THE UNIVERSITY *of* EDINBURGH
School of Biological Sciences

CrystalBoost: A Machine Learning Scoring Function to Accelerate Medicine Discovery

Exam Number: B221734

In partial fulfilment of the requirement for the Degree of Master of Science in Synthetic Biology & Biotechnology at The University of Edinburgh

2023

Supervisor: Dr. Douglas Houston

Word Count: 9535

Abstract

Virtual screening enables the identification of promising drug candidates by predicting binding affinity between ligands and protein targets. Scoring functions, which predict these interactions, remain imperfect, necessitating extensive wet lab testing. This project explored improving scoring functions by incorporating crystallographic binding poses into training data.

A gradient boosted decision tree model called CrystalBoost was developed by combining over 5000 crystal poses with the training data from SCORCH, a machine learning scoring function that showcases high performance. It was hypothesised these additional accurate binding poses would enhance model generalisability and docking power compared to the original SCORCH model.

CrystalBoost demonstrated superior performance over SCORCH on internal test sets. However, on the external DEKOIS 2.0 benchmark, CrystalBoost under-performed, indicating possible overfitting. A variant SCORCH model integrating CrystalBoost also showed reduced generalisability. Neither CrystalBoost nor the SCORCH model with CrystalBoost integrated exhibited improved docking power.

In this project, augmenting the training data with crystal poses didn't enhance the original SCORCH model as anticipated. While the initial hypothesis was unsupported, unexplored avenues, like using diverse decoy sets or multiple crystal poses per ligand, might still show potential benefits.

This project highlighted that simply increasing training data size does not guarantee improved performance. Careful data curation and model tuning is imperative. While the current work did not observe significant advantages from crystal poses, its potential could be unlocked when combined correctly with computational insights. This project contributed to the broader understanding of machine learning scoring function design principles.

Table of Contents

Abbreviations

VS - Virtual Screening

SBVS - Structure-Based Virtual Screening

LBVS - Ligand-Based Virtual Screening

SF - Scoring Function

MLSF - Machine Learning Scoring Function

FFNN - Feed-forward Neural Network

RF - Random Forest

GBDT - Gradient Boosted Decision Tree

SMILES - Simplified Molecular-Input Line-Entry System

RMSD - Root Mean Square Deviation

PDB - Protein Data Bank

MCS - Maximum Common Substructure

O3A - Optimal 3D Alignment

MMFF94s - Merck Molecular Force-Field 94 Static

UFF - Universal Force-Field

SDF - Structure Data Format

AUCPR - Area Under the Curve of the Precision-Recall

ROC-AUC - Area Under the Curve of the Receiver Operating Characteristic

EF - Enrichment Factor

DEKOIS - Demanding Evaluation Kits for Objective In Silico Screening

CSAR - Community Structure-Activity Resource

1. Introduction

1.1 Protein-Ligand Interactions in Drug Discovery

Bringing a new drug to the market is a substantial commitment, typically necessitating an average investment of $2.6 billion and a duration of 12 years [1]. Central to this process is the study of protein-ligand interactions. Currently, accurate prediction of these protein-ligand interactions necessitates wet lab experiments which contribute significantly to both the expense and duration of drug discovery. Computational methods currently exist to predict these interactions, but none are accurate enough to replace the need for wet lab experimentation. Therefore, there's a pressing need for a refined virtual screening (VS) approach that might mitigate the reliance on labor-intensive wet lab tests. If such a VS method could be developed, it could drastically accelerate drug discovery and reduce associated costs substantially.

A crucial aspect of the drug discovery process is identifying ligands with a high affinity for the drug target; these high-affinity ligands are known as lead compounds. Identifying lead compounds remains a challenge due to the intricate interactions of genomics and chemistry [1]. Virtual screening can assist in this process. There are two main branches of VS, Structure-Based Virtual Screening (SBVS) and Ligand-Based Virtual Screening (LBVS). LBVS takes a ligand-based approach by attempting to identify similarities between known active molecules, and a database of compounds to identify new drug targets. In contrast, SBVS predicts binding affinity by modelling the ligands onto the 3D structure of the target protein [2]. SBVS is the focus of this dissertation.

In SBVS, Scoring Functions (SF) are essential computational tools. They are used to computationally assess the binding affinity between protein-ligand complexes [3]. SF's

rank ligands based on predicted binding strengths. This is accomplished by positioning the ligands within a protein's binding site using docking programmes before the SF evaluates the pose's alignment and binding potential, factoring in crucial protein-ligand interactions.

1.2 Classical Scoring Functions: Limitations & Challenges

Classical Scoring Functions are linear functions that estimate protein-ligand binding affinity based on their structural and physicochemical attributes. These are broadly categorised into force field-based, empirical, and knowledge-based SFs [3].

Force-field based SFs calculate interaction energies between proteins and ligands, accounting for factors such as van der Waals forces, electrostatic interactions, and solvation energies. Their limitations stem from high computational demands and approximate nature which may not capture intricate protein-ligand dynamics [4]. Empirical SFs combine various interaction types such as hydrogen bonds, hydrophobic interactions, and electrostatics in a weighted manner to determine protein-ligand binding strength. These SFs often suffer from oversimplification and are compromised by inconsistent experimental data quality [5]. Knowledge-based SFs rely on known information from known protein-ligand complexes [4]. Knowledge-based SFs derive potential energy using a reference state that treats proteins and ligands as random collections of atoms. This simplification can overlook the inherent structured bonds within molecules, potentially compromising the reliability of the scoring function [5].

While classical SFs have significantly informed protein-ligand interaction predictions, their inherent limitations are prompting exploration of new methods, such as machine learning, to refine the drug discovery process.

1.3 Machine Learning Scoring Functions

Machine learning scoring functions (MLSFs) are a notable advancement over classical SFs for predicting protein-ligand interactions. MLSF models can capture the intricate, nonlinear relationships between proteins and ligands that classical SFs might overlook [6]. Furthermore, MLSFs have the ability to automatically identify features for modelling protein-ligand interactions; this reduces the need for manual feature engineering [7]. Benchmark studies have shown improved performance in MLSFs, especially gradient boosted decision trees (GBDT) and random forest (RF) models [8]. Deep learning models, including convolutional neural networks and graph neural networks, are capable of processing high-dimensional data and capturing both localised and global structural details [4, 9]. MLSFs can enhance binding affinity predictions by incorporating a variety of data types, such as structural and physicochemical characteristics [10]. Additionally, these algorithms can be easily updated and improved as more data becomes available.

1.4 SCORCH & Other Machine Learning Scoring Functions

Recently several MLSFs have emerged, including RF-ScoreVS v2, NNScore 1.0, and SCORCH [3, 11, 12]. The emphasis will be placed on SCORCH in this section, as it is the model this project aimed to improve upon.

NNScore 1 is an early example of a MLSF, employing a feed-forward neural network (FFNN) to predict protein-ligand interactions. FFNNs consist of a number of layers of artificial neurons; an input layer, a variable number of hidden layers, and an output layer, with the connections between layers being weighted. A visual explanation of the architecture of FFNNs can be seen in Figure 1. However, a limitation in NNScore's design is the omission of decoy ligands during training. Such decoys, mimic the physiochemical properties of active ligands, while differing in ways that ensure they are inactive. These decoys serve to train the model in differentiating between binders and

non-binders [3]. The lack of decoys in the training data for NNScore means its use in real-world scenarios is limited.

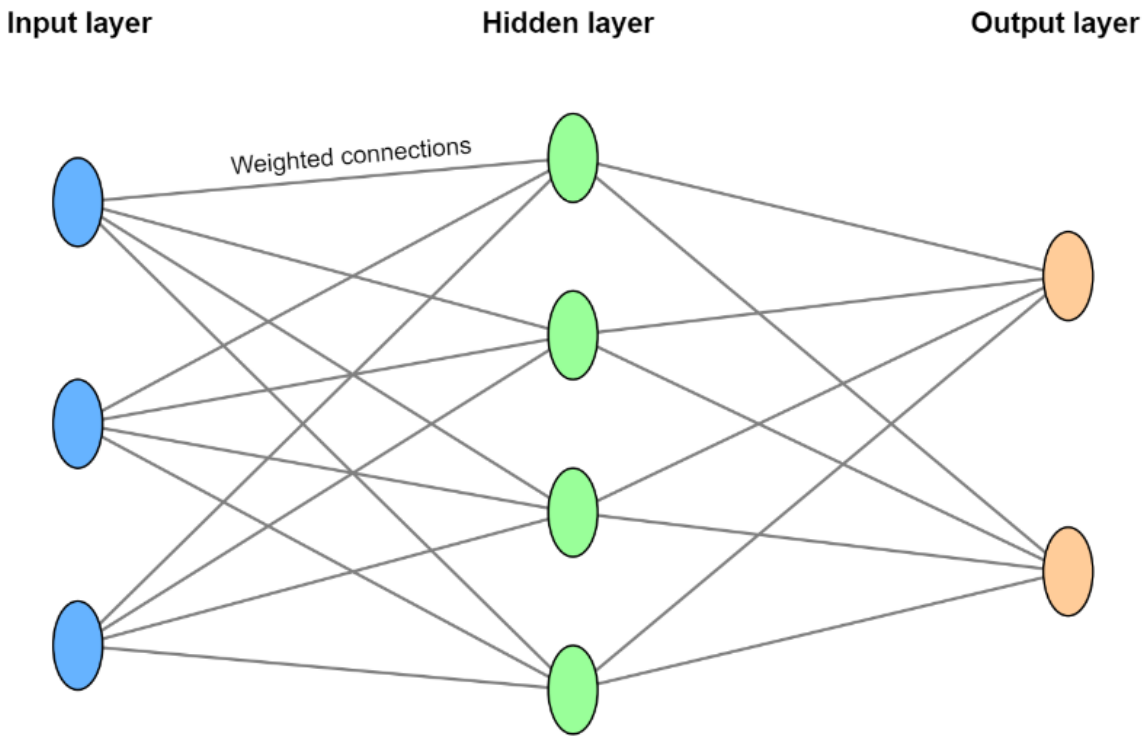


Figure 1: Visual representation of the architecture of a feed forward neural network model, created with a custom python script using matplotlib [13].

RF-Score-VS uses a random forest (RF) model as its foundation. RF models integrate multiple decision trees, each developed from random data subsets, with their combined predictions forming the final outcome. A visual representation of an RF model can be seen in Figure 2. RF-Score-VS trained with decoy ligands from the DUDE-E data set [11], improving its ability to discern true binders from non-binders. However, the DUD-E data set has known biases, as such the SF may learn these biases, compromising its predictive accuracy [3, 14].

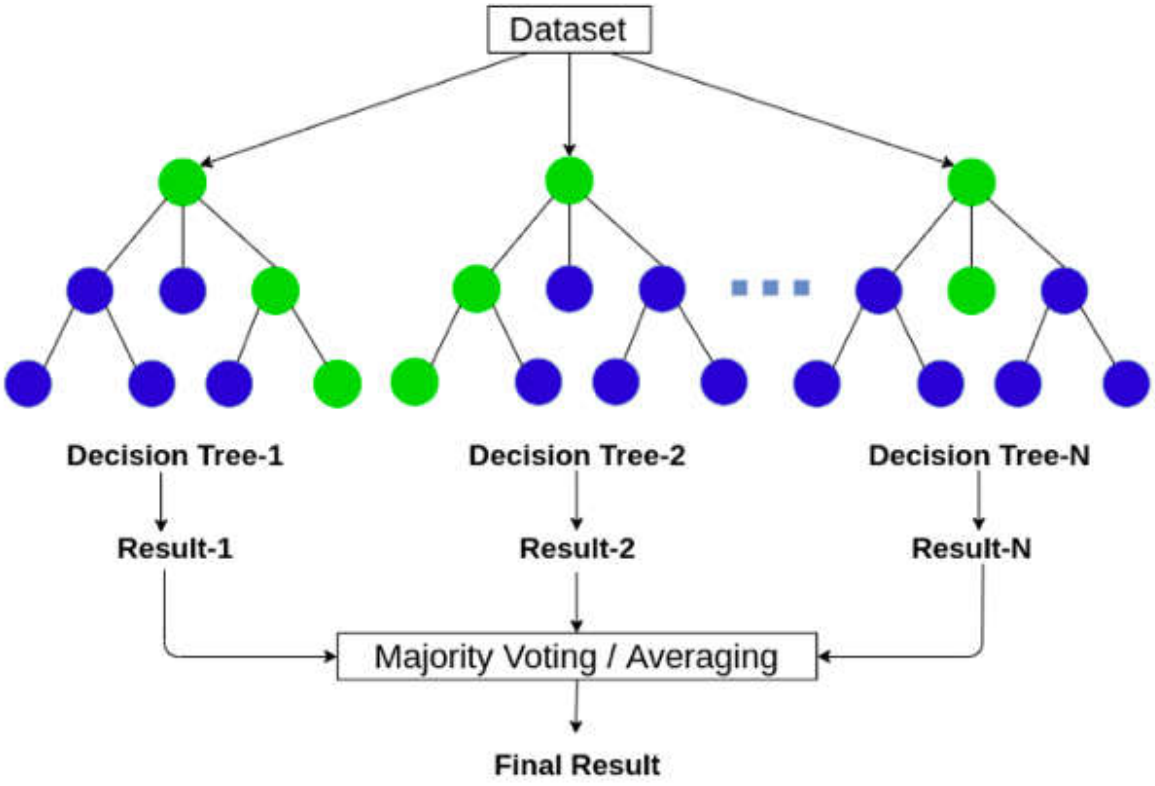


Figure 2: Visual representation of a random forest model, taken from [15]. Each individual decision tree provides a prediction, aggregation of these predictions leads to the final result.

SCORCH is the most important MLSF in the context of this project, with the objective being to improve the functionality of this MLSF. SCORCH differs from the other two models in some key ways. Most notably, instead of relying on a single type of ML model, SCORCH employs three different models; an feed-forward neural network, a wide-and-deep neural network, and a gradient boosted decision tree (GBDT). GBDTs operate through sequential decision trees, with each tree correcting the previous tree's errors. A visual representation of a GBDT can be seen in Figure 3. The wide-and-deep neural network consists of two components: the 'wide' part captures straightforward patterns quickly, whereas the 'deep' component learns intricate relationships over extended periods of time. SCORCH derives a consensus from the three models to provide a prediction. This type of consensus approach has proven benefits in literature [16]. Furthermore, in order to avoid the bias inherent in the DUD-E decoy data set, SCORCH makes use of a novel machine learning method of decoy generation

known as DeepCoy [17]. Lastly, SCORCH evaluates multiple poses for each ligand in its training data, labelling each pose based on its RMSD relative to the crystal pose (X-ray crystallography determined ligand binding pose). This ensures that during training, only true binders in the correct conformation are prioritised. [3].

