Machine learning methods for docking score prediction

by

Margarita Kovaleva

A dissertation submitted in fulfillment of the requirements for the degree of Bachelor of Applied Mathematics and Physics

program Molecular Biology and Biophysics

Supervisor: Valentin Borshchevskiy

Moscow Institute of Physics and Technology
(National Research University)

Department of General and Applied Physics
Biophysics Chair

2021

Contents

Chapter 1

Introduction

1.1 Structure-based drug discovery principles

Small molecule drug design and development is an inventive process of finding new medicines. This complex task can be divided into smaller parts[1]. The first major stage, drug discovery, consists of all the experimental and computational studies designed to move a program from the initial identification of a biological target to the identification of a compound with the potential to be clinically relevant. This stage can be broken down into several phases:

1. **Target discovery**
   One of the key stages is to choose the right target which can be manipulated to influence certain biochemical processes while not affecting the others. If the function of the target is defined only hypothetically, the validation is required to understand the link between it and phenotypic traits of the disease of interest.

2. **Lead discovery**
   Once a target has been identified and validated, the biological screening of large libraries of compounds is conducted to discover multiple drug candidates. The lead discovery stage will be discussed in more detail further in Section 1.2.

3. **Lead optimization**
   After the initial identification of leads, they can modified chemically to achieve improvements of affinity, selectivity, mode of action, synthesizability and ADMET (absorption, distribution, metabolism, excretion and toxicity)-properties. Since there are many ways to optimize the compound, this step is cyclic and time-consuming. Finally, the affinity of „original hits“, which is very low, increases by several orders of magnitude.

After lead optimization, several high-potency compounds should be selected as a clinical candidate.

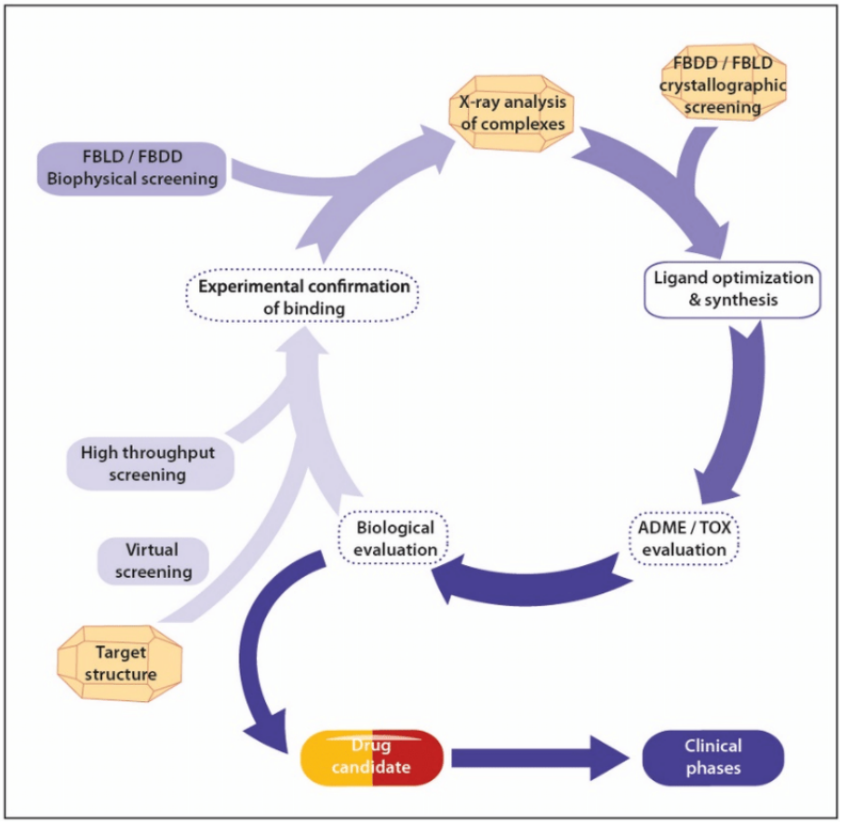


Figure 1.1: Drug discovery cycle. Taken from [2]

At the drug discovery stage the potency of the candidate is in the spotlight. However, high potency is not the only criteria to be examined. The second major stage, drug development, is a process of bringing the potential medicine to market. The candidate must be proven efficient and safe.

