* for 10 repeated 10-fold cross-validation. Samples in external set are only used to assess the final models. All algorithms used in this study were tested across samples in each fold and, after cross-validation, the best model for each algorithm was selected and used to predict the outcome of samples in the external validation set. This data split and model selection procedure was repeated 5 times and the average accuracy is reported.

3. Results and Discussion

In this section, results of the piecewise linear regression are presented and compared to other machine learning algorithms. We also illustrate the flexibiliy of the mathematical programming methodology and show how the division of regions can help elucidate the properties of QSAR data sets.

3.1. Parameter optimisation

Initial tests were run with a single round of 10-fold cross-validation to understand the impact of the regularisation parameter in the new mathematical programming model. For these tests, we varied the regularisation parameter using the following values: $\lambda \in [0.000, 0.005, 0.010, 0.200]$ with $\beta = 0.03$. Table 2 shows the effect of regularisation in terms of mean and standard deviations of MAE, CPU time required to run each test case and average number of features and regions detected by the algorithm.

These results clearly show an improvement in the performance of OPLRA with the introduction of the regularisation term in OPLRAreg. Prediction variable pIC_{50} in most data sets ranged from 4 to 11, but the mean absolute error of tests with no regularisation ($\lambda = 0$) was far beyond this range. The best regularisation parameter value was found to be $\lambda = 0.005$, where prediction accuracy was consistently better on all data sets when compared to tests with nonzero λ. OPLRAreg was also 4 to 10 times faster with the optimal regularisation parameter and the average number of features selected was around 20.

Sensitivity analysis with regards to β was also undertaken, where the regularisation parameter was fixed at $\lambda = 0.005$ and all five data sets were run with the following values for β: [0.01, 0.03, 0.05, 0.10, 0.15, 0.20]. Results showed that MAE did not change substantially in any test case (supplementary data). Therefore, parameters for OPLRAreg were set at $\lambda = 0.005$ and $\beta = 0.03$.

3.2. Algorithm results

On average, OPLRAreg detects 3 regions and selects 20 to 25 features for the QSAR data sets used in this study, as shown in Table 3. Illustrative examples of QSAR models generated by the algorithm for data sets hDHFR and NPYR1 can be seen in Figures 2 and 3, respectively. The distribution of scaled descriptor values for the partition feature is shown against biological activity (pIC_{50}), as well as breakpoints and equations detected for each region.

In the first example, shown in Figure 2, the partition feature is *MDEN-11*, a descriptor related to the distance between all primary nitrogen atoms in the molecular graph. Most samples in this data set have either MDEN-11 = 0 (23.4%) or MDEN-11 = 0.43 (71.96%) and OPLRAreg captures different equations for those cases. The algorithm assigns molecules without nitrogen atoms or with small distance between these atoms to region 1, another multiple linear relationship encompassing samples in $0.17 \leq$ MDEN-11 < 0.72 and it estimates that $pIC_{50} = 5.04$ for the few cases where *MDEN-11* is large. Most selected features are related to topological characteristics of the molecules and are either related to connectivity of atoms (topoShape, MDEN-22, MDEC-23, C1SP3, C3SP3, SC-5, SCH-5) or to the number of specific groups found in the molecules, as is the case of nE (number of glutamic acid) and fragments identified as Kier-Hall SMART descriptors (*khs-*) [19].

Similarly, we can interpret the breakpoints and equations for NPYR1 shown in Figure 3. *C1SP3* is the partition feature and it represents the number of singly bound carbon atoms bound to one other carbon. Descriptors are scaled during preprocessing of the data and the interval $[0,1]$ represents the original range $[0,41]$. Therefore, molecules with at most 4 such types of carbon ($C1SP3 \leq 0.11$) are predicted by equation in region 1 while those ranging from 4 to 11 atoms belong to region 2. Region 3 captures rare cases (only 8% of the samples) where molecules have more than 11 carbons with the defined connectivity.

3.3. Overall Variable Importance

In order to express the importance of molecular descriptors in the QSAR models by OPLRAreg, a simple metric would be to rank each feature according to the number of times it appears in equations across all regions. In order to also account for the number of samples that are represented in each region, another option for the variable importance measure may be the percentage of samples predicted by equations containing a specific feature. We have computed this percentage for each feature in the best OPLRAreg models selected after cross-validation and averaged across the five external validation sets to generate an overall importance score for these tests. Table 4 shows the top 15 features ranked according to this score per data set. The types of descriptors more frequently selected are briefly described below.

Fragment count: Descriptors that represent the number of specific fragments or substructures. Of these, Kier-Hall descriptors [19, 33], identified by the prefix *khs*, were selected more often and had a high score of importance in OPLRAreg models.

–*khs-** descriptors
–nRings6
–nBase
–nAtomLAC
–Aminoacids count (nG, nF)

MDE descriptors: Molecular Distance Edge descriptors represent the distance edge between specific atom types in the molecular graph. MDEO.11 and MDEO.22, for example, calculate the distance between all primary oxygen and all secondary oxygen, respectively.