SFs are assessed based on three primary metrics: screening power, ranking power, and docking power. Screening power is a measure of the SF's capacity to predict whether a ligand will bind to the target protein. Ranking power measures the SF's ability to correctly rank ligands by their affinity for the target protein. Docking power is the SF's performance in predicting the correct pose adopted by the ligand upon binding to the protein. In the initial phases of drug discovery, screening power is crucial, as it helps to filter and prioritise ligands for further experimental evaluation. On both an internal test data set, and independent benchmark tests, SCORCH consistently demonstrated superior performance on all three metrics compared to other scoring functions [3].

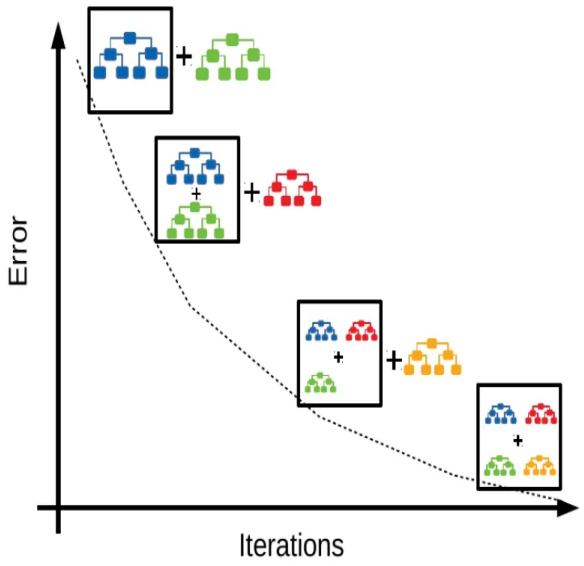


Figure 3: Visual representation of a gradient boosted decision tree model, taken from [18]. Each decision tree improves upon the prediction of the previous tree until a final result is produced.

1.5 CrystalBoost: A Machine Learning Scoring Function

The aim of this project was to improve the performance of the SCORCH scoring function. To accomplish this task, it was decided that a promising avenue of improvement lay in incorporating crystal poses into the training data of SCORCH.

Crystallographic poses are commonly used to gain insights into protein-ligand interactions. These poses provide an experimentally determined insight into a ligand's conformation within a protein's active site [19]. While they are often referenced in experimental contexts for their detailed depiction of ligand orientation, they also highlight the ligand's conformation during binding. Incorporating crystallographic poses into training data, alongside computationally derived poses like those from GWOVina [20] used in SCORCH, can diversify the dataset and potentially enhance model generalisability.

A cautious approach was necessary to include the crystal poses into the training data to avoid introducing bias. In the original SCORCH paper, a great effort was undertaken to ensure no bias crept into the data set. This included pre-processing all data identically, and deriving all ligands included in the training data from simplified molecular-input line-entry system (SMILES) strings. To integrate the crystal poses, ligands were first derived from their SMILES strings, producing various conformers. Conformers refer to the different spatial arrangements of a molecule that result from the rotation about a single bond. These generated conformers were then aligned to the original crystal pose, and the pose with the lowest RMSD was retained for each ligand. To minimise variables, the same features were computed and selected for the protein-ligand interactions as the original SCORCH paper. Additionally, it was decided that the GBDT model of SCORCH would be the model selected for enhancement due to two main reasons: its superior performance compared to other models within SCORCH, and the relatively shorter training time, making it more suitable within the project's timeframe. These processes culminated in the development of a new gradient boosted decision tree model named "CrystalBoost".

By supplementing the training data with crystal poses, it was expected that CrystalBoost would have enhanced generalisability compared to SCORCH. The inherent accuracy of these crystal poses, depicting true binding scenarios, was anticipated to provide the model with a broader perspective of genuine ligand-binding conformations. As such, this should improve performance metrics, especially in independent benchmark tests. It's hypothesised that a more generalised model, trained with diverse and accurate data, would exhibit superior performance particularly on independent data sets. Furthermore, the inclusion of crystal poses should theoretically improve the model's docking power, as providing experimentally determined examples of the poses taken by ligands upon binding should allow the model to refine its predictions of ligand conformations upon binding.

2. Materials & Methods

All Python scripts used in this project are available in the GitHub repository linked in the appendix. All Python scripts prior to model training and evaluation were ran in a Python 3.8.17 Anaconda environment, with the exception of MGLTools 1.5.7 scripts, which were ran in a Python 2.7.3 Anaconda environment. Model training and evaluation Python scripts were ran in a Python 3.6.15 Anaconda environment [21].

2.1 Data Sets

2.1.1 The SCORCH Data Set

The SCORCH data set is the data used to create the original SCORCH model [3]. This data consists of 5102 active ligands and 51,020 DeepCoy generated decoys. All active and decoy ligands were docked to their respective receptor with GWOVina, to produce a final data set of 75,859 protein-ligand pose complexes.

2.1.2 The Crystal Pose Data Set

The crystal pose data set consists of the 5084 active ligand crystal poses. All of these active ligands were used in the SCORCH data set. To guarantee that the new model remained unbiased, it was crucial to prepare the crystal pose data set in the same way the SCORCH data set was prepared. Thus, these crystal poses were derived from simplified molecular-input line-entry system (SMILES) strings, then prepared via the steps detailed in Section 2.2.

2.2 Data Preparation

2.2.1 Crystal Pose and Receptor PDB File Retrieval

The 5102 active ligands used in the SCORCH data set were downloaded from PDBBind, BindingMOAD and Iridium [22–24]. The PDBBind ”The Refined Set”, the BindingMOAD ”Non-Redundant Set” and the Iridium ”Highly Trustworthy Set” were used. Some protein-ligand complexes were missing from the BindingMOAD and PDBBind data sets. These missing complexes, with the PDB IDs 1RE8, 1TFZ, 1VJY, 2AZ5, 2PEL, 2X2M, 3G5D, 3PVW, 3RI1, 3SFI, 4ZJR, 4XRQ, 5E95, 5OKT, 5UEX, 5W7U, 6FP4, 6NLK, 6OA3, 6PGU, 5YJ0 and 4OCT, were present in the ”All of Binding MOAD” data set and were obtained via the *binding_moad_prep.py* script.

To process the data sets and convert to Protein Data Bank (PDB) file format, a suite of Python scripts, originally obtained from the XGBScore GitHub repository [25], were employed and subsequently modified. The *binding_moad_prep.py* script was used to retrieve the PDB files for the BindingMOAD data set. For the removal of water atoms and isolation of ligands and receptors across the PDBBind, BindingMOAD, and Iridium data sets, the scripts *PDB_pocket_isolator.py*, *parallel_moad_pocket_and_ligand_isolator.py*, and *Iridium_pocket_isolator.py* were used respectively. Within this context, receptors were defined as all atoms located within a 14 Å radius of the bound ligand.

2.2.2 Canonical SMILES Generation

To prevent bias and ensure data preparation remained consistent with the original SCORCH model, database crystal pose PDB files could not be used directly to train the model. Thus, RDKit 2023.3.2’s *MolFromPDBFile* and *MolToSmiles* functions [26] were used in *rdkit_2023_SMILES_generator.py* to generate canonical SMILES strings for all active ligands.

2.3 Conformer Generation

Conformer generation was carried out using a modified version of the *rdk_confgen.py* Python script obtained from the rdk_confgen GitHub repository [27]. This script was modified to accept SMILES strings as inputs instead of mol2 files, to accommodate the crystal pose SMILES generated previously. This conformer generation script utilises RDKit 2023.3.2 [26] to generate conformers, then Merck Molecular Force-Field 94 static (MMFF94s) was used for optimisation to find the most stable 3D structure for the conformer. Due to a significant proportion of the active ligands having $\geq$ 13 rotatable bonds, 300 conformers were generated per active ligand as per the recommendations by Ebejer *et al.* [28]. Conformers were saved to multi-conformer Structure Data Format (SDF) files. Due to incompatibility with the conformer generation technique, 11 active ligands were removed from the data set leaving a set of 5091 total active ligands. The active ligands removed had the following PDB IDs: 1W3K, 5J41, 1W3L, 4UND, 1V0K, 5LYR, 4V27, 4CD4, 4AD3, 4AD2, and 1FH7.

2.4 Alignment to Crystal Pose

The original crystal pose PDB file was converted to SDF format for alignment using RDKit 2023.3.2's *MolFromPDBFile* and *MolToMolFile* functions. All conformers were aligned to their original crystal pose using the Python script *MCS_rdkit_sdf_alignment.py*. This script utilises the Maximum Common Substructure (MCS) alignment method by taking advantage of RDKit 2023.3.2's *FindMCS* and *AlignMol* functions. This method works by finding the maximum common substructure between the conformer and the crystal pose, then aligning the conformer to the crystal pose based on this MCS. The Root Mean Square Deviation (RMSD) of each conformer to the crystal pose was saved and an average RMSD across all conformers was calculated. For each ligand, the conformer with the lowest RMSD to its respective crystal pose was saved to a PDB file. The mean RMSD of each of these lowest-RMSD conformers was

also calculated. These aligned conformer files were then used as the crystal poses for model training in this project.

2.5 Data Set Splitting

The PDBQT files were divided into three distinct data sets: training (n = 4115), testing (n = 510), and validation (n = 459). To avoid potential biases, the same data splits as used in the original SCORCH paper were maintained. This ensured that no overlap occurred between the sets, as any deviation might lead to unintentional data leakage where ligands previously utilised for training could be assigned to the test set and validation sets, or vice versa. Furthermore, by adhering to the previously established splits, the same data stratification was preserved. This stratification was based on structure resolution and dissociation constant, guaranteeing uniform distributions of these characteristics across the training, testing, and validation sets [3].

2.6 Feature Computation, Selection & Scaling

The SMILES derived crystal poses and the receptor PDB files were converted to PDBQT file format using MGLTools 1.5.7 [29]. MGLTools was also used to add Gasteiger charges, add hydrogens, and merge non-polar hydrogens with their parent atoms. The Python scripts *prepare_ligand4.py* and *prepare_receptor4.py* from MGLTools were used to accomplish this task.

A modified version of the *scorch.py* script [3] was used to calculate, select and scale the features used to train the CrystalBoost model. This SCORCH script used BINANA 1.3 [30] to calculate 531 molecular attributes such as ligand atom types, the number of rotatable bonds (nRot), the summation of pairwise electrostatic energies, and to examine various molecular interactions, such as hydrogen bonds and salt bridges. It

also calculates Extended Connectivity Interaction Features (ECIFs) for each protein-ligand pairing as described by Sánchez-Cruz *et al.* [31]. These ECIFs provided insights into atomic characteristics like symbol, valency, and connectivity patterns, resulting in a comprehensive set of 1,540 ECIFs being generated. Additionally, Kier flexibility, the ability of a molecule to bend or twist due to the interactions and arrangements of its atoms and bonds, was also calculated. [32]. A full description of the features calculated can be seen in the original SCORCH paper [3].

seven ligands were incompatible with the feature computation functions, and were thus removed from the data set. The PDB IDs of these ligands are: 5EWA, 5EV8, 5EW0, 5EVB, 5EVK, 5EW9, 5EVD. This left a total of 5084 ligands in the total data set.

After feature computation, feature selection was also carried out by the *scorch.py* Python script, using pre-defined features previously selected for in the original SCORCH paper. This involved selecting only the most important protein-ligand interaction features for use in training the model. These features were also scaled to between 0 and 1 by the Python script [3].

2.7 Pose Labelling

In the SCORCH data set, all active ligand poses were labelled based on binding affinity and RMSD to the crystal pose. Strong binders with an RMSD of < 2 Å were assigned a label of 1, strong binders with an RMSD of > 4.5 Å were assigned a label of 0. All weak binders and decoys were labelled 0. As only crystal poses were present in the crystal pose data set, ligands only needed to be labelled based on binding affinity. Binding data was obtained from PDBBind, BindingMOAD or Iridium depending on the source of the protein-ligand complex, and the python script *ligand_labelling.py* was used to label the crystal pose data set. A label of 1 was assigned to ligands with a K_i, K_d, or IC_{50} of $\leq 25\ \mu M$. All other ligands were assigned a label of 0.

2.8 Model Training

The Python script *XGBoost_training.py* was used with XGBoost 1.4.2 [33] to train all Gradient Boosted Decision Tree (GBDT) models in this project. The train, validation and test data sets from the SMILES derived crystal poses were combined with the SCORCH train, validation and test data sets. The combined final data sets contained a total of 80,946 protein-ligand pose complexes, 65,527 of which were used to train the models.

Scikit-learn 0.24.2's [34] function *GroupKFold* along with sci-kit-optimize 0.9.0's [35] *gp_minimize* function were used to carry out 5-fold cross-validation for hyperparameter tuning. This process identifies the ideal settings for the XGBoost model training function. *GroupKFold* ensures that the poses from the same ligand are not present in both the training and validation data sets during cross-validation, preventing data leakage. The *gp_minimize* function then searches for the best hyperparameters using Bayesian optimisation, testing each set of hyperparameters using the data set splits created by *GroupKFold*.

The hyperparameter search space was adjusted between the training of each model. Once the optimal hyperparameters were identified, the model was trained using 20,000 boosting rounds, with early stopping of 400 epochs on the validation test set. Early stopping was used to stop the training process once model performance was no longer improving on the validation set to prevent overfitting; a phenomenon in which a model learns the training data too closely, resulting in decreased performance on unseen data. The final model was trained on both the training and validation test sets, with the optimal number of boosting rounds identified by the early stopping training.

2.9 Model Evaluation

All models produced were evaluated by plotting a learning curve that shows how the model improved with each training step, comparing results from the training data and validation set. The learning curve plots the model's Area Under the Curve of the Precision-Recall (AUCPR) at various rounds of model training. This step was carried out prior to use of the validation set in the training of the model. It can be used to assess the model's learning rate, and judge if the model is overfitting or underfitting by comparing the model's performance on the test set to the validation set. For each model trained, it's AUCPR on the test data set was recorded; this was also used as an evaluation metric.

2.10 Model Comparison

Four different models were compared in order to judge the impact of including the crystal poses in the training set on model performance.

- The best-performing GBDT model (CrystalBoost)
- The original SCORCH model
- The stand-alone original SCORCH GBDT model
- The SCORCH model with CrystalBoost in place of the original SCORCH GBDT model

Two internal test data sets and an external benchmark data set were used to compare the screening power of the different models, while docking power was evaluated on a single external benchmark data set. During screening power evaluation, the highest score from all the considered poses determined the score for the respective protein-ligand complex. Thus, if any of the poses achieved a high score, the ligand was deemed a binder. Conversely, if no pose achieved a high score, the ligand was classified as a non-binder. This is in line with the methodology used in the original

SCORCH paper [3].

Screening power was measured using three different performance metrics, while docking power was assessed in a pose-ranking manner. The details of screening power and docking power evaluations are discussed below.