1.2 Lead discovery stage

To identify compounds which can be utilized in clinical setting, two general methods are applied, physical high throughput screening (HTS) and computer-aided drug-design. Since HTS is expensive and takes more time, money and resources than *in silico* screening, it is common approach to select small subset of molecules from large libraries by means of computational methods - by virtual screening, and after that test them in physical HTS. The use of computers in drug discovery makes the process of candidate search more quick and cost-efficient.

With respect to virtual screening, there are two different techniques: ligand-based and structure-based. The first one uses similarity model or quantitative structure-activity relationship (QSAR) to search for possible ligands and does not need structure of a target, but

a known ligand as a template. Structure-based techniques, in contrast, try to simulate the physics of protein-ligand binding and calculate a quantitative score intended to correlate with the free energy of binding, requiring a target structure but not target-specific bioactivity data. Structure-based methods include molecular dynamics and computational docking[3]. Further, QSAR and molecular docking will be considered in detail in this work.

1.3 Small molecules and databases

One of the significant questions is where to search candidates for lead discovery stage, as long as chemical space is almost infinite; indeed, it was estimated[4] that the number of molecules containing 30 heavy atoms (or non-hydrogens) outweighs 10^{60}. Usually, potential leads are searched among so called "small molecules". Currently, this term typically implies an organic compound whose molecular weight is less than roughly 1000 Da[5]. Such compounds may be chemically stable and spread through the body to reach a targeted protein after being injected or ingested. However, molecular weight is not the only criteria for drug-like molecules. For example, Lipinski's rule of 5[1], suggests that drug-like compounds will have:

- a molecular weight lower than 500;
- a logP (octanol-water partition coefficient, or lipophilicity) below 5;
- less than 5 hydrogen bond donors;
- less tan 10 hydrogen bond acceptors;
- less than 10 rotatable bonds.

Many related rules have been subsequently modified and proposed as the "Rule-of-Three", which defines fragment properties with:

- an average molecular weight $\leq$ 300 Da;
- a calculated partition coefficient (Clog P) $\leq$ 3;
- the number of hydrogen bond donors $\leq$ 3;
- the number of hydrogen bond acceptors $\leq$ 3;
- the number of rotatable bonds $\leq$ 3.

Also, Pfizer's "Rule of 3/75" has been described which states that compounds with

- a ClogP of $\leq$ 3;

- topological polar surface area (TPSA) ≥ 75

have the best chances of being well tolerated from a safety perspective in vivo.

Compounds are gathered in libraries, many of which are freely available online. The number of purchasable items in these libraries which can be examined in drug design has grown significantly in recent years. For example, ZINC20[6] is a free database of commercially available compounds that in its current version contains nearly two billions of small molecules, while in 2015[7] the library comprised roughly 120 million drug-like molecules.

There is a common way to store information about molecules compound libraries. For example, SMILES notation is often used.

1.4 QSAR models

One of the major computational tools in drug design is quantitative structure-activity relationship, QSAR - a method of mapping chemical structure to molecular properties[8]. Some features, like molecular weight, can be calculated directly for any structure, synthesised or just virtual, while other characteristics, e.g. logP, should be measured only experimentally. QSAR models attempt to predict these unknown properties based on the information from the molecules which have been already tested experimentally. Thus, accurate QSAR model can noticeably simplify the compound selection process, providing the information about their biological activity.

The history of QSAR modelling started in 1964 with publication of the Hansch's et al. paper[9] dedicated to correlation between biological activity and chemical structure. This work was important because of several ideas:

- parameters describing electric, steric and hydrophobic molecular properties had been combined in one equation;
- parabolic model for lipophilicity-activity relationship had been proposed based on the reasoning that drug should, on the one hand, circulate in the bloodstream (i.e. be soluble in water), on the other hand, penetrate cell membranes (i.e. dissolve in lipids);
- it had been suggested that logP of a molecule is an additive parameter: the partial contributions of a substituent to the log P of any molecule are almost the same.

The introduction of simple QSAR model had played a great role in understanding how molecular structure influences its biological affinity. The application of the method gave rise

to vast amount of publications in wide range of areas[10]: from plant growth regulation and metabolism to cytotoxicity and carciogenicity issues.