–MDEN.11
–MDEN.13
–MDEN.22
–MDEN.33
–MDEC.12
–MDEC.13
–MDEC.22
–MDEC.33
–MDEO.11

Carbon connectivity: Descriptors describing carbon types.

–C1SP3
–C3SP2
–C3SP3

Log P descriptors: Descriptors related to the lipophilicity of molecules, an important property determinant of the absorption, transport and excretion of a drug. The logarithm of the partition coefficient, log P, measures the affinity of a molecule for a lipid over an aqueous medium and can be approximated by various numerical methods:

–ALogP
–ALogP2
–XLogP
–MLogP

BCUT descriptors: Descriptors based on eigenvalues of a matrix representation of the molecular graph where diagonal weights contain either atomic **w**eight, partial **c**harge or **p**olarizability properties of molecules.

–BCUTc.1l
–BCUTc.1h
–BCUTw.1l
–BCUTp.1h

BCUT descriptors condense a great deal of information and are more difficult to interpret. It is harder to relate

Table 3: Average number of regions and selected features found by OPLRAreg during cross-validation

	rDHFR	hDHFR	P20309	NPYR1	NPYR2
Regions	3.10 (±0.31)	3.00 (±0.00)	3.00 (±0.06)	3.46 (±0.58)	3.04 (±0.19)
Features	22.30 (±2.24)	18.93 (±2.13)	25.53 (±2.50)	22.66 (±2.58)	24.95 (±2.89)

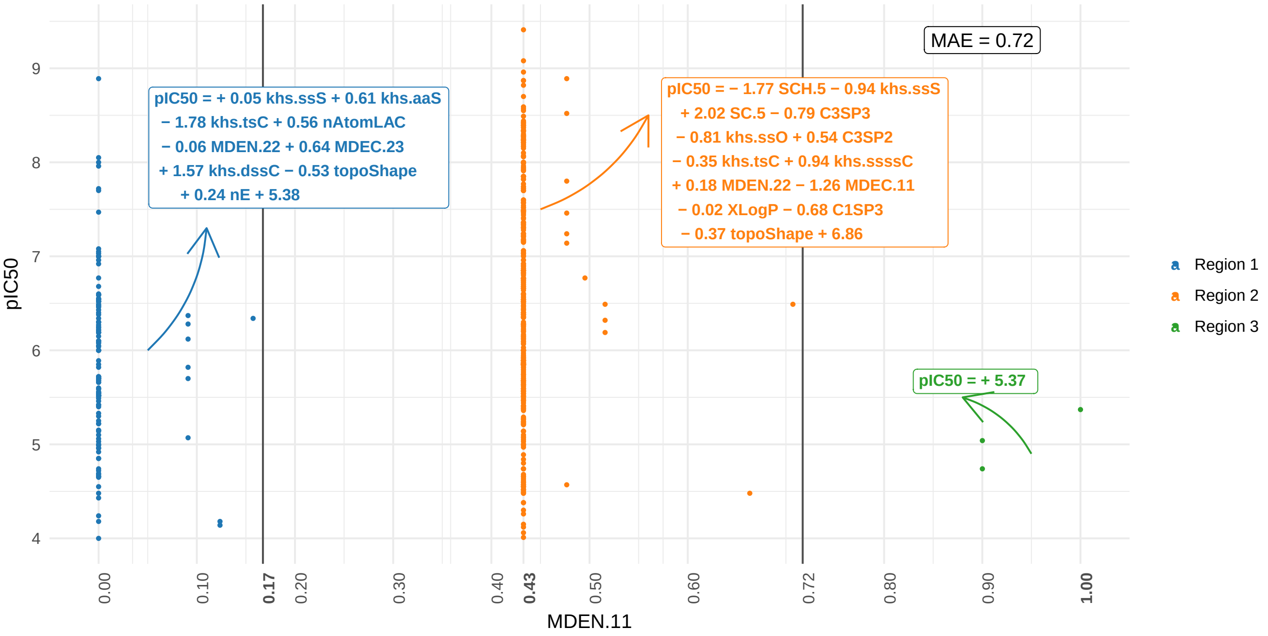


Figure 2: Breakpoints, regions and equations found by OPLRAreg for data set hDHFR