2.10.1 Evaluation Metrics

Area Under the Curve Precision-Recall

Area Under the Curve Precision-Recall is a commonly-used evaluation metric for binary classifiers, such as the scoring function models tested in this project. The metric is derived from the precision-recall curve, which plots precision against recall for varying decision thresholds. In this case, the threshold is the score returned by the model above which a ligand is considered a binder. Precision is a measure of how many of the identified positive cases are true positives. Recall determines the proportion of actual positive cases that the model correctly identified.

$$Precision = TP/(TP+FP)$$

$$Recall = TP/(TP+FN)$$

TP = True Positive, FP = False Positive, and FN = False Negative

AUCPR summarises the information from the precision-recall curve into a single value, offering an overall assessment of the model's performance. The closer the AUCPR is to 1.0, the better the model's performance. In this project, AUCPR was the most important metric, as it is particularly good at measuring a model's ability to distinguish

a positive class (binder) from a larger number of negatives (non-binders) [36].

In this project, the Python script *model_evaluator.py* was used to plot a Precision-Recall curve using Scikit-learn 0.24.2's *precision_recall_curve* function and to calculate AUCPR using Scikit-learn 0.24.2's *average_precision_score* function for all models.

Receiver Operating Characteristic - Area Under the Curve

The Receiver Operating Characteristic curve (ROC Curve) and Area Under the ROC Curve (ROC-AUC) provide insights into a model's ability to differentiate between positive and negative classes across various decision thresholds. The ROC curve plots the true positive rate (sensitivity) against the false positive rate. The ROC-AUC quantifies the overall ability of the model to distinguish between classes. A perfect model would achieve an ROC-AUC of 1.0.

While in this project the AUCPR is the most important metric, the ROC-AUC provides a more intuitive plot that serves to highlight the ability of the models. The ROC-AUC is sensitive to imbalanced data sets like the ones used in this project, as such it can be too optimistic in its measure of a model's performance. AUCPR accounts for this imbalance, making it's evaluation more accurate.

Enrichment Factor

Enrichment Factor (EF) measures the ability of a virtual screening method to distinguish active compounds from a data set. It compares the concentration of true active compounds in the top x% of a ranked list to what would be expected from a random selection. Consider the top 1% of molecules sorted by a scoring function; EF determines how many times more actives are present in the scoring function ranked list compared to a randomly ordered one. For example, an EF value of 1 suggests a performance analogous to random selection, whereas an EF of 5 indicates that the scored top 1%

contains five times as many active molecules [37].

In this project, the Python script *enrichment_factor.py* was used to calculate EF at the thresholds of 0.5%, 1%, 2% and 5% with Open Drug Discovery Toolkit (ODDT) 0.7's *enrichment_factor* function.

2.10.2 Screening Power

Internal Test Data Sets

The models were evaluated using two distinct internal test data sets. The first set is the one outlined in the original SCORCH paper [3]. The second set is a novel compilation, merging the crystal poses from the previously mentioned test data split with the data from the SCORCH test set. This data set is referred to as CrystalBoost test set. The screening power of all four models was evaluated on both test sets by calculating AUCPR and EF as previously described.

DEKOIS 2.0 Independent Benchmark

Demanding Evaluation Kits for Objective *In silico* Screening (DEKOIS 2.0) is data set designed to act as a benchmark for assessing the performance of virtual screening workflows [38]. The DEKOIS 2.0 data set used in this project is identical to that used to benchmark the performance of SCORCH [3], consisting of 18 diverse protein targets, with 1200 decoys and 40 actives present for each target, selected from the full DEKOIS 2.0 data set [39]. The screening power of all four models was evaluated by calculating AUCPR and EF as previously described.

2.10.3 Docking Power

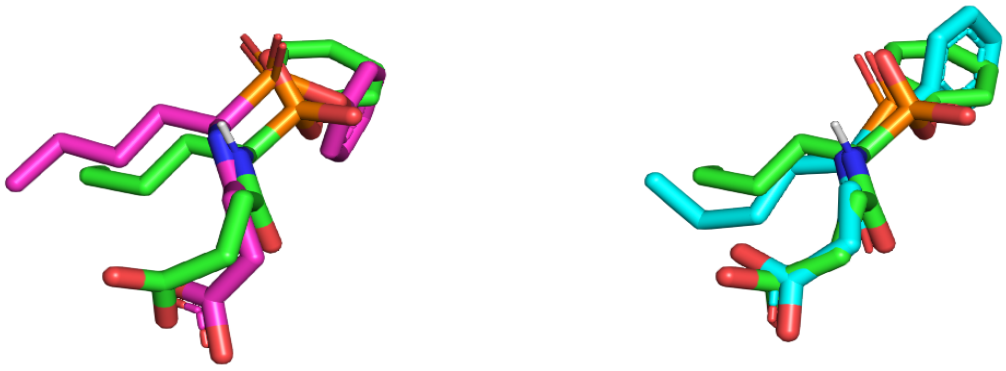
CSAR 2014 Native Pose Identification Benchmark

The CSAR (Community Structure-Activity Resource) benchmark is a collection of data sets used to benchmark structure-based drug discovery tools [40]. In this project, the CSAR 2014 native pose identification benchmark was used to asses the docking power of the various models. This benchmark is designed to test the model's ability to accurately predict the original position and orientation of a ligand in a protein's binding site. The data set consists of 22 active ligands for 3 different receptors. For every active ligand there is 1 near-native pose within 1 Å RMSD of the crystal pose, and 199 decoy poses. [3]. All poses were ranked by every model, with the mean rank for the identified near-native poses being calculated to provide an aggregate measure of model performance.

3. Results & Discussion

3.1 Conformer Generation

Conformer generation was an important step in the process of re-making the crystal poses from SMILES strings. A common method for generating a 3D structure from a SMILES string is via RDKit's *MolFromSmiles* and *EmbedMolecule* functions. This method, while effective, often produces a conformation that might not closely resemble the crystal pose due to the vast conformational space a molecule can occupy. Conformer generation, then subsequent alignment back to the crystal pose results in a pose much closer to the original. Additionally, exploring the full diversity of the conformational space is an important process [28]. Molecules usually exist as a collection of three-dimensional shapes that change over time, rather than having just one fixed three-dimensional structure. Their properties and chemical reactivity are closely tied to this ensemble of conformations. To fully understand a molecule's ”structure,” you need to know all the different structures that make up its conformational ensemble. An incomplete set of conformations gives an incomplete picture of the molecule's structure [41]. Thus, exploring the conformational space by generating 300 conformers per molecule, then aligning back to the crystal structure constructs a more biologically relevant picture for the scoring function model. Figure 4 highlights the improved alignment of a conformer generated pose to a crystal pose compared to an *EmbedMolecule* derived pose. The structure of the lowest-RMSD conformer pose is clearly closer to the crystal pose than that of the *EmbedMolecule*.



(a) (b)

Figure 4: (a) PyMol [42] visualisation of *EmebedMolecule* pose (pink) aligned to crystal pose (green). (b) PyMol visualisation of lowest-RMSD conformer pose (cyan) aligned to crystal pose (green).

The final conformer generation was carried out using RDKit 2023.3.2, with MMFF94s force-field optimisation to stabilise the 3D structure of the molecule. 11 of the 5102 ligands were incompatible with this force-field generation method, however this represents only 0.22% of the total data set, leaving 99.77% available for alignment.

When the *rdk_confgen.py* script previously mentioned was modified to use (Universal Force-Field) UFF optimisation, 5101 out of 5102 ligands successfully generated conformers, however MMFF94s derived conformers were still deemed to be favourable. This decision was made upon reviewing the literature surrounding these force-field generation methods. UFF is a more generalised force-field, with the advantage of a broad applicability. This likely explains why more ligands successfully generated conformers, however this generalisability comes at the cost of accuracy [43].

MMFF94s was developed by Merck [44] and was found to be superior to other generalised force-fields in working with small molecules [45]. It has also been found to have equivalent or improved performance when compared to more recently developed force-fields, and is integrated into all leading commercial molecular modeling applications. [46].

To verify the literature consensus, 300 UFF-optimised conformers were generated per ligand and aligned via MCS alignment to their respective crystal structure as described in section 2.4. The mean RMSD for the lowest-RMSD conformer per ligand was observed as 0.853 Å with UFF and 0.847 Å with MMFF94s. While RMSD performance was almost identical, the weight of literature opinion, and the lack of observable benefit in using UFF force-field over MMFF94s justified the loss of 0.22% of the data set, and lead to the latter being chosen to generate the conformers for the final model.

The decision to generate 300 conformers per molecule was done on the basis of the analysis carried out by Ebejer *et al.* [28]. In this paper, multiple experiments were carried out to determine the minimum number of conformer's needed to be generated to create a pose similar to that of the crystal structure. By varying the number of conformers generated between 10-100, a figure of 300 conformers per ligand was obtained if the ligand has $\geq$ 13 rotatable bonds. Ligands with a higher number of rotatable bonds require more conformers generated as the flexibility bestowed by these rotatable bonds increases the potential conformational space [28]. In total, there were 187 ligands in the data set with $\geq$ 13 rotatable bonds. As such, to accommodate these more flexible molecules it was decided to generate 300 conformers per ligand.

3.2 Alignment

Molecular alignment involves superimposing two or more molecular structures to achieve an optimal overlap between them [47]. This technique played a pivotal role in this project. For each ligand, all 300 generated conformers were systematically aligned to the corresponding crystal pose. This alignment served three primary objectives:

- To identify the conformer with the lowest RMSD in relation to the crystal pose, ensuring structural similarity.
- To orient the conformer in accordance with the spatial orientation of the crystal pose.
- To position the conformer correctly within the protein's active site, ensuring that it maintains a biologically relevant stance.

These objectives underpinned the overarching aim of reproducing a biologically relevant crystal pose starting from a SMILES strings. In this project, we evaluated four distinct molecular alignment techniques: RDKit's *MolAlign*, PyMol's *align* function, RDKit's Optimal 3D Alignment (O3A), and RDKit's MCS alignment [48–50]. Each of these alignment techniques functions differently. The *MolAlign* method in RDKit, aligns molecules by minimising the RMSD between their atomic coordinates. The align function in PyMol performs a sequence alignment followed by a structural superposition. RDKit's O3A alignment attempts to minimise RMSD, but differs from the standard *MolAlign* by also other features such as atom type and charge to achieve a more chemically meaningful alignment. Lastly, MCS alignment identifies the largest common substructure between two molecules and uses this shared framework as a basis for alignment.

These four methods were originally evaluated by calculating the mean RMSD of the lowest-RMSD conformer per ligand for each alignment method. Figure 5 clearly shows that O3A had the lowest mean RMSD at 0.18 Å. However, when viewed in PyMol [42], the aligned conformers showed obvious issues. These O3A aligned conformers were

consistently seen to be clipping through the protein surface. Additionally, the ligands were structured very differently to the crystal pose. Figure 6 highlights the problems with the O3A aligned pose, while contrasting it to the MCS aligned pose and crystal pose.

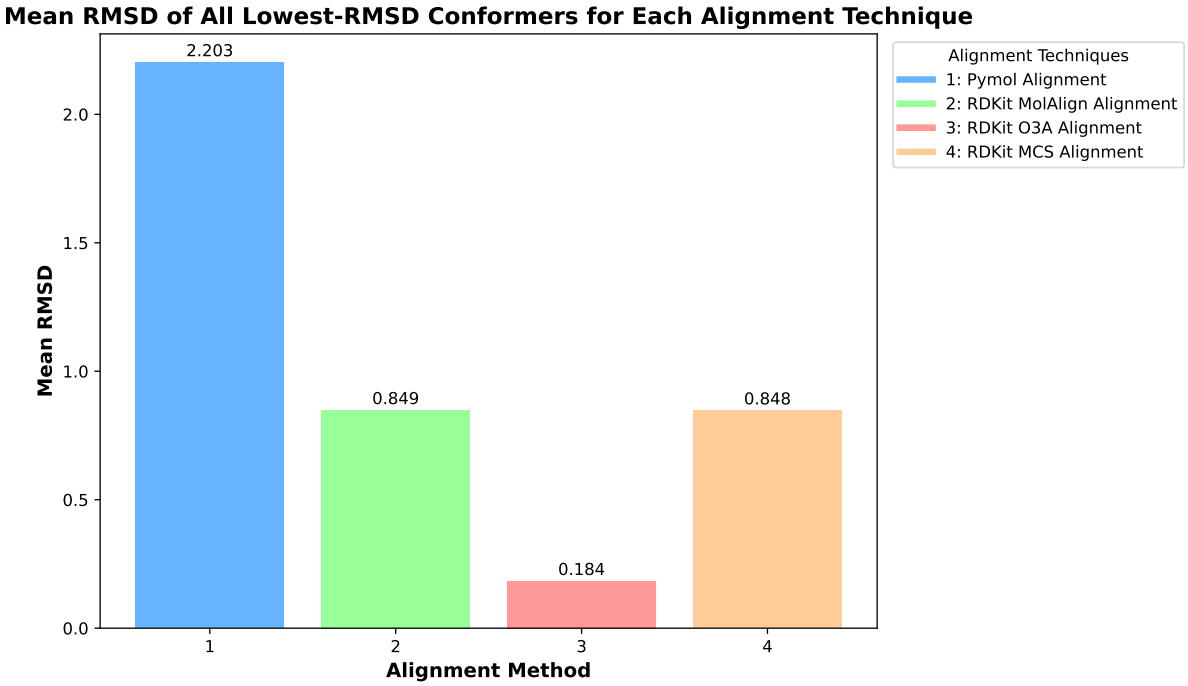
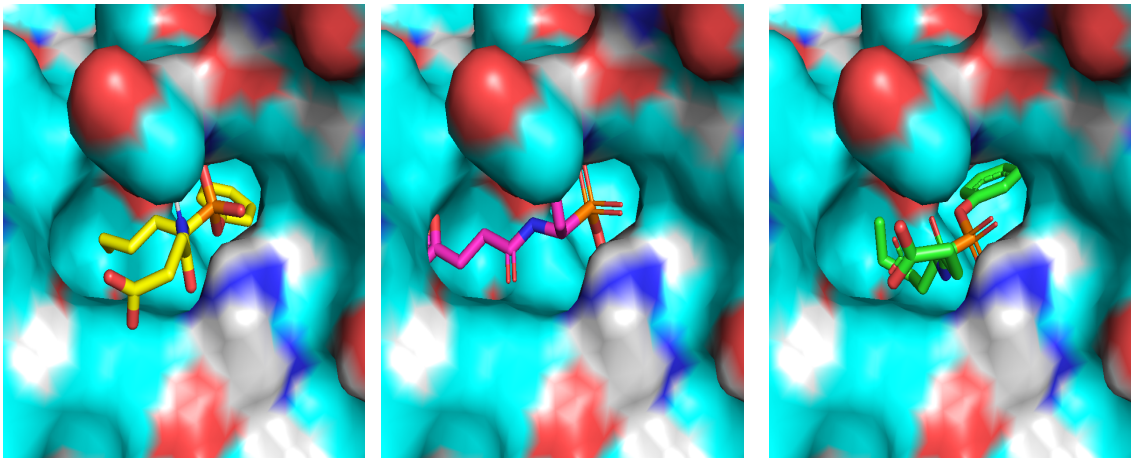


Figure 5: Four alignment methods compared by mean RMSD of the lowest-RMSD conformer per ligand for each alignment method.



(a) (b) (c)

Figure 6: PyMol visualisation of various poses of ligand 1A0Q and its receptor. (a) Crystal pose, (b) O3A aligned pose, (c) MCS aligned pose.