Nowadays QSAR modeling is widely practiced in academy, industry, and government institutions around the world. QSAR models find broad application for assessing potential impacts of chemicals, materials, and nanomaterials on human health and ecological systems.

Considering drug-receptor interaction models, traditional QSAR is receptor-independent, what means that only ligand data is handled. The main assumption in QSAR is that every property of a molecule is a function of the data derived from its structure.

1.5 Molecular descriptors

In order to predict chemical properties, a relationship between the property of a molecule and the parameters which characterize its structure should be established. To build this property, the molecules' structures should be converted to a convenient, numerical form - molecular descriptors. In the Handbook of Molecular Descriptors[11], molecular descriptor is defined as the final result of a logic and mathematical procedure which transforms chemical information encoded within a symbolic representation of a molecule into a useful number.

Descriptors should meet several criteria[12]:

- invariant to atom and bonds renumbering;
- invariant to rotation and shift;
- calculated unambiguously;
- easy to operate with: values mustn't be too large or below the machine precision limit;
- not a complex number.

Also, it is preferable that descriptor:

- can be interpreted physically or structurally;
- is not correlated trivially with other descriptors;

- does not change significantly due to slightly structure alteration;
- is not limited to a small subset of compounds.

Many different types of numerical descriptors are described in the literature. Most popular molecular descriptors are 2D-descriptors, which derive from two-dimensional, also called topological, representation of a molecule, which defines the connectivity of atoms in the molecule in terms of the presence and nature of chemical bonds[13].

1.5.1 Morgan/circular fingerprints

Also referred as extended-connectivity fingerprints, or ECFP[14]. One of the most common representations in the class of 2D fingerprints, which are computationally inexpensive and easy to interpret and visualize. The ECFP generation process has three sequential stages[14]:

1. An initial assignment stage in which each non-hydrogen atom has an integer identifier assigned to it (All rules which are independent of atom numbering can be applied). These identifiers are gathered into a fingerprint set.
2. An iterative updating stage in which each atom collects its own identifiers among with identifiers of its neighbours (excluding hydrogen atoms) into an array, sorted according to the current identifiers and the order of the bond order (single, double, triple, and aromatic), and a hash function is used to convert this array into a new integer identifier. The iteration is repeated a predefined number of times. As the process is repeated, the atom identifier represents a substructure of increasing size.
3. A duplicate identifier removal stage. Structural duplication is a situation when two different atoms contain information about identical structural regions of a molecule. The hashed identifier generated for these two atoms will be different, even though they represent the same underlying structure. In order to avoid adding useless redundancy to the fingerprint. To identify such duplicates, each feature keeps track of the substructure it represents in a molecule. Before the newly generated features from an iteration are appended to the fingerprint set, they are checked to see if any structural duplicates exist. The removal of duplicates has the additional effect that, at some number of iterations, fewer features will be generated than at the previous iteration level, and at some larger number of iterations, no more new features will be generated.

Finally, the list of identifiers is folded into a vector (2048-bit by default).

The ECFP rule for generating initial identifiers is derived from the properties used in the Daylight atomic invariants rule[15]:

- number of nearest-neighbour non-hydrogen atoms;
- number of bonds attached to the atom (not including bonds to hydrogens);
- atomic number;
- atomic mass;
- atomic charge;
- number of hydrogens connected to the atom;
- is the atom in a ring (1) or not (0) - additional property, not included in Daylight atomic invariants rule.

To create an integer identifier from this information, these values are hashed into a single 32-bit integer value.

Depending on the number of iterations, different names age are given to fingerprints. For example, in the RDKit library default number of iterations in the Morgan fingerprints is 2, which corresponds to ECFP4, where 4 represents the *diameter* of the atom environments considered.

Different hash functions can be chosen, but should meet special requirements: to map arrays of integers randomly and uniformly into the 2^{32}-size space of all possible integers; without uniform coverage, the collision rate may increase, leading to a loss of information.