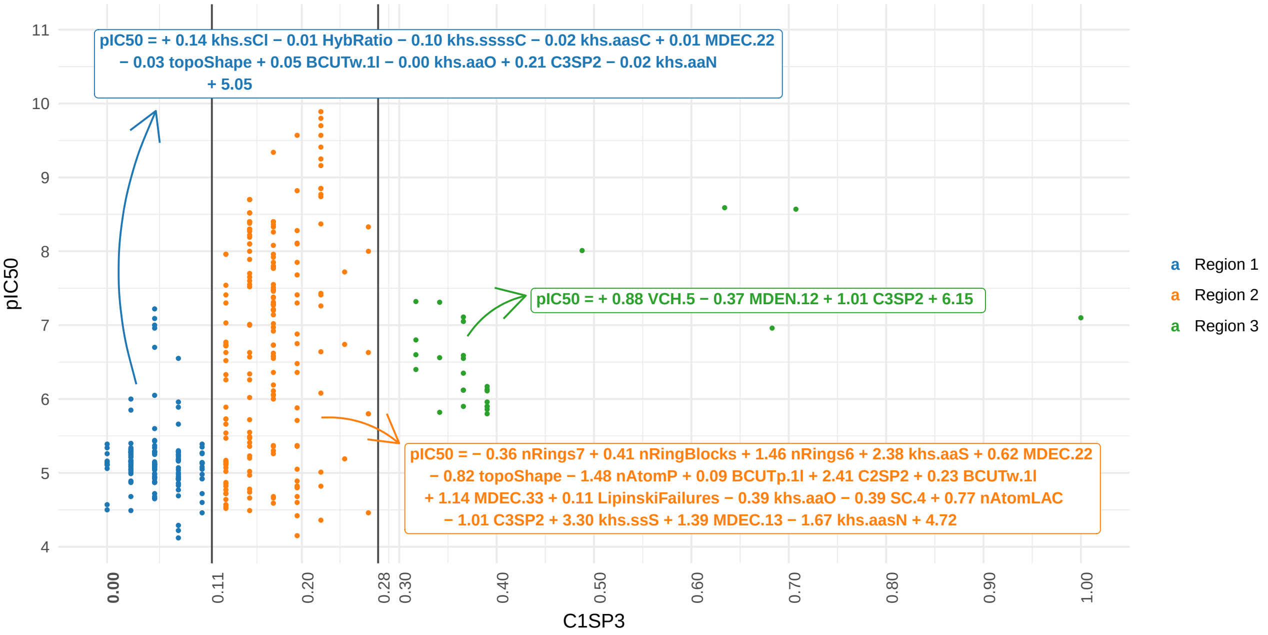


Figure 3: Breakpoints, regions and equations found by OPLRAreg for data set NPYR1

Table 4: Top 15 features and their importance score for each data set

Rank	rDHFR		hDHFR		P20309		NPYR1		NPYR2	
	Descriptor	Score	Descriptor	Score	Descriptor	Score	Descriptor	Score	Descriptor	Score
1	VC.5	98.86	khs.aaNH	99.45	MDEC.33	98.21	SC.6	99.15	SC.4	99.73
2	ALogP	95.94	VP.7	99.45	BCUTc.1h	98.19	BCUTw.1l	91.71	MDEO.11	99.47
3	MDEN.13	93.90	khs.ssS	94.60	BCUTw.1l	98.13	C3SP2	77.01	khs.ddssS	96.47
4	MDEC.22	91.77	topoShape	93.95	nG	98.13	khs.aaO	74.01	C3SP3	92.35
5	SCH.6	85.03	ALogp2	87.55	VCH.6	98.13	khs.aaS	71.98	LipinskiFailures	91.71
6	MDEC.33	84.80	khs.aaN	87.55	ATSm1	97.45	C3SP3	70.90	BCUTp.1h	91.51
7	MDEC.13	82.26	MDEN.22	84.19	khs.aaaC	97.45	MDEC.12	70.20	C3SP2	91.44
8	khs.ssNH	81.01	XLogP	78.93	nF	96.77	nAtomLAC	64.27	khs.aaO	91.44
9	ALogp2	80.11	MDEC.22	78.78	nRings6	94.39	LipinskiFailures	64.24	HybRatio	91.31
10	C1SP3	76.80	MDEN.11	78.78	MDEN.33	92.86	nRings6	62.29	khs.sF	91.31
11	nRings6	75.82	nBase	78.78	LipinskiFailures	91.84	SC.4	62.15	khs.ssO	86.36
12	tpsaEfficiency	73.66	LipinskiFailures	77.21	khs.dsCH	91.04	khs.aaaC	61.58	tpsaEfficiency	86.36
13	BCUTc.1l	72.69	MDEO.22	76.66	khs.dssC	90.22	MDEO.11	61.30	BCUTc.1l	83.69
14	khs.aaaC	72.39	C3SP2	75.89	khs.ssNH	89.56	khs.aasN	60.85	MDEN.22	83.69
15	khs.dssC	72.39	C1SP3	75.83	ALogp2	89.20	khs.sCl	60.34	ATSc3	82.80