In light of the issues with the O3A poses, MCS aligned poses were chosen. MCS alignment had the second lowest mean RMSD at 0.85 Å, being marginally lower than *MolAlign* alone. Importantly, this figure of 0.85 Å falls within the resolution at which the crystal poses provided by all three databases were measured. This means that both the O3A alignment and the MCS alignment are likely capturing the true binding configurations with similar fidelity, given that both their RMSDs are below the 2.5 Å resolution threshold. Furthermore, the MCS aligned poses were visually similar to the crystal structure, with some being almost identical, as seen in Figure 7.

As an extra measure, an XGBoost GBDT model was trained on O3A aligned conformers with identical parameters as the final MCS aligned conformer model. AUCPR was evaluated using the SCORCH test data set. The O3A model had an AUCPR of 0.90, compared the MCS model's 0.92. This serves to highlight the importance of not solely relying on quantitative metrics like RMSD, but also using qualitative assessments to ensure biological relevance.

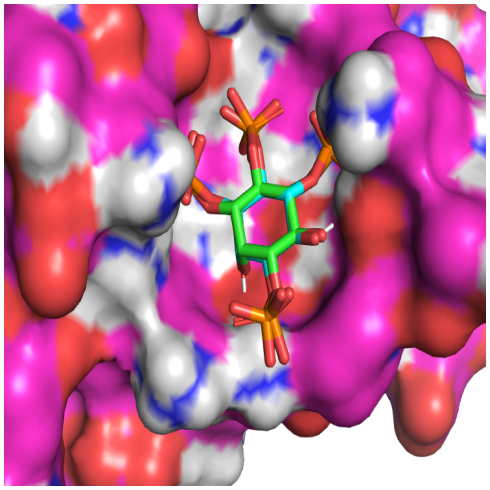


Figure 7: Lowest-RMSD MCS pose (green) aligned to crystal pose (pink) of ligand 1B55 with its receptor.

3.3 Feature Calculation

During the feature calculation phase, seven ligands were identified as being incompatible with the feature calculation script. These ligands uniformly produced the error: "*** *open babel error in tetstereotowedgehash failed to set stereochemistry as unable to find an available bond*." This error is indicative of open babel encountering issues related to the stereochemistry of these ligands. Notably, all seven ligands exhibited structural similarities: each contained either a cyclopentane carboxylic acid or cyclohexane carboxylic acid moiety. Moreover, six out of these seven ligands displayed almost identical overall structures. This similarity suggests that the shared structural features among these ligands might be the underlying cause of the observed error. The similarities between these ligands, can be seen in Figure 8.

Feature calculation is vital step in the creation of an accurate machine learning model. In the context of protein-ligand interaction, the differences between a strong and weak binder can be very subtle [51]. Thus, correctly calculating and selecting the correct features is of paramount importance. These features capture the attributes that describe the physical, chemical, and spatial characteristics of the ligands. For example, these attributes may include bond lengths, hydrophobic interactions or electrostatics. These features enable the prediction of binding affinity and reactivity.

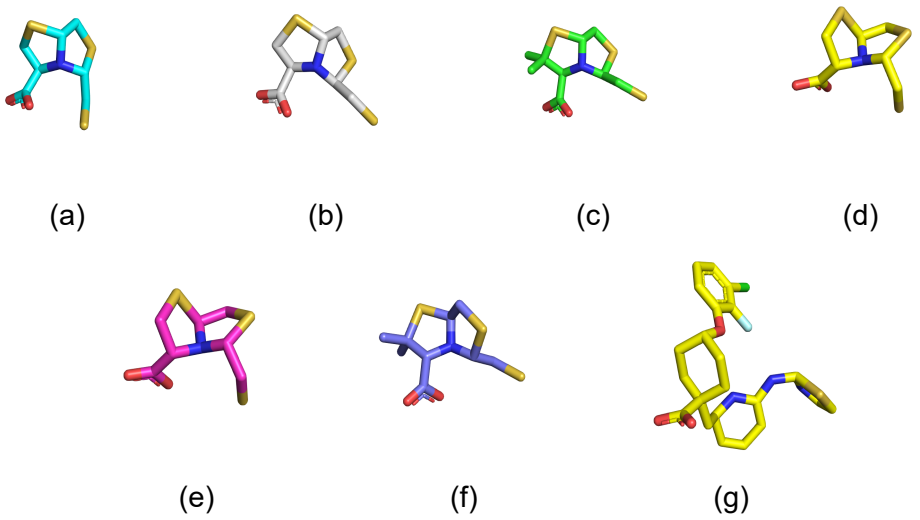


Figure 8: PyMol visualisation of ligand's (a) 5EV8, (b) 5EvB, (c) 5EVD, (d) 5EVK, (e) 5EW0, (f) 5EWA, (g) 5EW9, showing common structural features in error producing ligands.

In the original SCORCH paper, a wide-range of features were calculated for every protein-ligand complex. These features ranged from basic atom-type pairwise counts to intricate extended connectivity interaction features (ECIFs). A rigorous feature selection process was then carried out to identify the most important features that contributed maximally to the model's performance.

For the purpose of this project, it was deemed crucial to maintain consistency with the methodology used in the original SCORCH paper to minimise the variables involved. In order to achieve this, the following precautions were taken during feature calculation and selection:

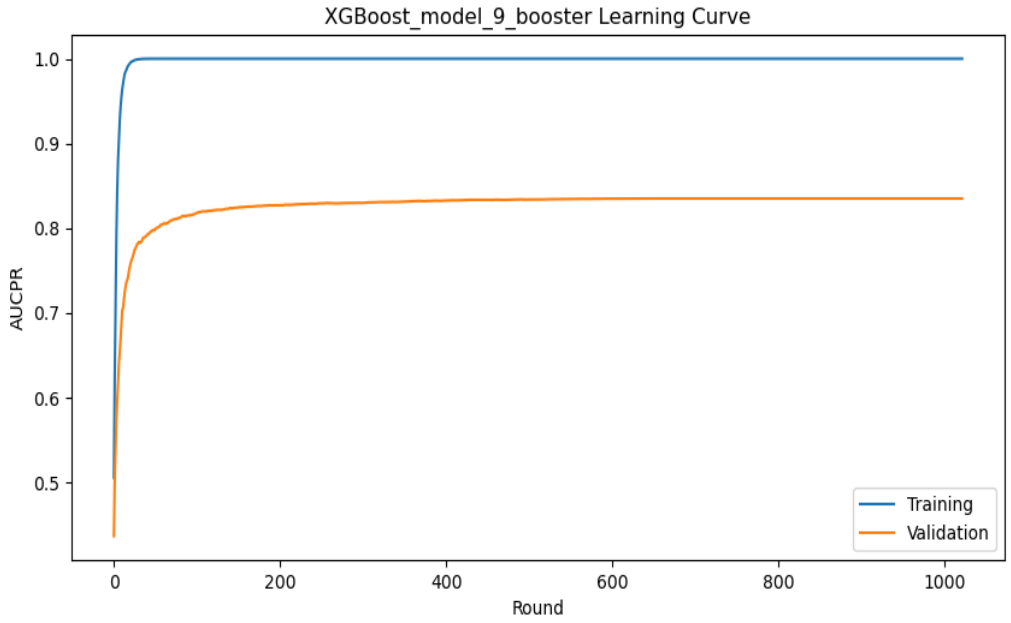
- The *scorch.py*, script was used for feature calculation, ensuring that the features calculated remained consistent.
- The same features were selected, ensuring that the features which were used to train the final model mirrored those from SCORCH.
- The exact same scalar derived from the SCORCH study was applied to ensure absolute consistency in scaling the features between 0 and 1.

By attempting to adhere to the procedures carried out in the original paper as strictly as possible, any deviations in results observed in final model performance can be attributed to the inclusion of crystal poses to the data set, and not differences in methodology.

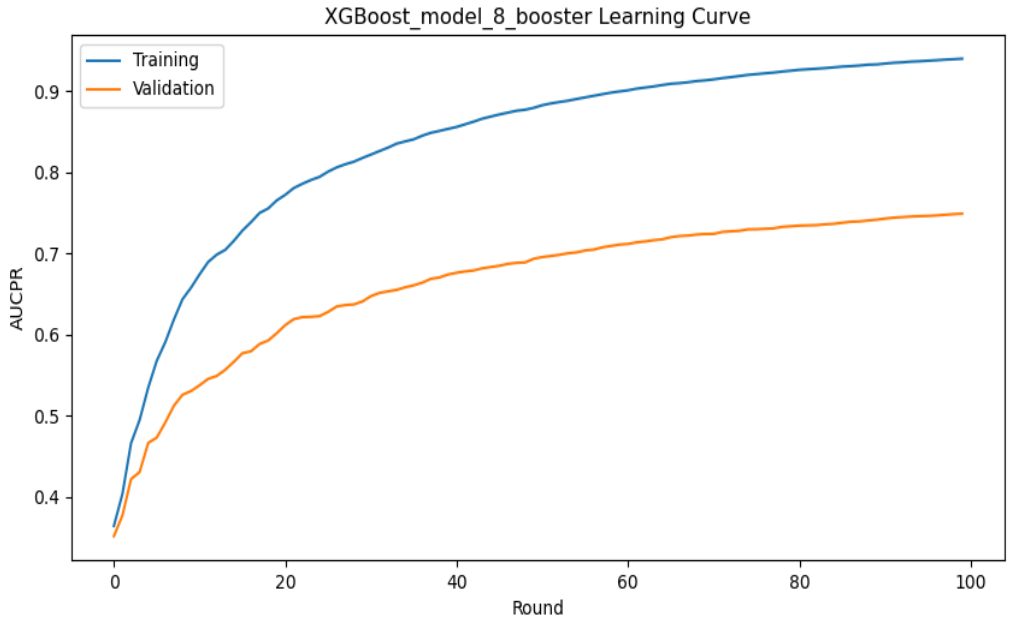
3.4 Model Training & Evaluation

Eleven GBDT models were developed with XGBoost 1.4.2. During training, a learning curve was constructed for each model, juxtaposing AUCPR from the training data set with that of the validation data set over successive iterations. Learning curves are important for understanding how the model is preforming as it is being trained and identifying signs of overfitting. Furthermore, all models underwent evaluation against the CrystalBoost test data set, and both AUCPR and ROC-AUC values were recorded.

After comprehensive review of the results from the validation and test data sets, Model 9, termed ”CrystalBoost”, emerged as the optimal performer and was subsequently selected as the project’s final model. The learning curves of both CrystalBoost and Model 8 are shown in Figure 9. Model 8 was a relatively poorly performing model, and was chosen to highlight the range of outcomes of model training, and to accentuate the ability of the optimal model. A comparison of AUCPR and ROC-AUC values across all models can be found in Table 1.



(a)



(b)

Figure 9: Learning curves for (a) CrystalBoost (Model 9) and (b) Model 8. Validation data set = Orange, Train data set = Blue. The difference in graph shape is due to differences in amount of rounds used for training. Of note, CrystalBoost achieves much higher validation data set scores.

Table 1: AUCPR and ROC-AUC values for all models produced, evaluated on the unseen CrystalBoost test data set. Model 9 (highlighted in yellow) was the final model, dubbed ”CrystalBoost”.

Model	AUCPR	ROC-AUC
Model 1	0.920719	0.920719
Model 2	0.929878	0.929878
Model 3	0.918834	0.918834
Model 4	0.904664	0.904664
Model 5	0.924417	0.924417
Model 6	0.926174	0.926174
Model 7	0.912501	0.912501
Model 8	0.887795	0.887795
Model 9	0.936766	0.936766
Model 10	0.934068	0.934068
Model 11	0.924812	0.924812

The learning curves in Figure 9 highlight the improved performance of CrystalBoost over an the inferior Model 8. By observing the validation curve (orange), you can see the curve reaches a height of approximately 0.8 AUCPR for CrystalBoost, compared to approximately 0.7 AUCPR for Model 8.

The discernible gap between the performance on the test and validation data sets for both Model 8 and CrystalBoost's learning curves is indicative of overfitting. Overfitting is a process in which the model learns the training data too well, showcasing high-performance on the training set, then significantly reduced performance on unseen data sets [52]. Essentially, the model may capture noise specific to the training set rather than the true features determining ligand binding. However, in spite of the visible overfitting, CrystalBoost demonstrates satisfactory performance on both the validation and test sets, leading to its selection as the final model for this project.

An important observation from the learning curves, particularly in Figure 9 (a), is the plateauing of the model's performance on the validation set around 600 rounds, with the training halting at 1000 rounds. This behaviour resulted from the early stopping rounds process used in model training. Early stopping is an important process in attempting to prevent model overfitting. It terminates the training process if there's no improvement

in performance on the validation set after a specified number of successive rounds, or ”epochs” [52]. When a model trains for an extended number of rounds, its performance can decline due to heightened overfitting; thus, implementing such a strategy is vital to prevent this issue. In this case, a threshold of 400 epochs was chosen, meaning when the model failed to improve on the validation set between rounds 600 and 1000, the model training process was halted.

Table 1 highlights that adjusting the hyperparameter search space has quite a small impact on model performance. All models with the exception of model 11 were trained on identical data, with model 11 being trained on the O3A aligned ligands. The only difference between models was the specific hyperparameter search space defined.

A hyperparameter is a predefined value that is set prior to model training to enhance its performance. The search space refers to the specified range or set of these hyperparameters that the model explores to identify the most effective configuration. These hyperparameters were optimised between a range of set values in a process called hyperparameter tuning in this project.

The *gp_minimize* function was used for this tuning process. This function employs Bayesian optimisation to search for the optimal set of hyperparameters. Part of this tuning process involved 5-fold cross-validation, performed with the *GroupKFold* function. 5-fold cross-validation splits the data set into five subsets (folds), training the model on four of them while using the fifth fold as a validation set. The validation fold is then varied across 5 iterations. This process helps to prevent overfitting. A graphical explanation of 5-fold cross-validation can be found in the appendix section A.2. The *GroupKFold* function differs from regular cross-validation by dividing the data set into distinct groups, ensuring that data from any single group doesn’t appear in both training and validation sets simultaneously. In this context, it was used to prevent poses of the same ligand being included in both validation and training sets. Importantly, this

validation set is distinct to the pre-split validation set that is used to create the learning curve.

For each hyperparameter set proposed by *gp_minimize*, the model's performance was assessed on multiple train-validation splits dictated by *GroupKFold*. The average performance across these splits provides an unbiased estimate which is then fed back into the *gp_minimize* function. This approach ensures an effective evaluation of hyperparameters within the defined search space.

Despite signs of overfitting, the CrystalBoost model outperforms the other models created, and thus was selected to be compared to the original SCORCH model to see if any improvement in performance could be observed by the inclusion of crystal poses to the data set.