1.5.2 Atom pairs fingerprints

According to its name, the atom-pair fingerprint is constructed using pairs of atoms (as features) and their topological distances. Features used in atom's descriptions contain:

- atom's chemical type;
- the number of non-hydrogen atoms attached to the it;
- the number of bonding π-electrons that it bears

The process of generating atom pair fingerprints consist of several steps:

1. Heavy atom are identified and the shortest distance between each pair of them is calculated;
2. Features of atoms are encoded;

3. Encoded features are converted into bit strings and represented as an integer;

4. Strings are concatenated and passed to a hash function.

1.6 Molecular docking

Structure-based method, named docking, requires a crystal structure of a target, obtained from NMR, cryo-electron microscopy or X-ray data. In principle, docking can screen virtual libraries of great size and diversity, selecting only the best-fitting molecules for synthesis and testing. However, it has serious disadvantage: the technique has a high percentage of false-positive hits. Besides, docking cannot calculate the affinity of compounds accurately[16], still being an effective ranking tool.

There are two classes of method needed for docking implementation[17]: the search algorithm, which is responsible for searching through different orientations, or poses, of a ligand in a binding pocket; and the scoring function, which estimates the binding affinities of poses.

Search algorithms must simultaneously be fast and cover chemical space effectively. If search space consists of all possible conformations, it is impossible to explore all of it with present computational power; on the contrary, ignoring some degrees of freedom can lead to inaccurate docking results. Nowadays, search algorithms can be classified into the following groups of algorithms:

- Rigid-body: both ligand and target are treated as rigid body, and only rotational and transitional freedom is considered;
- Flexible-ligand: protein flexibility still doesn't considered, but ligand flexibility does;
- Flexible ligand-flexible-protein: both ligand and target are considered as flexible. One of the ways to take into consideration the target flexibility is to dock ligand into multiple fixed receptor conformations[18];

Scoring functions (SF) must predict binding free-energy accurate enough, so different types of physical interactions and entropic effects should be taken into account. However, current computational resources constrain the algorithm's complexity. Thus, SF make approximations in order to compromise between speed and accuracy. One can divide a wide range of scoring functions into four classes[19],[20]:

- Empirical methods: the SF sums up the contributions of several terms, each related to energetic factor in protein-ligand binding. This terms might be "rewarding", such as hydrogen binding, or "penalties", e.g. frozen rotatable bonds. To define coefficients

before each term, a regression analysis is conducted with a train set of experimentally determined structures with known binding affinity data.

- Physics-based (or force-field based) methods: as well as in empirical methods, binding energy is decomposed into energy terms, but the difference is that physical-based approach fully relies on theory (force field, quantum mechanics and solvent models);

- Knowledge-based methods: the approach uses the so-called inverse Boltzmann statistic principle. According to Boltzmann statistics, the probability of occurrence of a given state with energy E is proportional to $\exp(-E/kT)$, where k is a Boltzmann constant and T is an absolute temperature; thus, the inverse Boltzmann law calculates the energies from the probabilities. Hereby, atoms in ligand and target are classified according to their molecular environment; potentials for each pair are derived from inverse Boltzmann analysis of the training dataset, and knowledge-based potential is constructed as a sum of all pairwise potentials.

- Descriptor-based (machine learning-based) methods: a relatively new group of algorithms, utilizing QSAR analysis. Unlike other types of SF, descriptor-based scoring functions don't have mathematical functional form; they make use of machine-learning algorithms with different types of descriptors used as features.

Usually, docking is implemented in an exhaustive style: the binding affinities of all molecules in the library are assessed. Therefore, a considerable part of docking time is spent on working with low-scoring molecules. The situation is getting worse as docking libraries have been growing exponentially over the past decade; for example, ZINC, a popular database of commercially available compounds for virtual screening contained roughly 1 billion molecules in 2020[6], which is eight times more than in 2015[7]. With an increase of libraries' sizes, computational costs for screening campaigns have become beyond imagination (e.g., 475 CPU-years in case of [21]). Thus, new strategies must be applied to diminish the computational costs in screening campaigns. One of the possible solutions is to combine classic molecular docking with QSAR modelling in order to remove molecules which are, according to QSAR analysis, less likely to bind the target from screening campaigns.