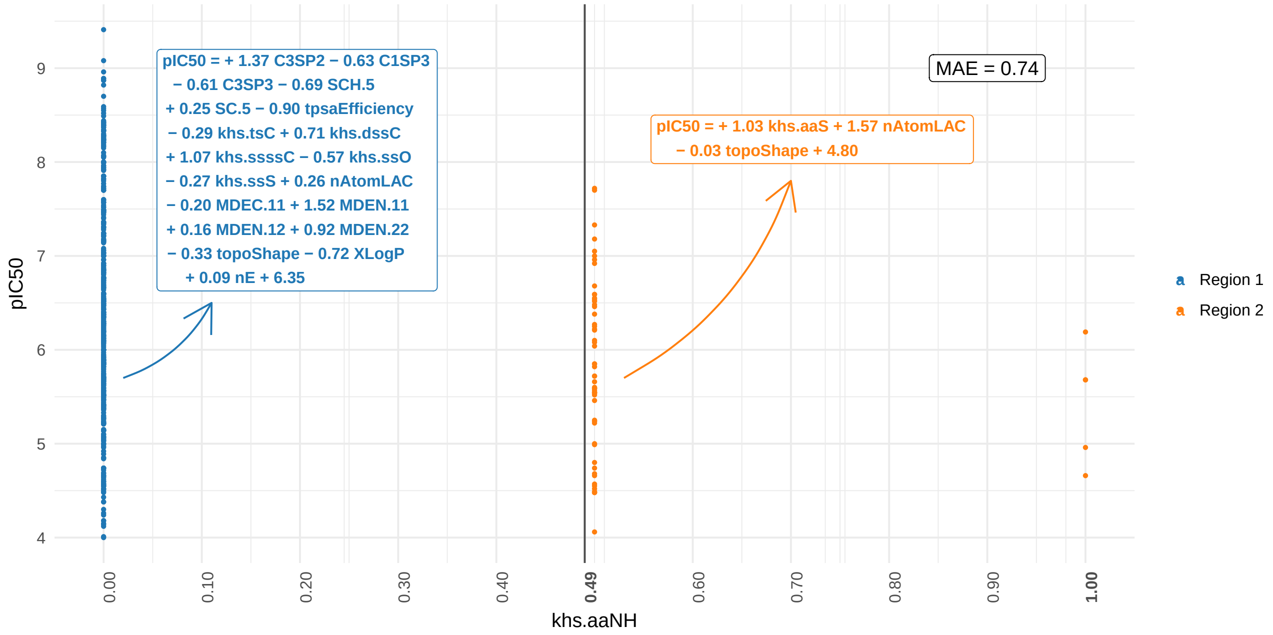


Figure 4: Piecewise model for hDHFR inhibitors with khs.aaNH as the partition feature

the values of these descriptors to properties in the molecular graph in the same way as descriptors describing fragments, atom types or distances. However, BCUT descriptors have been proven useful in QSAR models as representative features of ligand-receptor interactions [34]. A possible workaround to interpret QSAR models where these features have been deemed important is to complement the analysis of BCUT values with other correlated descriptors and visual data exploration [35].

3.4. Custom constraints to the model

In the previous sections, we showed how OPLRAreg automatically finds a feature to split the data into regions. Now, suppose that we want to discover the possible structure-activity relationships of inhibitors for a particular attribute of interest and we have reasons to believe that the data can be split into a known number of regions. The proposed method is flexible enough to accommodate this requirement, i.e. it allows the user to specify which descriptor to use to partition the data and, if necessary, the exact number of regions.

To provide an illustrative example, OPLRAreg identifies the alternative optimal piecewise model shown in Figure 4 for hDHFR inhibitors when we use $f^* =$ khs.aaNH as the partition feature. The fragment captured by khs.aaNH is an aromatic nitrogen connected to a single hydrogen atom and the value calculated by the descriptor is simply the number of occurrences of this fragment in a molecule. In this dataset, there are only three distinct values of khs.aaNH: $[0,1,2]$, scaled to $[0.0,0.5,1.0]$, as shown in the graph. Examples of compounds with the distinct khs.aaNH values are shown in Figure 5.

The model identified by OPLRAreg splits the data in two regions by the breakpoint khs.aaNH $= 0.49$, which in practice separates the compounds containing the fragment (khs.aaNH > 0) from those without the fragment (khs.aaNH $= 0$). Examples of compounds with distinct khs.aaNH values are shown in Figure 5. One possible explanation is that the hydrogen atoms in the fragment could form H-bonds with hDHFR, affecting the binding to the protein thus leading to the two distinct rules of potency, as identified by the algorithm. This hypothesis would have to be proven by computational or experimental means, i.e. docking or appropriate assay at the right pH and bioactive conformation. The accuracy of the new model (MAE $= 0.74$) is very similar to the one identified by the standard workflow in Figure 2 (MAE $= 0.72$) and the selection of one over the other would depend on the practical applications of this QSAR model.

It is worth noting that in cases where known structure-activity relationships exist, custom constraints discussed above represent valid and even encouraged modifications to the standard procedure presented in Algorithm 1. In such cases of additional, user-specific constraints being specified, the algorithm requires fewer iterations and OPLRAreg will run faster than the original workflow since the loop for 2 regions will not be executed. It can run faster still, if the number of regions is small and also specified beforehand, as only one MIP model will need to be solved.

3.5. Comparison with other algorithms

Results obtained with OPLRAreg were compared to other machine learning algorithms available through the R package *caret*, following the validation scheme shown in Figure 1. Five nonlinear algorithms (Support Vector Machine Radial [36, 37], Random Forest [38], Neural Networks [39], Generalised Linear Model, Random Generalised Linear Model [40]) and four linear algorithms (Lasso, Linear Regression, Partial Least Square and Elastic Net) in *caret* were used. OPLRAreg parameters were set to $\lambda = 0.005$ and $\beta = 0.03$. Default parameters were used for Random GLM ($nBags = 100$ and default settings for *nFeaturesInBag*). Parameters for other algorithms in *caret* package were defined by grid search, as used in [28].