3.5 Comparison to SCORCH and Other Scoring Functions

The primary hypothesis at the outset of this project posited that the inclusion of crystal poses in the training data for the GBDT model could enhance SCORCH's performance by increasing the scoring function's generalisability. In order to test this hypothesis, four models was compared: The CrystalBoost model, The original SCORCH model, the SCORCH GBDT model, and a modified SCORCH model where its original GBDT was substituted with the CrystalBoost model. The rationale behind the selection of these models was threefold: to to establish a foundational understanding of SCORCH's performance, to compare the CrystalBoost model directly with its predecessor, and to gauge the impact of the CrystalBoost's model inclusion in SCORCH.

To evaluate the performance of CrystalBoost, both the screening power and docking power of all four models were compared directly to one another. Furthermore, to contextualise the model's performance within the broader scope of existing research, the

four models were compared against the third-party scoring functions RF-ScoreVS v2 and NNScore 1. The results of this evaluation are discussed in this section.

3.5.1 Screening Power

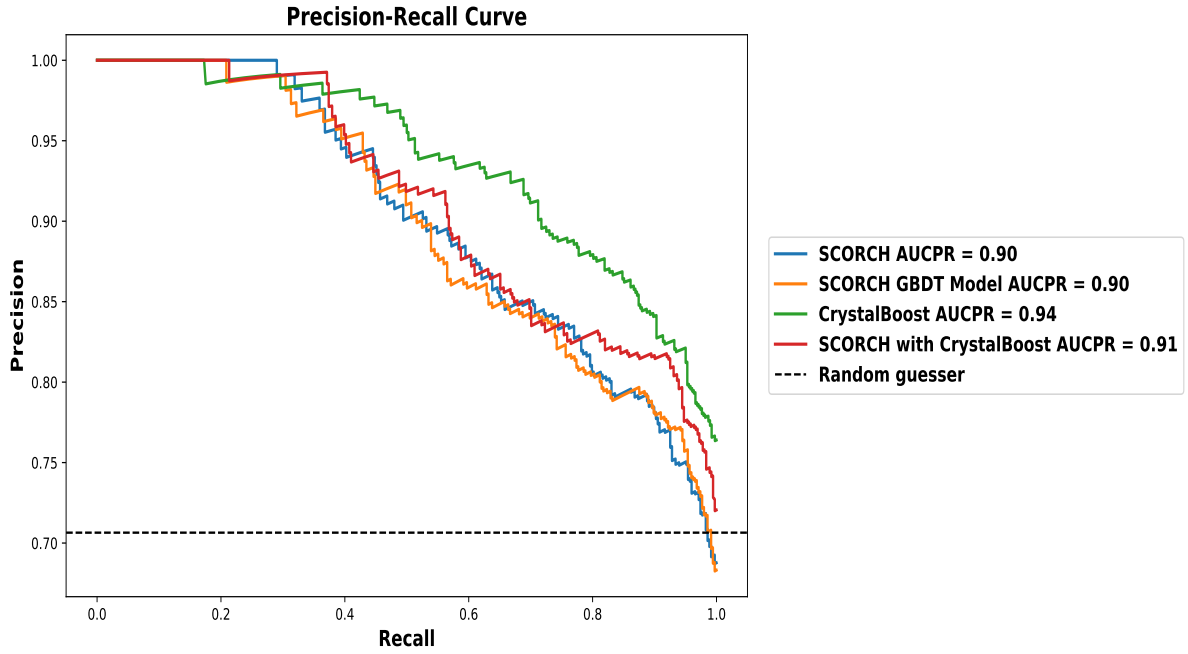
Screening Power on Test Data Sets

Screening power was assessed for both the original SCORCH test data set, and for the new CrystalBoost test data set. Figure 10 shows both the PR curve and ROC curve for all four models on the CrystalBoost test set. While Figure 11 shows the AUCPR of all 4 models on both the CrystalBoost test data set and the SCORCH test data set.

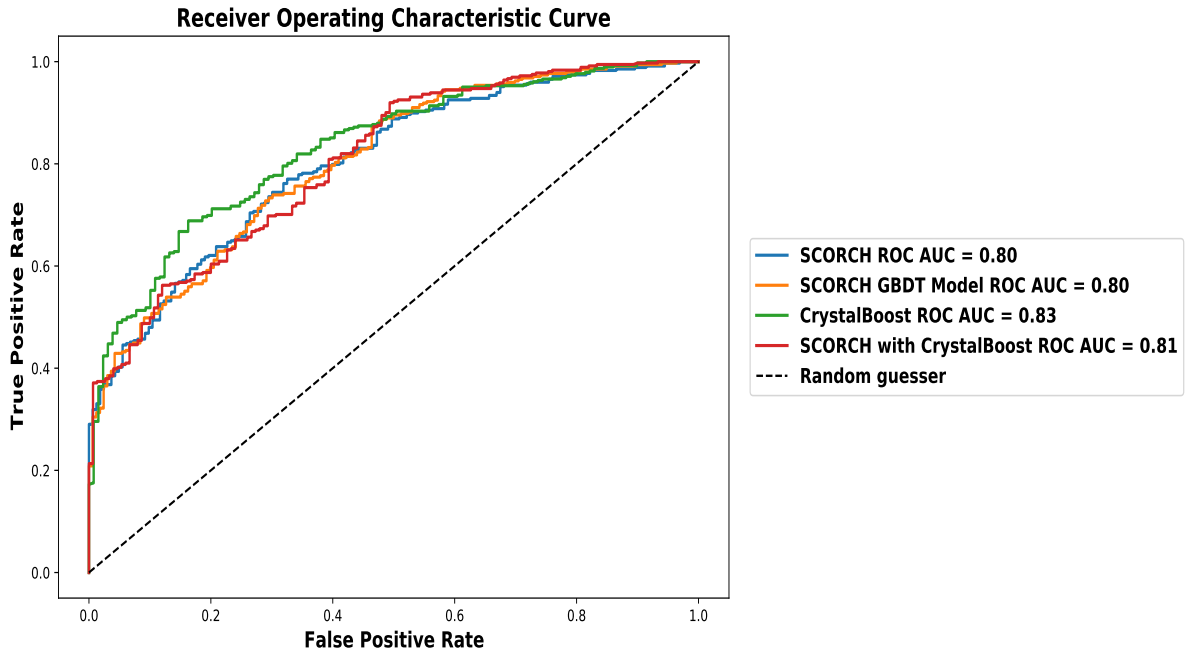
Figure 10 demonstrates that CrystalBoost outperforms the other models. The SCORCH model incorporating CrystalBoost had the second highest performance, while the original SCORCH and the SCORCH GBDT stand alone model shared similar, lesser performances. This result was further evident in the ROC curve, which while not as reliable a measure as AUCPR in this context, shows a more intuitive visualisation of how the performance of each model outperforms a random guesser.

A focus on the AUCPR values, as illustrated in Figure 11, further highlights the performance of the CrystalBoost model on both test sets. Notably, a reduced performance is observed for CrystalBoost when assessed on the SCORCH test data set; yet, it retains its superior ranking compared to the other tested models.

From these results, an initial hypothesis was formed that the enhanced performance of CrystalBoost could be attributed to the inclusion of crystal poses.

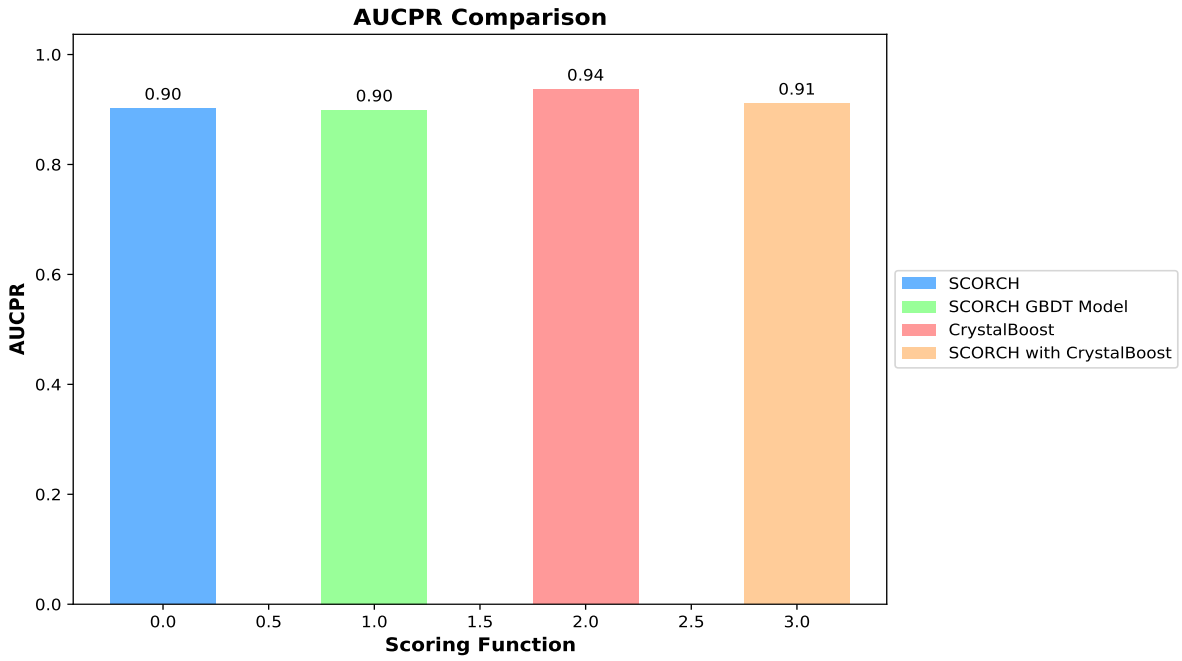


(a)

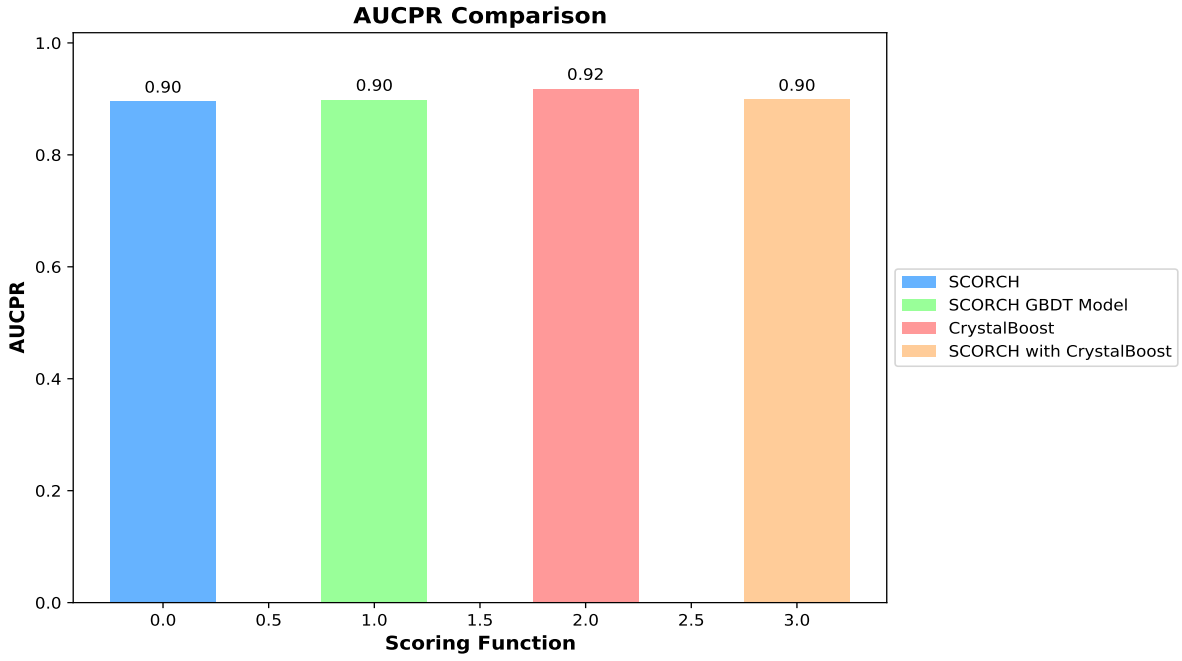


(b)

Figure 10: (a) PR curve for all four models tested on the CrystalBoost test set, with a random guesser included as a baseline. (b) ROC Curve for all four models on the CrystalBoost test data set, with a random guesser as a baseline.



(a)



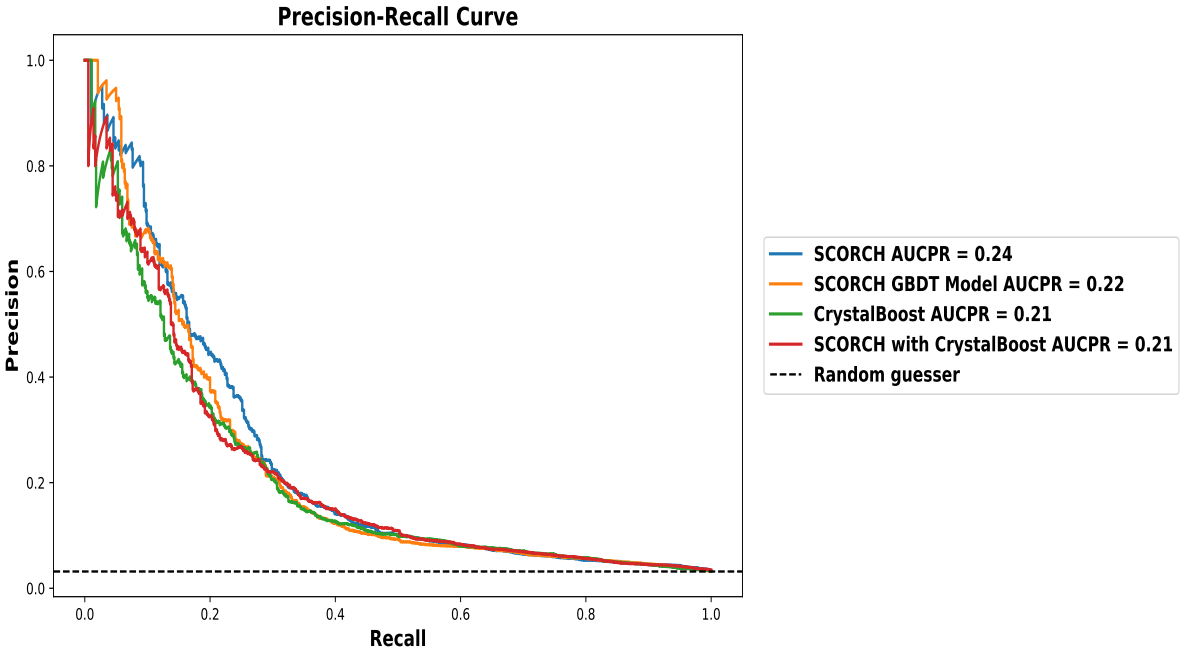
(b)

Figure 11: AUCPR values for each model tested on (a) The CrystalBoost test set and (b) The SCORCH test set. CrystalBoost is seen to outperform all other models. For comparison, a perfect AUCPR is 1.0