1.7 Related works

In a number of works, the construction of an iterative algorithm has been reported. For instance, Deep Docking[22] (DD), a platform which employs QSAR deep models trained on docking scores to iteratively predict the docking outcome for all entities in the library and put away unfavorable molecules, has been presented. Authors had chosen Morgan fingerprints as features, and

had constructed the pipeline in the following way: at initial step, the training subset is docked into the target; in all the rest of the steps, docked molecules are divided into 'hits' and 'non-hits' according to the score cut-off, the DL model is retrained on the received information and predict docking outcomes on the all entities of the library, and the predefined number of molecules selected randomly from predicted 'hits' and docked in order to augment the training set.

Another example of iterations usage is a paper[3] describing the application of the Bayesian optimization technique for docking hits search. This approach uses so called acquisition functions. They define docking for which molecules should be performed on the next iteration. Authors had tested several types of acquisition functions, but the most simple, greedy acquisition function had been estimated to be the most effective in exploring docking hits (greedy acquisition function selects the molecules which have the most negative docking scores for docking). The active learning algorithm was utilized to docking hits prioritizing in datasets of various sizes: from Enamine 10k (10,540 compounds) to ultra-large datasets used in large docking campaign[16] by Lyu et al (99.5 million molecules). A work with the ultra-large library is of the greatest interest: if the batch size is 0.4%, the message-passing neural network (MPN) model finds 87.9% of the top-50000 (ca. top-0.05%) scores after exploring 2.4% of the total pool, NN - 74.7% and random forest - 71.4%.

The information about another researches is given in Table 1.1.

Paper	Targets and Library	Fingerprints	Docking software	ML models
Efficient iterative virtual screening with Apache Spark and conformal prediction[23]	HIV-1 protease, PTPN22, MMP13 and CTDSP1; SureChEMBL (2.2 million)	signature molecular descriptor	OEDocking TK	inductive conformal prediction (ICP) approach with SVM
Deep Docking: A Deep Learning Platform for Augmentation of Structure Based Drug Discovery[22]	nuclear receptors (AR, ERα, PPARγ), linases (CAMKK2, CDK6, VEGFR2), GPCRs (ADORA2A, TBXA2R, AT1R), ion channels (Nav1.7, GLIC, GABAA3); ZINC (1.36 billion)	Morgan (radius 2, size 1024)	FRED	feed-forward NN
State of the art iterative docking with logistic regression and Morgan fingerprints[24]	99 million for AmpC, 138 milion for D4	Morgan (radius 2, size 8192,with pharmacophoric invariants)	DOCK3.7	logistic regression
Lean-Docking: Exploiting Ligands' Predicted Docking Scores to Accelerate Molecular Docking[25]	LIT-PCBA data set (15 proteins)	unfolded counted atom pairs	CCDC Gold, Glide, AutoDock-Vina, FRED ,MOE	L2-regularized linear support vector regressor (SVR)

Accelerating High-Throughput Virtual Screening Through Molecular Pool-Based Active Learning[3]	thymidylate kinase (PDB ID: 4UNN) AmpC β-lactamase (PDB ID: 12LS); Enamine10k, Enamine50k HTS collection (2.1 million) for 4UNN ZINC (99 million) for AmpC	Atom-pair (radius from 1 to 3, size 2048)	Autodock Vina, DOCK3.7	random forest, feed-forward NN, directed message passing NN
Efficient Exploration of Chemical Space with Docking and Deep-Learning[26]	D4 receptor, AmpC β-lactamase, MT1; 99 million for AmpC, 138 milion for D4, 151 million for MT1	Morgan (radius 2, size 2048)	DOCK3.7, Glide SP	graph-convolutional NN