All algorithms were trained on the same training/testing data splits for 10-fold cross-validation, repeated 10 times. For each algorithm, the best model corresponding to the smallest MAE in the internal test set was used to predict the activity of samples in the external validation set.This process was repeated five times and the performance of each algorithm in the external validation set is shown in Figure 6 for NPYR1, NYPR2, CHRM3 and hDHFR datasets and Figure 7 for dataset rDHFR. The box plots represent the distribution of prediction error and dots outside the boxes represent outlier predictions for each algorithm.

In tests illustrated in Figure 6, performance of OPLRAreg was similar to state-of-the-art algorithms such as Random Forest, SVM Radial and Neural Networks. The average error of these algorithms was below 1 log unit and close to $MAE = \pm 0.60$, which is the expected error for biological activity reported in ChEMBL [12].

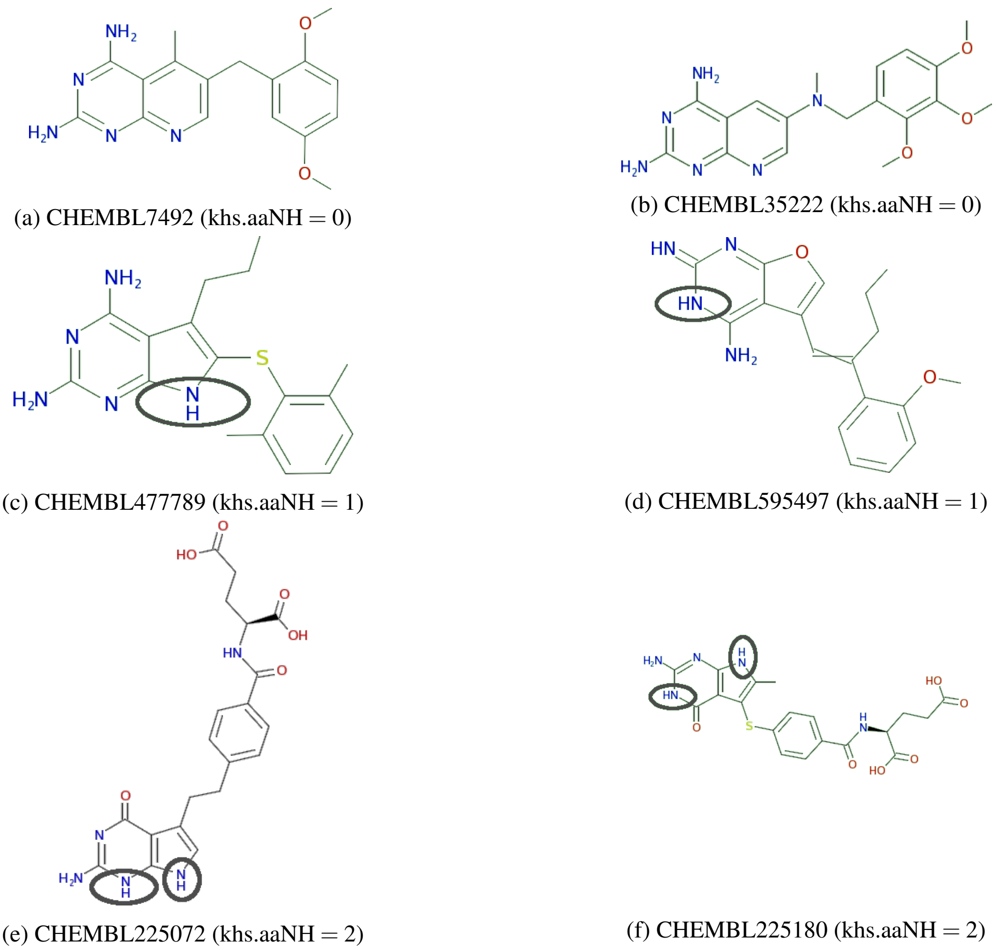
Compared to OPLRAreg, these algorithms produce less interpretable or mathematically explicit models. In Random Forest models, for example, a randomly selected subset of features is used to navigate feature space and reach a numerical outcome in the form of a decision tree. Such a regression tree in itself is somewhat easy to interpret, in terms of linking molecular descriptors to an activity prediction. However, as the final prediction of Random Forest involves averaging across hundreds of decision trees, the resulting model becomes a convoluted means of modelling structure-activity relationships, so clarity in how molecular descriptors contribute to drug activity becomes hard to interpret. Similar effects are also noted with respect to models produced by SVM and Neural Networks. OPLRAreg, however, offers an optimal means of separating the data set into appropriate regions, with each region specifying a clear, mathematical relation of molecular descriptors to predicted activity. Furthermore, as illustrated above, OPLRAreg can also be customised through user-specified mathematical constraints.

Apart from average prediction performance, results for each algorithm show the number of outlier predictions (Figure 6). The existence of such outliers can be attributed to the heterogeneity of data sets, as well as the presence of activity cliffs, both inherent limitations of QSAR models [41–43]. However, we note that models built with Linear Regression, Lasso and in some test cases Random GLM, appeared to have produced more outliers. The error in some individual predictions have reached more than ten orders of magnitude in CHRM3 and hDHFR data sets, indicating poorer performance of tests through these algorithms.