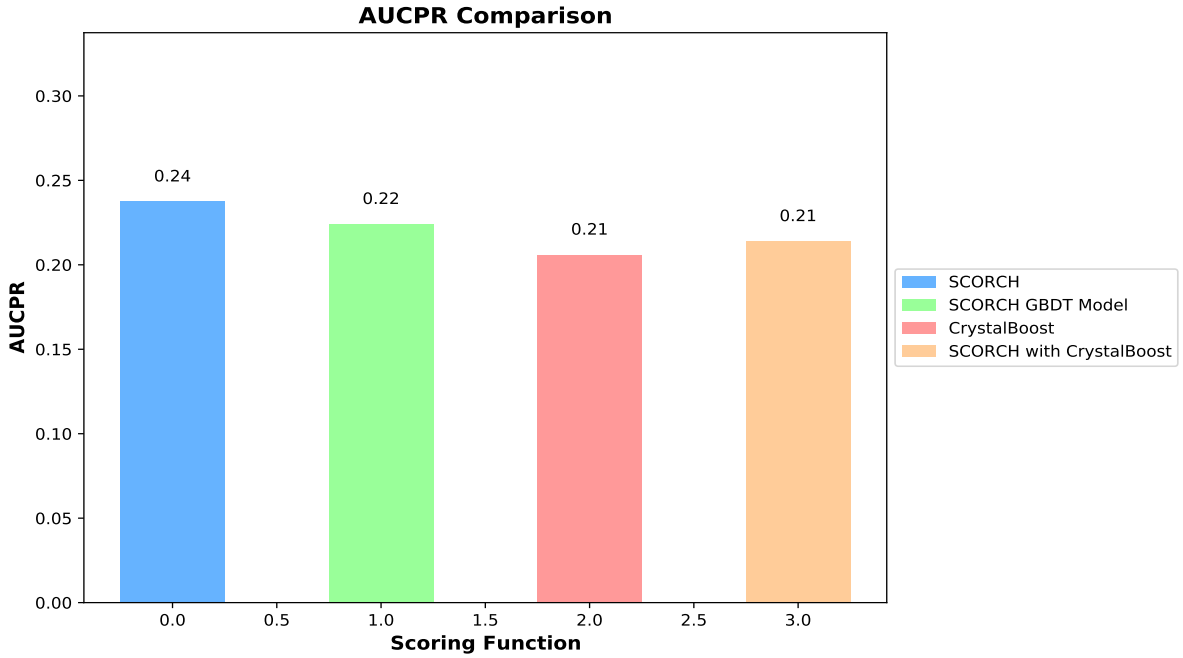
Screening Power on DEKOIS 2.0 Data Set

To ensure the results seen on the test data sets were not merely an overfitting artifact and to establish the model's generalisability, it was important to test its performance on an independent benchmark. Thus the performance of all four models was tested on the DEKOIS 2.0 dataset. As detailed in Section 2.10.1, this data set serves as a benchmark tailored for virtual screening workflows. Both the PR curve and the AUCPR were documented for each model using this benchmark. The outcomes of this evaluation are presented in Figure 12.

The results from the DEKOIS 2.0 benchmark dispute the initial hypothesis drawn from the results from the test data sets. In this benchmark, both the CrystalBoost model and the SCORCH model integrating CrystalBoost were outperformed by the original SCORCH model and its standalone SCORCH GBDT model. Of note, all four models displayed substantially lower AUCPR scores on the DEKOIS 2.0 data set than on the internal test data sets. This observation was also noted in the original SCORCH paper, where overfitting to the ”DeepCoy” decoys was proposed as a potential explanation [3]. A significant proportion of the poses included in the SCORCH and CrystalBoost data sets are DeepCoy generated decoy ligands. As described in Section 1.4, DeepCoy is a novel method of decoy generation, designed to mitigate decoy bias [17]. Considering the evidence of overfitting present in the learning curve in Figure 9, it is hypothesised that the CrystalBoost model may suffer from a similar issue, but to a larger degree. This explanation accounts for the higher scores on both internal test data sets containing DeepCoy decoys, but diminished performance on the external benchmark devoid of these ligands. This showcases a lack of generalisability, with high performance on familiar data - a hallmark of overfitting.



(a)



(b)

Figure 12: (a) PR Curve of all four models evaluated on the DEKOIS 2.0 independent benchmark data set, with a random guesser as a baseline. (b) AUCPR values for all four models on the DEKOIS 2.0 data set.

Enrichment Factor

Enrichment Factor (EF), as detailed in Section 2.10.1, is an invaluable metric in virtual screening, serving to measure the ability of a scoring function to prioritise active molecules over decoys in the early stages of a ranked list. It offers a practical perspective on a model's utility, emphasising the importance of identifying bioactive compounds with minimal computational resources expended.

In the evaluation performed on the DEKOIS 2.0 data set, the results showed that CrystalBoost consistently registered the lowest enrichment factor across all thresholds. The SCORCH model with CrystalBoost integrated had the second lowest performance, then the SCORCH GBDT, and finally the original SCORCH model showcased the highest enrichment. These results can be observed in the Figure 13. The trend of enrichment values levelling out as the EF threshold is raised can be attributed to the increasing likelihood of detecting true actives as the sample size grows. The EF results presented here serve to reinforce the lack of generalisability of CrystalBoost compared to SCORCH, and lend credence to the overfitting hypothesis.

Despite indications of overfitting, placing CrystalBoost's performance in the backdrop of other machine learning scoring functions offers broader perspective. When performance was compared on the same DEKOIS 2.0 dataset, both RF-ScoreVS v2 and NNScore 1 achieved lower AUCPR scores than any of the four models evaluated in this project. Specifically, these scoring functions recorded AUCPR values of 0.151 and 0.069, respectively. Furthermore, when assessing the enrichment factor at the same thresholds on the DEKOIS 2.0 data set, both RF-ScoreVS v2 and NNScore 1 consistently registered lower enrichment values compared to those obtained by CrystalBoost and SCORCH in this project [3]. This suggests that, even though CrystalBoost may not be an improvement to SCORCH on the DEKOIS 2.0 data set, it still outperforms several comparable scoring functions in the domain.

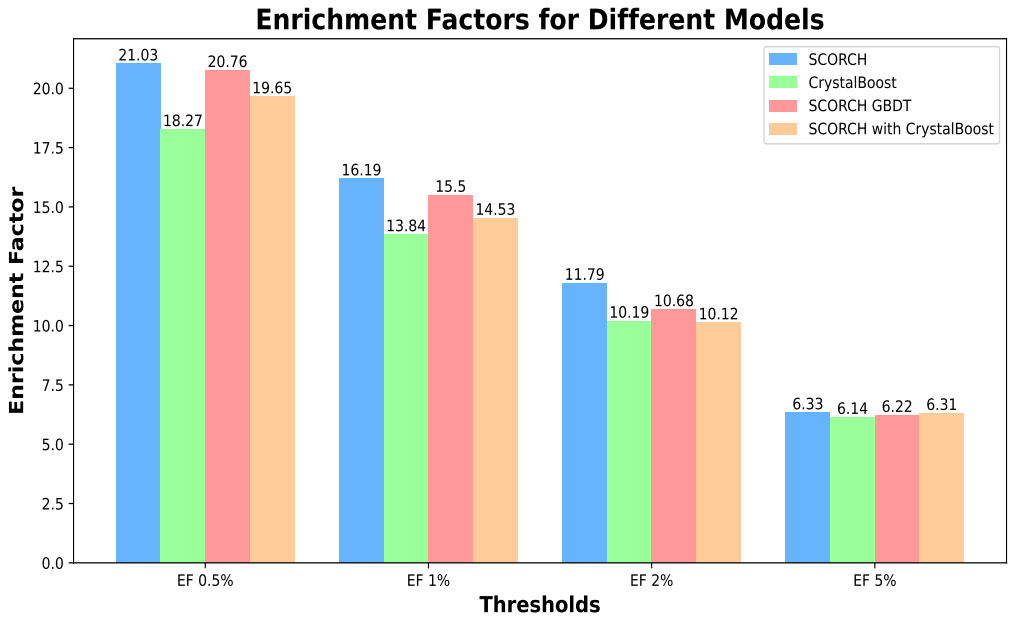


Figure 13: Enrichment factor (EF) of all four models at varying EF thresholds

3.5.2 Docking Power

Docking power evaluation was carried out using the CSAR 2014 native pose identification benchmark as described in Section 2.10.3.

From the results displayed in Figure 14, it's interesting to note that the SCORCH GBDT model outperformed all other models. This is in line with past literature which found GBDT models to be especially effective at structure-based virtual screening [8]. Following SCORCH GBDT, the original SCORCH model was the second highest performer, suggesting the strength of the GBDT model is improving its result.

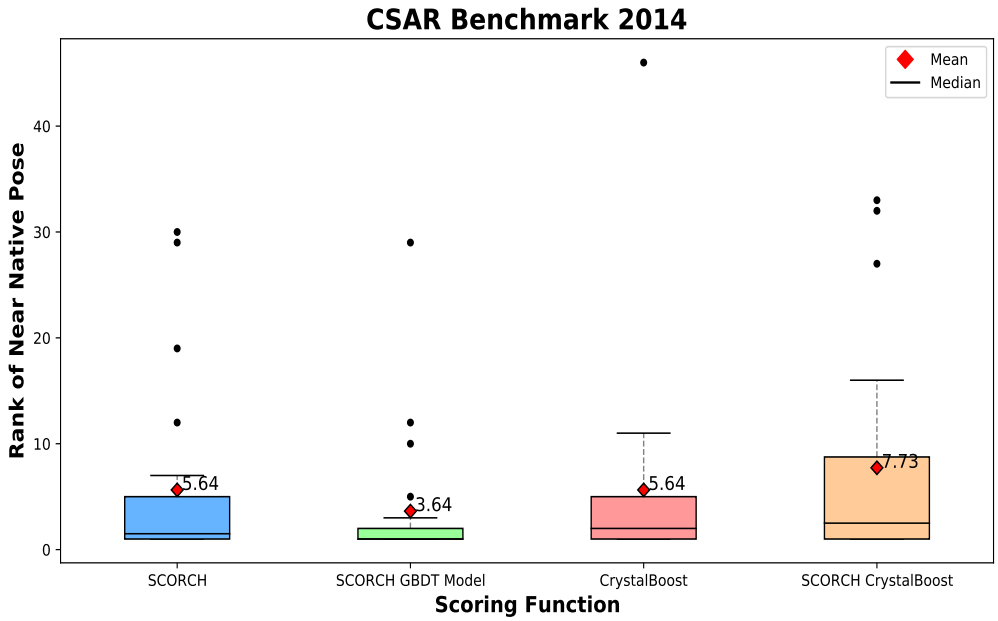


Figure 14: Ranks of near-native pose on the CSAR 2014 native pose benchmark as ranked by all four models. Red diamond indicates the mean rank of near-native pose, a lower value is a better result.

One might reasonably assume that the inclusion of crystal poses into a model would bolster its native pose identification capability as understanding the true binding pose can provide valuable insights into ligand-receptor interactions. However, both the SCORCH with CrystalBoost and the standalone CrystalBoost model noticeably underperformed on this benchmark.

One possible explanation for this could be in the nature of the crystal poses themselves. While they represent the actual binding configurations, the poses derived from X-ray crystallography might not always represent the most energetically favoured configurations. This is due to factors such as potential noise from electron density uncertainties and the influence of the crystal environment. This variability and uncertainty in poses could introduce conflicting signals during model training, which might account for the observed decrease in docking power [53].

Relating to the issue of data quality, it is important to consider the balance between the size and quality of the training data. Adding more data does not guarantee im-

proved performance, especially if that data reinforces certain patterns while neglecting others. This might explain why the original SCORCH, without the added crystal poses, managed to outperform its CrystalBoost, despite its larger quantity of data [54].

3.5.3 Evaluation of Hypothesis and Implications for Model Enhancement

In summary, the hypothesis that augmenting the training data of SCORCH with crystal poses to produce a new GBDT model will improve model performance seems unlikely to be true. CrystalBoost, even with a more extensive training data set inclusive of more true binders, didn't demonstrate the anticipated enhancement in benchmark performance. Furthermore, given that the combination of conformer generation with MCS alignment yielded mean RMSDs within the range of the resolution of the crystal poses, it suggests that our method likely represented the crystal poses with a high degree of accuracy. This further bolsters the argument against the initial hypothesis, as the lack of improvement can't be attributed to inaccuracies in representing the crystal poses supplied by the databases.

The fact that the model didn't show strong docking power is particularly surprising; a larger sample of genuine native poses would theoretically heighten its ability to discern such configurations. Furthermore, there was a slim margin of improvement observed between CrystalBoost and Model 11 in Table 1, in spite of Model 11 being trained on O3A aligned poses which were obviously flawed, and distant from a biologically realistic pose. This indicates that even the inclusion of accurate crystal poses as seen in CrystalBoost is unlikely to improve the model's performance. The superior AUCPR of Model 11 over SCORCH on the CrystalBoost test set further strengthens the argument for overfitting.

However, there are still some unturned stones. While the MCS-aligned poses have a mean RMSD value within the range of the crystal pose resolution, refining the meth-

ods for conformer generation and alignment might not necessarily just reduce RMSD, but could offer other performance or applicability advantages. By addressing potential overfitting and exploring the use of multiple crystal poses for each ligand, rather than a singular pose, advantages of integrating crystal poses could be discerned. These ideas will be discussed in more detail in the future work section.

4. Future Work

While the evidence presented in this project refutes the utility of incorporating crystal poses into the SCORCH training data, there still remains possible avenues of future work that may yet prove the addition of crystal poses to be useful.

4.0.1 Reducing Overfitting

A clear issue with model overfitting has been shown in this project, this overfitting issue has a direct impact on model generalisability, and as such may be negating the generalisability benefits theorised to be bestowed by the addition of crystal poses to the data. Multiple methods exist to combat overfitting which could be explored in future work.

One such example is the new *StratifiedGroupKFold* function in scikit-learn version 1.3.0 [34]. This functions similarly to the *GroupKFold* function utilised in this project. However, *StratifiedGroupKFold* goes a step further by aiming to produce stratified folds with non-overlapping groups. In the original SCORCH paper, the active protein-ligand complexes were randomly split into training, test, and validation sets using the *StratifiedShuffleSplit* function of scikit-learn 0.24.2. This splitting ensured identical distributions across all sets by stratifying them based on structure resolution and dissociation constant [3]. The advantage of using *StratifiedGroupKFold* in potential future work would be its ability to combine the benefits of both stratification and group-based splitting to improve balance in the cross-validation step, and thus reduce overfitting.

In this project, identical features were selected as were SCORCH in order to minimise variables. However, different data sets contain unique inter-feature relationships [55].

As such, carrying out feature selection specifically for this new data set may improve model performance and and reduce overfitting [56]. Furthermore, XGBoost has multiple methods of reducing overfitting in GBDT models built-in such as adjusting tree depths, implementing sub-sampling, and column sub-sampling, which can help prevent the model from excessively fitting to the training data [33]. While these parameters were adjusted during hyperparameter tuning discussed in Section 2.8, more experimentation with these methods may yield better results.