Table 1.1: A overview of related papers

Chapter 2

Methods

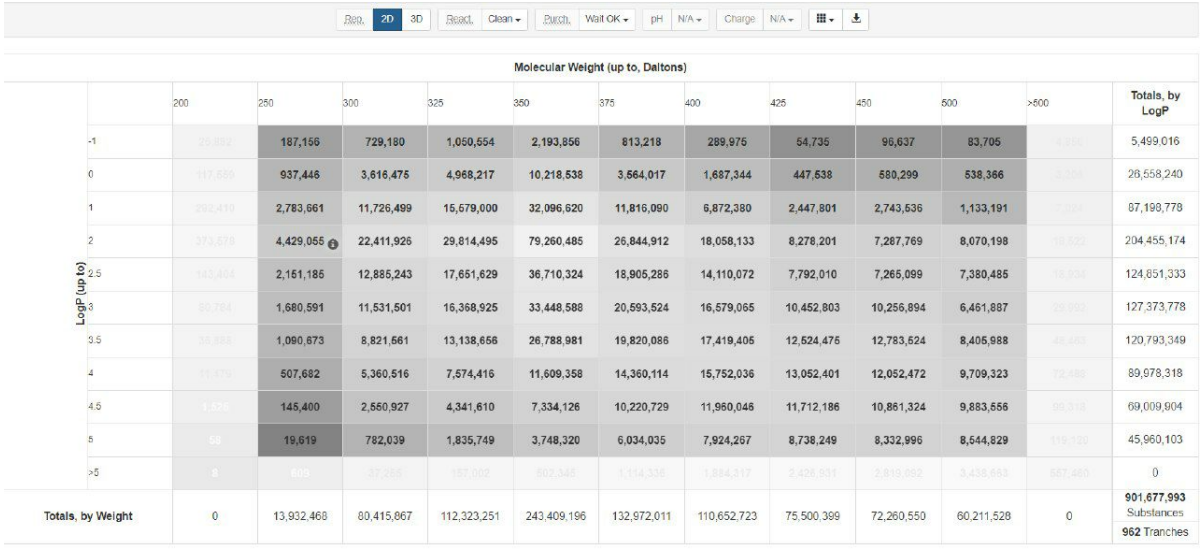
The pipeline of this work relies on the following steps:

1. Library selection;
2. Fingerprints generation;
3. ML model selection;
4. Iterative algorithm building.

All steps will be considered in detail in this chapter.

2.1 Dataset selection

SMILES of drug-like molecules from ZINC20 library were taken as a dataset for this work with following rules: $200 < MW < 500$, $logP < 5$. The distribution of number of compounds on their molecular weight and log P is showed on Fig.2.1.



LogP (up to)	200	250	300	325	350	375	400	425	450	500	>500	Totals, by LogP
-1	29,892	187,156	729,180	1,050,554	2,193,856	813,218	289,975	54,735	96,637	83,705	4,836	5,499,016
0	147,656	937,446	3,616,475	4,968,217	10,218,538	3,564,017	1,687,344	447,538	580,299	538,366	3,564	26,558,240
1	292,410	2,783,661	11,726,499	15,679,000	32,096,620	11,816,090	6,872,380	2,447,801	2,743,536	1,133,191	7,224	87,199,778
2	373,578	4,429,055	22,411,926	29,814,495	79,260,485	26,844,912	18,058,133	8,278,201	7,267,769	8,070,198	18,522	204,455,174
2.5	142,464	2,151,185	12,885,243	17,651,629	36,710,324	18,905,286	14,110,072	7,792,010	7,265,099	7,380,485	16,934	124,851,333
3	80,764	1,680,591	11,531,501	16,368,925	33,448,588	20,593,524	16,579,065	10,452,803	10,256,894	6,461,887	20,992	127,373,778
3.5	35,888	1,090,673	8,821,561	13,138,656	26,788,981	19,820,086	17,419,405	12,524,475	12,783,524	8,405,988	45,468	120,793,349
4	11,475	507,682	5,360,516	7,574,416	11,609,358	14,360,114	15,752,036	13,052,401	12,052,472	9,709,323	72,488	89,978,318
4.5	1,525	145,400	2,560,927	4,341,610	7,334,126	10,220,729	11,960,046	11,712,186	10,861,324	9,883,556	99,318	69,009,904
5	68	19,619	782,039	1,835,749	3,748,320	6,034,035	7,924,267	8,738,249	8,332,996	8,544,829	119,130	45,960,103
>5	8	605	37,255	187,092	502,348	1,114,336	1,884,317	2,428,931	2,819,692	3,438,653	567,460	0
Totals, by Weight	0	13,932,468	80,415,867	112,323,251	243,409,196	132,972,011	110,652,723	75,500,399	72,260,550	60,211,528	0	901,677,993 Substances / 962 Tranches

Figure 2.1: Drug-like molecules in ZINC20 library

2.2 Fingerprints generation

Morgan fingerprints were generated from SMILES (2048 bits in the fingerprint) using chemfp[27] 1.6.1 with following parameters:

- radius 2 or 3;
- 2048 or 4096 bits in the fingerprint;
- chemical-feature invariants are not used
- chilarity information is not included
- bond type information is included

Atom pair fingerprints were generated with RDKit module, with 2048 bits in the fingerprint, chilarity not included.