The case of rDHFR proved a more variable case, in terms of comparative tests. The performance of OPLRAreg was once again similar to Random Forest, SVM Radial and Neural Networks, as shown in Figure 7. On the other hand, the error distribution of Linear Regression, Lasso and Random GLM was much worse than in the previous data sets and have exceeded hundreds and even thousands of orders of magnitude. These algorithms are shown in their own separate box and scale to allow comparison, and such variability in results suggests that they are more prone to overfitting.

4. Conclusions

In this study, we report the development and application of a piecewise linear regression algorithm based on mixed integer programming models for predictive QSAR tasks. We have illustrated how such a combinatorial optimisation framework under a robust validation scheme can be used to predict biological activity of chemical compounds against a common target and showed that this approach offers interpretable, customisable models with acceptable accuracy



(a) CHEMBL7492 (khs.aaNH = 0)

(b) CHEMBL35222 (khs.aaNH = 0)

(c) CHEMBL477789 (khs.aaNH = 1)

(d) CHEMBL595497 (khs.aaNH = 1)

(e) CHEMBL225072 (khs.aaNH = 2)

(f) CHEMBL225180 (khs.aaNH = 2)

Figure 5: Example of compounds with different khs.aaNH values. The fragments counted by the molecular descriptor are highlighted in the relevant figures.

of prediction. The datasets and source code are available at https://github.com/KISysBio/qsar-models.

Interpretability is one of the major drawbacks of black-box machine learning and deep learning algorithms and the method presented here contributes towards more interpretable QSAR models. OPLRAreg not olny determines optimal splits of the data into different subgroups (regions) using one of the molecular descriptors in the data set, but also identifies a suitable equation to predict the biological activity of samples that fall in each of these regions. The descriptor used to split the data as well as the linear equations in each region are output by the algorithm in a transparent manner. A modeller can then use such information and compare the features that are more relevant in activity prediction across different subgroups of compounds.

The proposed method has a comparable prediction accuracy compared to other non-linear and less interpretable algorithms. In addition, it offers unique benefits that stem from the properties of mathematical programming. In addition to its interpretable capabilities, OPLRAreg allows for customisation of the model. Where possible, the method allows that the modeller specifies the exact number of regions and the molecular descriptor to be used in order to partition the data. This flexibility also allows the user to compare different grouping of molecules according to the specific needs of a QSAR project.

We intend to improve the algorithm by introducing automatic variable transformations in the model and alternative definition of the regions in the future. The algorithm can potentially cope better with the non-linearities of this type of data, while retaining the transparent and interpretable capabilities of the mathematical model. In the future, we would also like to develop an optimisation model akin to the inverse QSAR problem to design new potent compounds on the basis of selected molecular descriptors identified by OPLRAreg in each sub-group of molecules.

Acknowledgements. Jonathan Cardoso-Silva acknowledges funding from CAPES, Brazil (Process number 13312138). Financial support by The Leverhulme Trust (RPG- 2012-686 and RPG-2015-240) and the UK Engineering & Physical Sciences Research Council (EP/I033270/1 and EP/M027856/1) are also gratefully acknowledged.

Key words: qsar, regression, piecewise regression, mathematical programming, integer programming

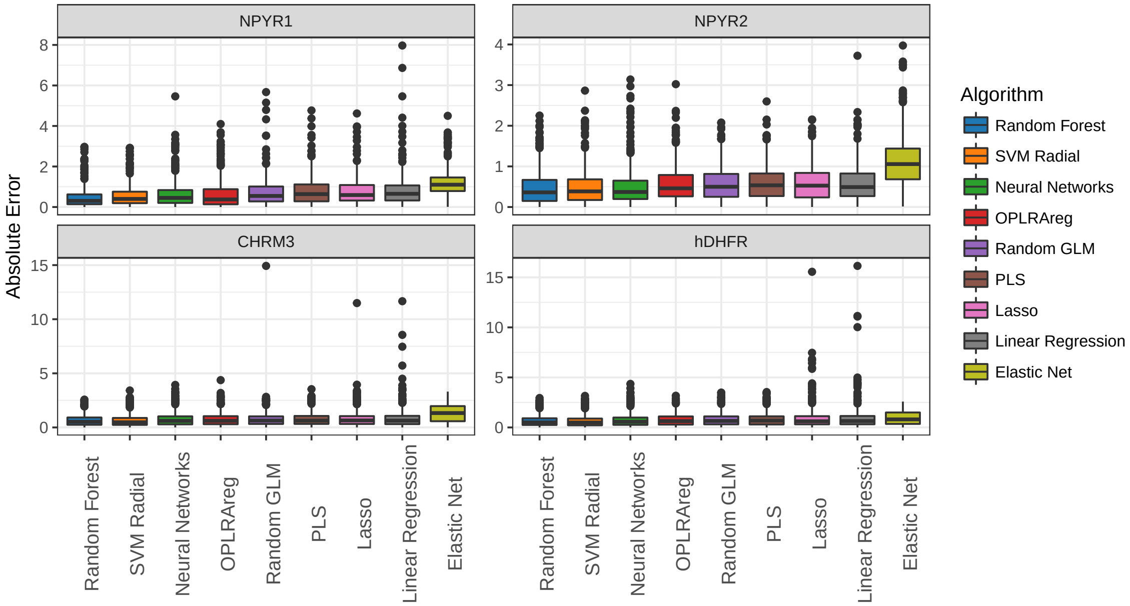


Figure 6: Performance of OPLRAreg compared to other machine learning algorithms. The box plots show the distribution of error for samples in the external set, combining all five batches of test.