It is likely that both SCORCH and CrystalBoost are overfit to the DeepCoy decoys in the training data set. To mitigate the overfitting observed with the current decoys, enhancing the decoy diversity could improve model generalisability.

DeepCoy decoys were chosen for use in SCORCH to avoid the bias inherent in the popular DUD-E decoy data set [3, 14]. Another decoy data set, DUDE-Z was also designed to mitigate the bias in the DUD-E. These DUDE-Z decoys are explicitly charge-matched to the 3D protonation states of the active ligands at pH 7.4, ensuring the molecules are more physiologically relevant. Additionally, structural dissimilarity filters are present in DUDE-Z to provide more diverse decoy structures, which could counteract overfitting to specific decoy characteristics [57]. By augmenting the CrystalBoost data set with DUDE-Z decoys alongside DeepCoy-generated decoys, the resultant decoy diversity might effectively reduce overfitting.

4.0.2 Alternative Alignment and Conformer Generation Methods

While the conformer generation technique used in this paper proved to be precise when measured by mean RMSD, alternative methods exist that might offer conformations of higher biological relevance. A prominent example is the commercial software, OMEGA. Recognised as a top-performing commercial conformer generation tool, OMEGA stands out for its proficiency in generating bioactive conformers [58–

60]. OMEGA works by using a modified version of the MMFF94s akin to the one used in our project. Importantly, benchmarks have highlighted its significant improvement in accuracy over RDKit, the foundational tool behind the *rdk_confgen.py* script used in this project. A non-commercial alternative to OMEGA can be found in the conformer generator tool Conformator, which is notable not only for its versatility in handling different ligand input formats but also for its performance, which is comparable to that of OMEGA [61].

An improved conformer technique could be coupled with a better alignment tool, such as BCL::MolAlign. This tool is noted for its highly accurate alignment results. Of particular relevance, it has been found to be more accurate than the maximum common substructure (MCS) approach, which was used in this project. An attempt was made the use BCL::MolAlign in this project, but difficulties in obtaining an academic license prevented its implementation within the available timeframe.

4.0.3 Inclusion of an Array of Crystal Poses

As discussed in Section 2.3, molecules do not typically exist defined structure, but they instead vary across a range of conformations over time. [41]. Furthermore, considering the electron density and resolution issues discussed in Section 3.5.2 and Section 3.2, it may be that a single crystal pose does not accurately represent the inherent flexibility and variability of a ligand's binding pose. By incorporating an ensemble of crystal poses for each ligand that fit within the acceptable electron density boundaries, a better approximation of the true binding behavior of ligands may be obtained, thus improving model performance [62]. Perhaps this multi-pose strategy, combined with improved conformer generation and alignment would yield an improvement to the performance of SCORCH and CrystalBoost, revealing the benefit of the inclusion of crystal poses.

5. Conclusion

This project investigated the potential of incorporating X-ray crystallography-derived poses into the training data of machine learning of the machine learning scoring function SCORCH. The hypothesis was that by supplementing the GWOVina generated poses with experimental crystal poses, it would enhance the model's generalisability and docking power by providing more diverse and accurate examples of true binding configurations.

To test this hypothesis, a gradient boosted decision tree model called CrystalBoost was developed by combining crystal poses for 5084 protein-ligand complexes with the original SCORCH training data. This model was compared to the original SCORCH model, the standalone SCORCH GBDT model, as well as the SCORCH model incorporating CrystalBoost in place of its original GBDT model. While CrystalBoost demonstrated superior performance on internal test sets, it under-performed relative to SCORCH on the external DEKOIS 2.0 benchmark. This discrepancy indicates possible overfitting to the training data. Furthermore, neither CrystalBoost nor the SCORCH model including CrystalBoost showed improved docking power.

In conclusion, the findings of this project refute the original hypothesis. Incorporating crystal poses into the training data did not enhance model generalisability or docking power compared to the original SCORCH model. It is worth noting however that the overfitting issue observed may have overshadowed any generalisability benefit bestowed by the inclusion of the crystal poses. Thus, avenues remain unexplored that could potentially reveal advantages of including crystallographic data. This includes using a more diverse set of decoys, including multiple crystal poses per ligand, and experimenting further with methods to reduce overfitting. Further research into ma-

chine learning scoring function design and training data curation may yet demonstrate the utility of leveraging crystallographic poses alongside docking-programme derived poses. While the initial hypothesis was not validated, this project contributed to the broader understanding of the development machine learning scoring functions.

Bibliography

1. Dhakal, A., McKay, C., Tanner, J. J. & Cheng, J. Artificial intelligence in the prediction of protein–ligand interactions: recent advances and future directions. *Briefings in Bioinformatics* **23,** bbab476 (2022).
2. Yasuo, N. & Sekijima, M. Improved method of structure-based virtual screening via interaction-energy-based learning. *Journal of Chemical Information and Modeling* **59,** 1050–1061 (2019).
3. McGibbon, M., Money-Kyrle, S., Blay, V. & Houston, D. R. SCORCH: improving structure-based virtual screening with machine learning classifiers, data augmentation, and uncertainty estimation. *Journal of Advanced Research* **46,** 135–147 (2023).
4. Yang, C., Chen, E. A. & Zhang, Y. Protein–ligand docking in the machine-learning era. *Molecules* **27,** 4568 (2022).
5. Liu, J. & Wang, R. Classification of current scoring functions. *Journal of chemical information and modeling* **55,** 475–482 (2015).
6. Rayka, M. & Firouzi, R. GB-score: Minimally designed machine learning scoring function based on distance-weighted interatomic contact features. *Molecular Informatics* **42,** 2200135 (2023).
7. Li, H., Sze, K.-H., Lu, G. & Ballester, P. J. Machine-learning scoring functions for structure-based drug lead optimization. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **10,** e1465 (2020).
8. Shen, C. *et al.* Can machine learning consistently improve the scoring power of classical scoring functions? Insights into the role of machine learning in scoring functions. *Briefings in Bioinformatics* **22,** 497–514 (2021).

9. Gomes, J., Ramsundar, B., Feinberg, E. N. & Pande, V. S. Atomic convolutional networks for predicting protein-ligand binding affinity. *arXiv preprint arXiv:1703.10603* (2017).

10. Wallach, I., Dzamba, M. & Heifets, A. AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery. *arXiv preprint arXiv:1510.02855* (2015).

11. Wójcikowski, M., Ballester, P. J. & Siedlecki, P. Performance of machine-learning scoring functions in structure-based virtual screening. *Scientific Reports* **7,** 46710 (2017).

12. Durrant, J. D. & McCammon, J. A. NNScore: a neural-network-based scoring function for the characterization of protein- ligand complexes. *Journal of chemical information and modeling* **50,** 1865–1871 (2010).

13. Hunter, J. D. Matplotlib: A 2D graphics environment. *Computing In Science & Engineering* **9,** 90–95 (2007).

14. Chen, L. *et al.* Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening. *PloS one* **14,** e0220113 (2019).

15. Ha, V. T. Experimental Study on Remaining Useful Life Prediction of Lithium-Ion Batteries Based on Three Regressions Models for Electric Vehicle Applications (2023).

16. Houston, D. R. & Walkinshaw, M. D. Consensus docking: improving the reliability of docking in a virtual screening context. *Journal of chemical information and modeling* **53,** 384–390 (2013).

17. Imrie, F., Bradley, A. R. & Deane, C. M. Generating property-matched decoy molecules using deep learning. *Bioinformatics* **37,** 2134–2141 (2021).

18. Pal, A. *Gradient boosting trees for classification: A beginner's guide* Oct. 2020. `https://medium.com/swlh/gradient-boosting-trees-for-classification-a-beginners-guide-596b594a14ea`.

19. Renaud, J.-P. *et al.* Biophysics in drug discovery: impact, challenges and opportunities. *Nature reviews Drug discovery* **15,** 679–698 (2016).

20. Wong, K. M., Tai, H. K. & Siu, S. W. GWOVina: A grey wolf optimization approach to rigid and flexible receptor docking. *Chemical Biology & Drug Design* **97,** 97–110 (2021).

21. *Anaconda Software Distribution* version Vers. 2023.03-1. 2023. `https://anaconda.com/`.

22. Wang, R., Fang, X., Lu, Y. & Wang, S. The PDBbind database: Collection of binding affinities for protein- ligand complexes with known three-dimensional structures. *Journal of medicinal chemistry* **47,** 2977–2980 (2004).

23. Hu, L., Benson, M. L., Smith, R. D., Lerner, M. G. & Carlson, H. A. Binding MOAD (mother of all databases). *Proteins: Structure, Function, and Bioinformatics* **60,** 333–340 (2005).

24. Warren, G. L., Do, T. D., Kelley, B. P., Nicholls, A. & Warren, S. D. Essential considerations for using protein–ligand structures in drug discovery. *Drug Discovery Today* **17,** 1270–1281 (2012).

25. McGibbon, M. *XGBScore* `https://github.com/miles-mcgibbon/XGBScore`. 2021.

26. Landrum, G. *et al. rdkit/rdkit: 2023_03_2 (Q1 2023) Release* version Release_2023_03_2. June 2023. `https://doi.org/10.5281/zenodo.8053810`.

27. iwatobipen. *rdk_confgen* `https://github.com/iwatobipen/rdk_confgen`. 2021.

28. Ebejer, J.-P., Morris, G. M. & Deane, C. M. Freely available conformer generation methods: how good are they? *Journal of chemical information and modeling* **52,** 1146–1158 (2012).

29. Morris, G. M. *et al.* AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *Journal of computational chemistry* **30,** 2785–2791 (2009).

30. Durrant, J. D. & McCammon, J. A. BINANA: a novel algorithm for ligand-binding characterization. *Journal of Molecular Graphics and Modelling* **29,** 888–893 (2011).

31. Sánchez-Cruz, N., Medina-Franco, J. L., Mestres, J. & Barril, X. Extended connectivity interaction features: improving binding affinity prediction through chemical description. *Bioinformatics* **37,** 1376–1382 (2021).

32. An index of flexibility from molecular shape descriptors. *Progress in clinical and biological research* **291,** 105–109. ISSN: 0361-7742 (1989).

33. Chen, T. & Guestrin, C. *Xgboost: A scalable tree boosting system* in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (2016), 785–794.

34. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **12,** 2825–2830 (2011).

35. Head, T., Kumar, M., Nahrstaedt, H., Louppe, G. & Shcherbatyi, I. *scikit-optimize/scikit-optimize* version v0.9.0. Oct. 2021. `https://doi.org/10.5281/zenodo.5565057`.

36. Saito, T. & Rehmsmeier, M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one* **10,** e0118432 (2015).

37. Bender, A. & Glen, R. C. A discussion of measures of enrichment in virtual screening: comparing the information content of descriptors with increasing levels of sophistication. *Journal of chemical information and modeling* **45,** 1369–1375 (2005).

38. Bauer, M. R., Ibrahim, T. M., Vogel, S. M. & Boeckler, F. M. Evaluation and optimization of virtual screening workflows with DEKOIS 2.0–a public library of challenging docking benchmark sets. *Journal of chemical information and modeling* **53,** 1447–1462 (2013).

39. Ibrahim, T. M., Bauer, M. R. & Boeckler, F. M. Applying DEKOIS 2.0 in structure-based virtual screening to probe the impact of preparation procedures and score normalization. *Journal of cheminformatics* **7,** 1–16 (2015).

40. Carlson, H. A. *et al.* CSAR 2014: a benchmark exercise using unpublished data from pharma. *Journal of chemical information and modeling* **56,** 1063–1077 (2016).

41. Agrafiotis, D. K., Gibbs, A. C., Zhu, F., Izrailev, S. & Martin, E. Conformational sampling of bioactive molecules: a comparative study. *Journal of chemical information and modeling* **47,** 1067–1086 (2007).

42. Schrödinger, LLC. *The PyMOL Molecular Graphics System, Version 1.8* 2015.

43. Addicoat, M. A., Vankova, N., Akter, I. F. & Heine, T. Extension of the universal force field to metal–organic frameworks. *Journal of chemical theory and computation* **10,** 880–891 (2014).

44. Halgren, T. A. MMFF VI. MMFF94s option for energy minimization studies. *Journal of computational chemistry* **20,** 720–729 (1999).

45. Halgren, T. A. MMFF VII. Characterization of MMFF94, MMFF94s, and other widely available force fields for conformational energies and for intermolecular-interaction energies and geometries. *Journal of Computational Chemistry* **20,** 730–748 (1999).

46. Tosco, P., Stiefl, N. & Landrum, G. Bringing the MMFF force field to the RDKit: implementation and validation. *Journal of cheminformatics* **6,** 1–4 (2014).

47. Hönig, S. M., Lemmen, C. & Rarey, M. Small molecule superposition: A comprehensive overview on pose scoring of the latest methods. *Wiley Interdisciplinary Reviews: Computational Molecular Science* **13,** e1640 (2023).

48. Landrum, G. *et al. rdkit.Chem.rdMolAlign module* `https://www.rdkit.org/docs/source/rdkit.Chem.rdMolAlign.html`.

49. Schrödinger, LLC. *The PyMOL Molecular Graphics System, Align Module* `https://pymolwiki.org/index.php/Align`.

50. Landrum, G. *et al. rdkit.Chem.MCS module* `https://www.rdkit.org/docs/source/rdkit.Chem.MCS.html`.

51. Dutkiewicz, Z. Computational methods for calculation of protein-ligand binding affinities in structure-based drug design. *Physical Sciences Reviews* **7,** 933–968 (2020).

52. Ying, X. *An overview of overfitting and its solutions* in *Journal of physics: Conference series* **1168** (2019), 022022.

53. Davis, A. M., Teague, S. J. & Kleywegt, G. J. Application and limitations of X-ray crystallographic data in structure-based ligand and drug design. *Angewandte Chemie International Edition* **42,** 2718–2736 (2003).

54. Srihith, D. & Sai, I. V. Training Data Alchemy: Balancing Quality and Quantity in Machine Learning Training. *Journal of Network Security and Data Mining* **6.**

55. Cai, J., Luo, J., Wang, S. & Yang, S. Feature selection in machine learning: A new perspective. *Neurocomputing* **300,** 70–79 (2018).

56. Guyon, I. & Elisseeff, A. An introduction to variable and feature selection. *Journal of machine learning research* **3,** 1157–1182 (2003).

57. Stein, R. M. *et al.* Property-unmatched decoys in docking benchmarks. *Journal of chemical information and modeling* **61,** 699–714 (2021).

58. Hawkins, P. C., Skillman, A. G., Warren, G. L., Ellingson, B. A. & Stahl, M. T. Conformer generation with OMEGA: algorithm and validation using high quality structures from the Protein Databank and Cambridge Structural Database. *Journal of chemical information and modeling* **50,** 572–584 (2010).

59. Friedrich, N.-O. *et al.* Benchmarking commercial conformer ensemble generators. *Journal of chemical information and modeling* **57,** 2719–2728 (2017).

60. Friedrich, N.-O. *et al.* High-quality dataset of protein-bound ligand conformations and its application to benchmarking conformer ensemble generators. *Journal of chemical information and modeling* **57,** 529–539 (2017).