2.3 Molecular docking

Docking was performed using ICM-Pro molecular modeling software. X-ray crystal structures of Human cannabinoid receptor 2 (CNR2) and adenosine receptor A2 (AA2AR) from Protein Data Bank (PDB) were used. CNR2 models were prepared using structure with antagonist AM10257 at 2.8 A resolution (PDB ID 5ZTY), while for AA2AR models the structure with antagonist ZM241385 at 1.8 A resolution (PDB ID 4EIY) was taken. The structures were converted from PDB coordinates to ICM objects using ICM-Pro conversion algorithm.

All compounds in the screening library were converted from SMILES to SDF format, hydrogen atoms were built and formal charges were assigned at pH=7.0 according to ICM pKa Model implemented in ICM-Pro.

2.4 Machine learning model selection

2.4.1 Metrics for models performance evaluation

In order to distinguish between "good" and "bad" ML models, several metrics were used. First of them is named recall and applicable to classifiers: recall is the ratio between correctly predicted items (true positive, TP) to all items in the class of interest (class of docking hits in our case), which consist of true positive and false negative (FN) items:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

To estimate a performance of a regressor, so-called top-k% score metric was utilized:

$$\text{top-k\% score} = \frac{\text{size of the intersection}}{k/100 \cdot \text{size of the dataset}},$$

where intersection is:

$$\text{intersection} = (\text{top-k\% molecules in true rating}) \cap (\text{top-k\% molecules in predicted rating})$$

When iterative algorithms were investigated, several parameters had to be checked: single models' top-1% scores were calculated as well as the recall of the complex models. Furthermore, the percentage of docking hits from the entire dataset was examined. In this case docking hits were defined as top-1% of all items in the set.

2.4.2 Reference models

To understand how great or poor the performance of the ML models is, two types of reference models were used. First model is second docking performed with effort=1 and it is supposed that it gives the upper bond: none ML model can outperform docking. The lower bound is taken from dummy regressors and classifiers: model returning random values according to normal or uniform distribution was utilized as a dummy regressor, and a classifier which simply returns the most frequent class was chosen to be dummy classifier.

2.4.3 Classifiers vs Regressors

Firstly, both regression and classification models were examined. Regressors were taught on the docking scores, while for classifiers the scores were binarised: molecules were ranked according to their score, and top 0.5%, 1%, 2% or 5% of the dataset were considered further as docking hits.

Recall and top-k% score are two metrics convenient for comparison between regressors and classifiers. Indeed, it is possible to "transform" regressor into classifier: the cut-off may be established so k% of the molecules with the best score are to become hits; after that transformation the calculated recall turns out to be equal to top-k% score. Thence, it was assumed in analysis that if the classifier's recall is lower than the regressor's top-k score, this classifier is "worse" than that regressor.

2.4.4 Selection procedure

The following classifiers and regressors were tested:

- Linear regression;
- Ridge (and RidgeCV);
- Lasso (and LassoCV);
- KNeighbours regressor (with one and five neighbours);
- KNeighbours regressor with Jaccard metric;
- DecisionTree regressor;
- RandomForest regressor;
- KNeighbours classifier (with five neighbour);
- DecisionTree classifier;
- RandomForest classifier;
- SGD classifier.

All models were imported from scikit-learn Python library and trained over 5 cross-validation repeats with different set sizes (8000, 40000, 80000, 160000 and 320000 compounds). Besides, classifiers were trained on different types of binarised scores depending on the percentage of the docking hits (0.5%, 1%, 2% or 5%).