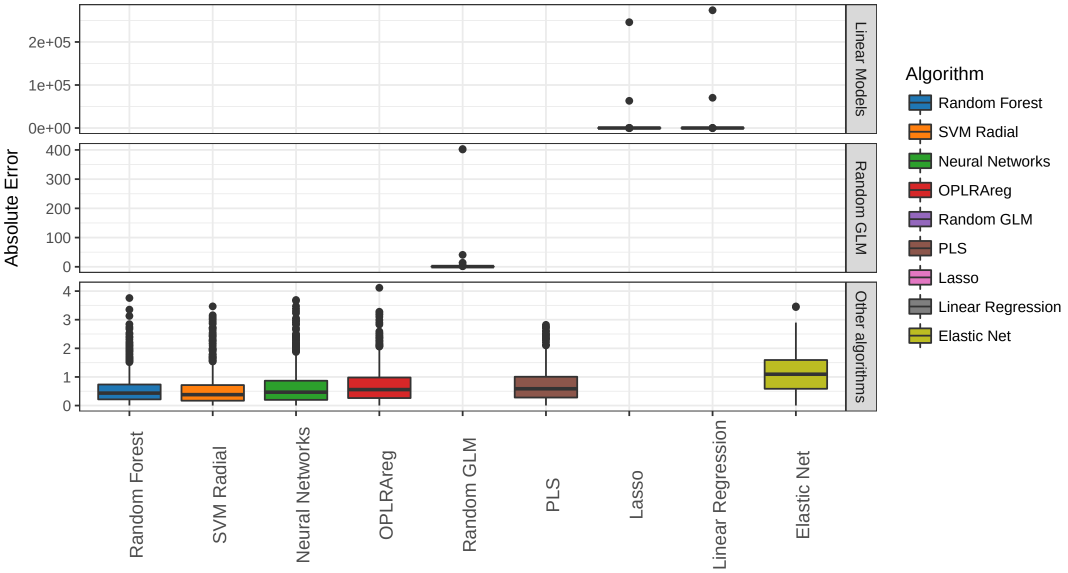


Figure 7: Comparison of OPLRAreg to other machine learning algorithms for dataset rDHFR. Linear regression, Random GLM and the Lasso algorithms produced results that were erroneous by many margins of magnitude and were represented in their own separate scale.

References

[1] A. Tropsha, Molecular Informatics **29**(6-7), 476–488 (2010).

[2] C. C. Melo-Filho, R. F. Dantas, R. C. Braga, B. J. Neves, M. R. Senger, W. C. G. Valente, J. M. Rezende-Neto, W. T. Chaves, E. N. Muratov, R. A. Paveley, N. Furnham, L. Kamentsky, A. E. Carpenter, F. P. Silva-Junior, and C. H. Andrade, Journal of Chemical Information and Modeling **56**(7), 1357–1372 (2016).

[3] A. Rescifina, G. Floresta, A. Marrazzo, C. Parenti, O. Prezzavento, G. Nastasi, M. Dichiara, and E. Amata, European Journal of Pharmaceutical Sciences **106**, 94–101 (2017).

[4] M. N. Gomes, R. C. Braga, E. M. Grzelak, B. J. Neves, E. N. Muratov, R. Ma, L. K. Klein, S. Cho, G. R. Oliveira, S. G.

Franzblau, and C. H. Andrade, European Journal of Medicinal Chemistry **137**, 126–138 (2017).

[5] J. B. O. Mitchell, Wiley Interdisciplinary Reviews: Computational Molecular Science **4** (2014).

[6] J. T. Leonard and K. Roy, QSAR & Combinatorial Science **26**(9), 980–990 (2007).

[7] Y. Zhou, Z. Ni, K. Chen, H. Liu, L. Chen, C. Lian, and L. Yan, Protein Journal **32**(7), 568–578 (2013).

[8] M. Salahinejad, T. C. Le, and D. A. Winkler, Molecular Pharmaceutics **10**(7), 2757–2766 (2013).

[9] F. Klepsch, P. Vasanthanathan, and G. F. Ecker, Journal of Chemical Information and Modeling **54**(1), 218–229 (2014).

[10] S. Karabulut, N. Sizochenko, A. Orhan, and J. Leszczynski, Journal of Molecular Graphics and Modelling **70**, 23–29 (2016).

[11] I. V. Tetko, D. M Lowe, and A. J. Williams, Journal of Cheminformatics **8**(1), 2 (2016).

[12] I. Cortes-Ciriano and A. Bender, Journal of Chemical Information and Modeling **55**(12), 2682–2692 (2015).