61. Friedrich, N.-O. *et al.* Conformator: a novel method for the generation of conformer ensembles. *Journal of Chemical Information and Modeling* **59,** 731–742 (2019).

62. Mobley, D. L. & Dill, K. A. Binding of small-molecule ligands to proteins:“what you see” is not always “what you get”. *Structure* **17,** 489–498 (2009).

6. Acknowledgements

I would like to thank my supervisor Dr. Douglas Houston for giving me the opportunity to work on this project, and for his continued guidance over the course of this project. I would also like to thank Miles McGibbon for his invaluable advice and help throughout the project.

A. Appendix

A.1 Dissertation GitHub Repository

https://github.com/B221734-2022/Dissertation

A.2 5-Fold Cross Validation Diagram

Figure A1: Visualisation of 5-Fold cross-validation produced with a custom Python script using Matplotlib [13].

A.3 Additional Model Evaluation Plots

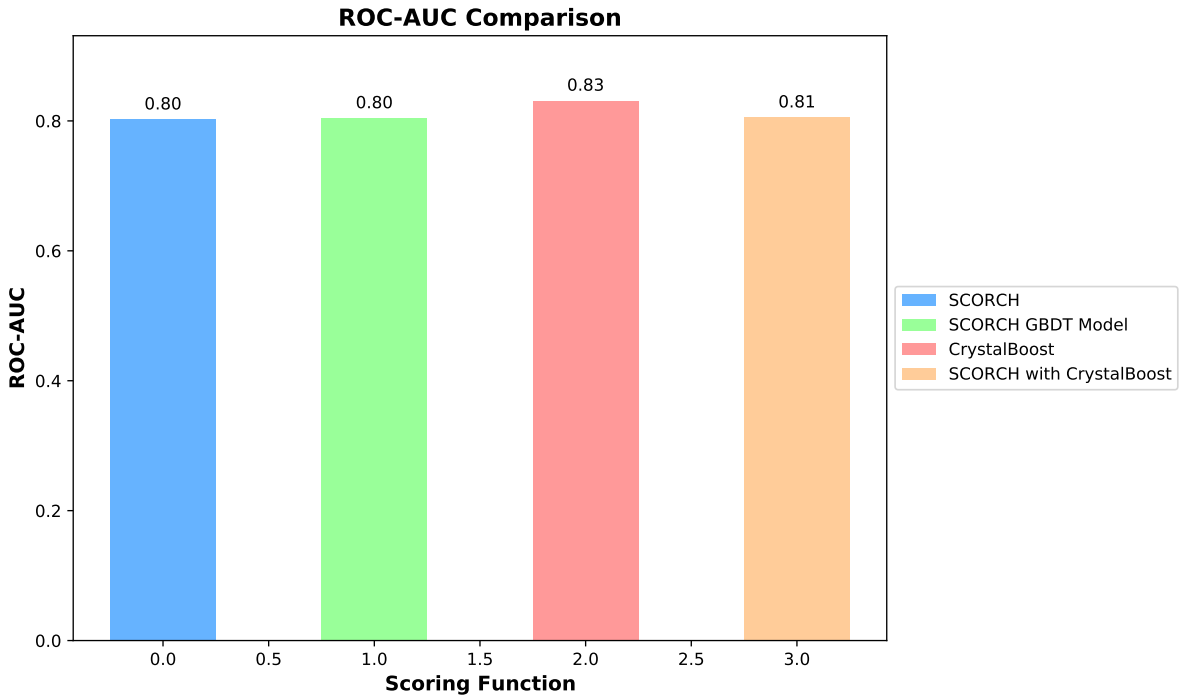


Figure A1: ROC-AUC values for all four models on the CrystalBoost test data set

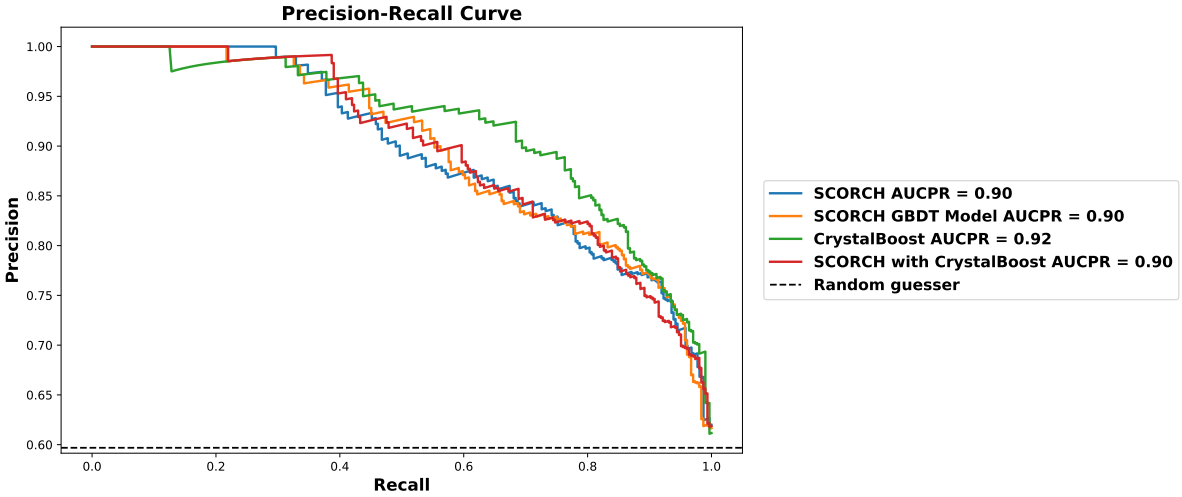


Figure A2: PR curve for all four models on the SCORCH test data set

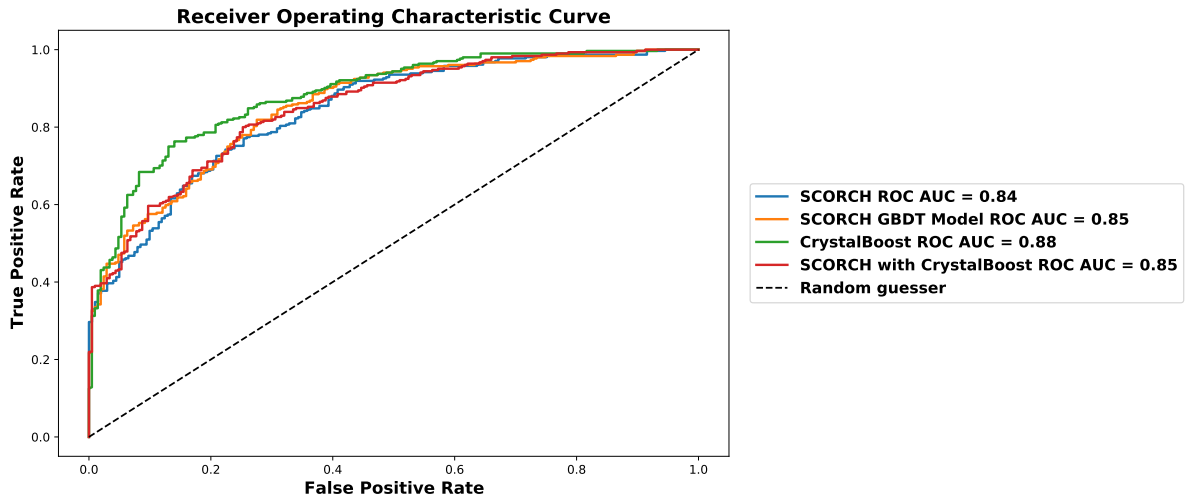


Figure A3: ROC curve for all four models on the SCORCH test data set

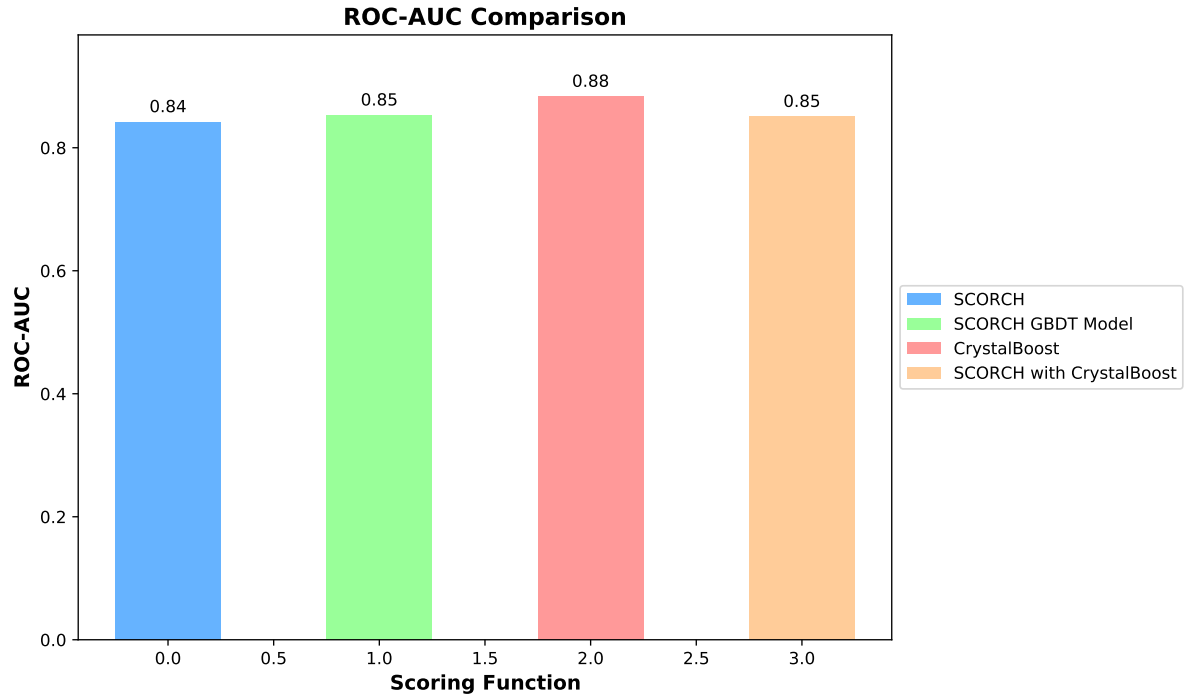


Figure A4: ROC-AUC values for all four models on the SCORCH test data set

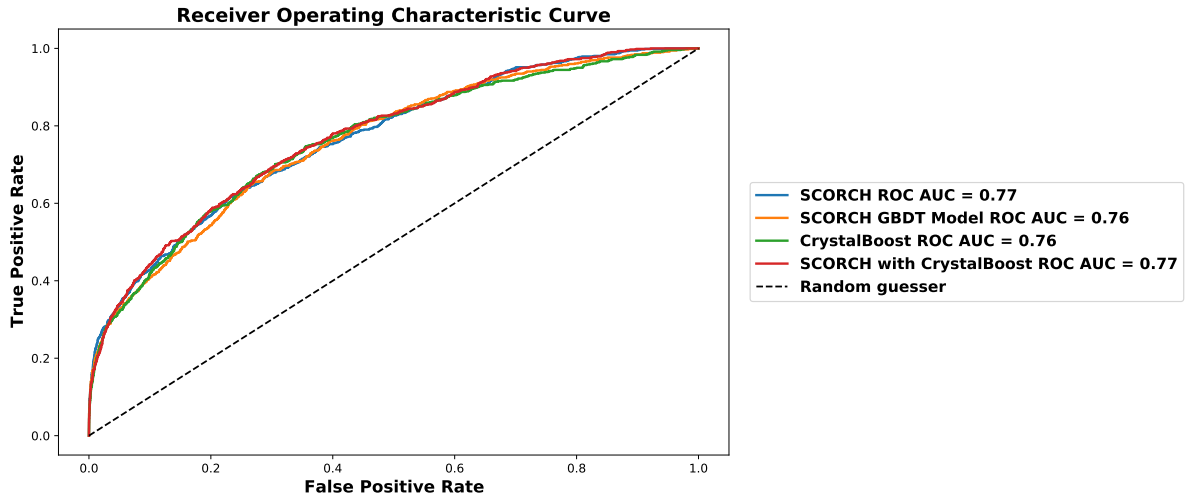


Figure A5: ROC curve for all four models on the DEKOIS 2.0 independent benchmark data set

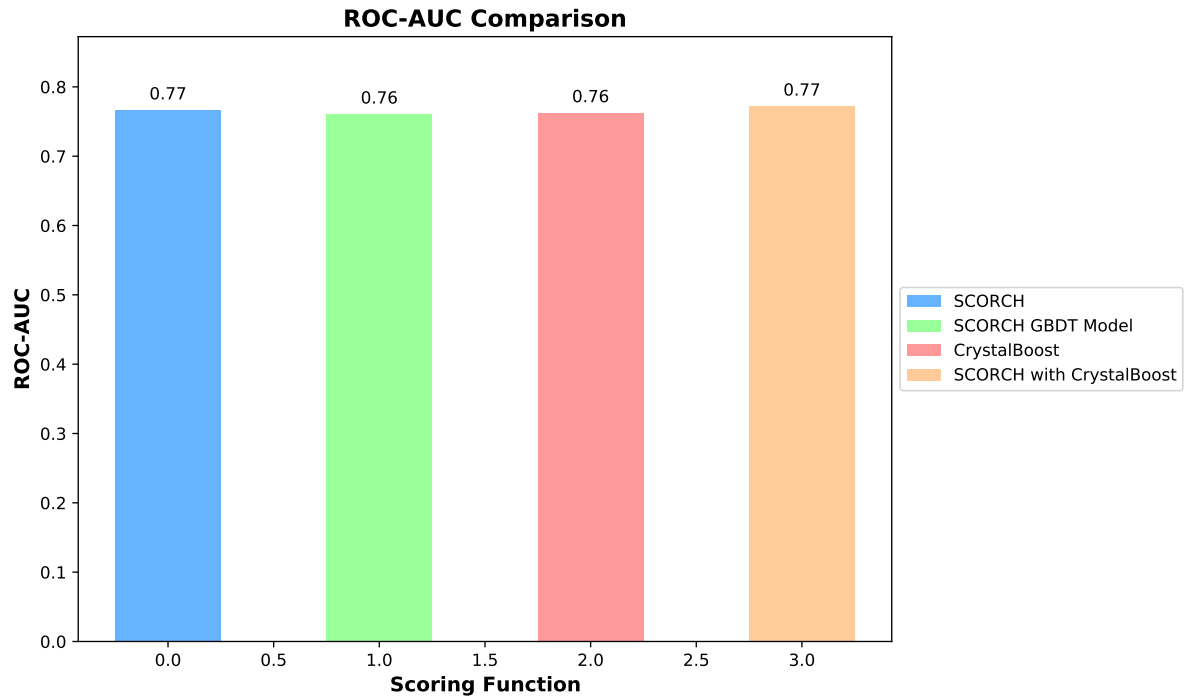


Figure A6: ROC-AUC values for all four models on the DEKOIS 2.0 independent benchmark data set