[13] T. Fujita and D. A. Winkler, Journal of Chemical Information and Modeling p. 269274 (2016).

[14] V. E. Kuzmin, P. G. Polishchuk, A. G. Artemenko, and S. A. Andronati, Molecular Informatics **30**(6-7), 593–603 (2011).

[15] P. G. Polishchuk, V. E. Kuźmin, A. G. Artemenko, and E. N. Muratov, Molecular Informatics **32**(9-10), 843–853 (2013).

[16] V. M. Alves, E. N. Muratov, S. J. Capuzzi, R. Politi, Y. Low, R. C. Braga, A. V. Zakharov, A. Sedykh, E. Mokshyna, S. Farag, C. H. Andrade, V. E. Kuzmin, D. Fourches, and A. Tropsha, Green Chemistry **18**(16), 4348–4360 (2016).

[17] P. Polishchuk, Journal of Chemical Information and Modeling **57**(11), 2618–2639 (2017).

[18] A. Cherkasov, E. N. Muratov, D. Fourches, A. Varnek, I. I. Baskin, M. Cronin, J. Dearden, P. Gramatica, Y. C. Martin, R. Todeschini, V. Consonni, V. E. Kuz'Min, R. Cramer, R. Benigni, C. Yang, J. Rathman, L. Terfloth, J. Gasteiger, A. Richard, and A. Tropsha, Journal of Medicinal Chemistry **57**(12), 4977–5010 (2014).

[19] D. Butina, Molecules **9**(12), 1004–1009 (2004).

[20] R. Kiralj and M. M. C. Ferreira, Journal of the Brazilian Chemical Society **20**(4), 770–787 (2009).

[21] S. Pirhadi, F. Shiri, and J. B. Ghasemi, RSC Adv. **5**(127), 104635–104665 (2015).

[22] L. Xu, H. Y. Fu, Q. B. Yin, Y. Fan, M. Goodarzi, and Y. B. She, Chemometrics and Intelligent Laboratory Systems **159**, 187–195 (2016).

[23] F. R. Burden and D. A. Winkler, QSAR and Combinatorial Science **28**(6-7), 645–653 (2009).

[24] G. Xu and L. G. Papageorgiou, Computers & Industrial Engineering **56**(4), 1205–1215 (2009).

[25] L. Yang, C. Ainali, A. Kittas, F. O. Nestle, L. G. Papageorgiou, and S. Tsoka, Mathematical Biosciences **260**, 25–34 (2015).

[26] J. C. Silva, L. Bennett, L. G. Papageorgiou, and S. Tsoka, The European Physical Journal B **89**(2), 39 (2016).

[27] G. Papadatos, A. Gaulton, A. Hersey, and J. P. Overington, Journal of Computer-Aided Molecular Design **29**(9), 885–896 (2015).

[28] G. Tsiliki, C. R. Munteanu, J. A. Seoane, C. Fernandez-Lozano, H. Sarimveis, and E. L. Willighagen, Journal of Cheminformatics **7**(1), 1–16 (2015).

[29] M. Kuhn, caret: Classification and Regression Training, 2016, R package version 6.0-73.

[30] C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann, and E. Willighagen, Journal of Chemical Information and Computer Sciences **43**(2), 493–500 (2003).

[31] L. Yang, S. Liu, S. Tsoka, and L. G. Papageorgiou, Expert Systems with Applications **44**, 156–167 (2016).

[32] K. Roy, R. N. Das, P. Ambure, and R. B. Aher, Chemometrics and Intelligent Laboratory Systems **152**, 18–33 (2016).

[33] L. H. Hall and L. B. Kier, Journal of Chemical Information and Modeling **35**(6), 1039–1045 (1995).

[34] D. J. Livingstone, Journal of Chemical Information and Computer Sciences **40**(2), 195–209 (2000).

[35] B. Pirard and S. D. Pickett, Journal of Chemical Information and Computer Sciences **40**(6), 1431–1440 (2000).

[36] A. Karatzoglou, A. Smola, K. Hornik, and A. Zeileis, Journal of Statistical Software **11**(9), 1–20 (2004).

[37] C. C. Chang and C. J. Lin, ACM Transactions on Intelligent Systems and Technology (TIST) **2**, 1–39 (2013).

[38] L. Breiman, Machine Learning **45**, 5–32 (2001).

[39] B. D. Ripley and N. L. Hjort, Pattern Recognition and Neural Networks, 1st edition (Cambridge University Press, New York, NY, USA, 1995).

[40] L. Song, P. Langfelder, and S. Horvath, BMC Bioinformatics **14**(1), 5 (2013).

[41] G. M. Maggiora, Journal of Chemical Information and Modeling **46**(4), 1535 (2006).

[42] D. Stumpfe and J. Bajorath, Journal of Medicinal Chemistry **55**(7), 2932–2942 (2012).

[43] M. Cruz-Monteagudo, J. L. Medina-Franco, Y. Perez-Castillo, O. Nicolotti, M. N. D. Cordeiro, and F. Borges, Drug Discovery Today **19**(8), 1069–1080 (2014).