2.5 Iterative algorithm development

In iterative algorithms, only regression methods were used. The pipeline of the algorithm consists of the following steps:

1. In the initial step, a predefined number of molecules are docked or taken from the list of already docked compounds to provide a test set;
2. The ML model is trained and integrated then into a complex model;
3. The complex model predicts the docking outcome;
4. The batch of molecules is docked: if the number of docking hits is larger than a predefined size, then the demanded number of hits are sampled; In the opposite situation, randomly sampled molecules from non-hits are added to all docking hits in order to gain the required amount;

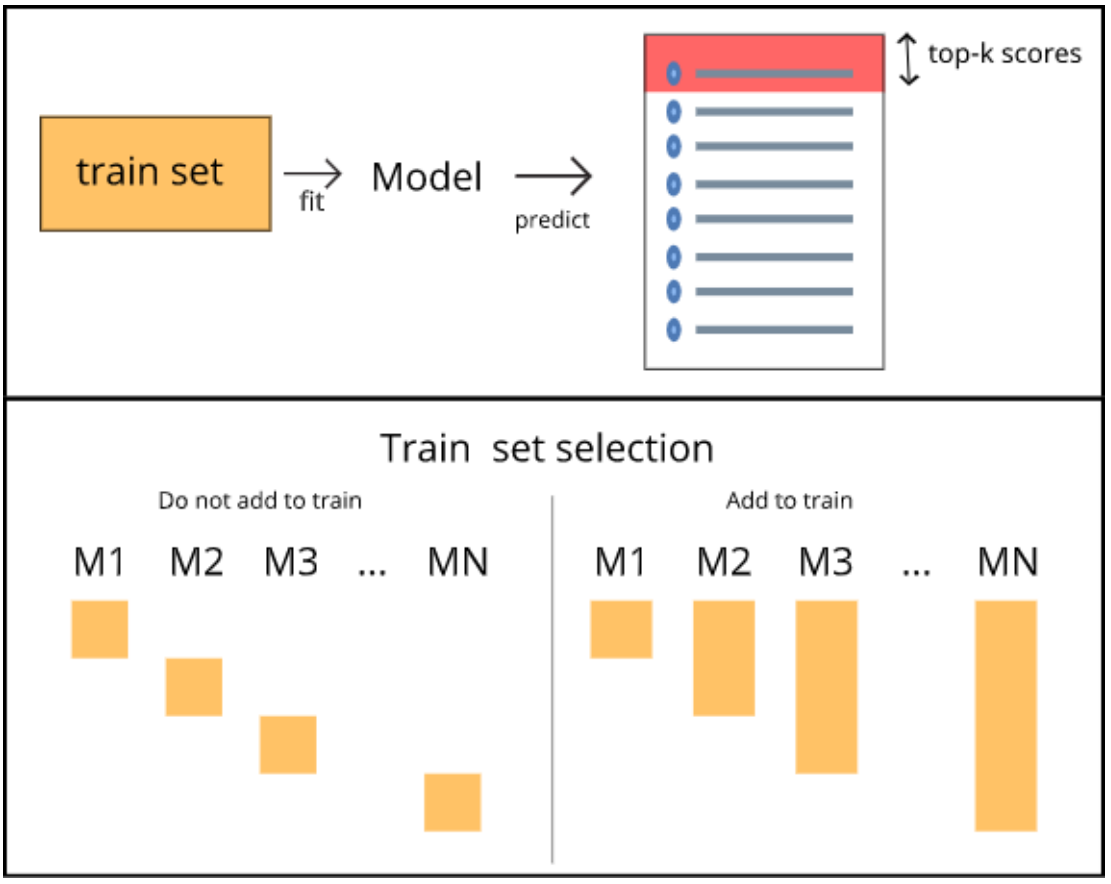


Figure 2.2: Two ways of treating the train set in iterative algorithm

5. Depending on the algorithm protocol (Fig. 2.2), the batch from the precious step is either added to the train dataset, or is utilized as a train dataset itself;

6. Steps 2-5 are repeated until the predetermined number of iteration is reached.

2.5.1 Train set selection

There are two ways to update train dataset from iteration to iteration (Fig. 2.2). On the one hand, it is profitable to make use of all the docked molecules, because more information is gathered. Most likely that single models in one round of iteration algorithm will be similar to each other and prioritize the molecules which properties close to each other. On the other hand, the selection of only the latest batch of docked molecules for training may cause more diversity in single models' parameters. Thus, models from different step will search for docking hits in various parts of chemical space. Both approaches were tested within the iterative algorithms.

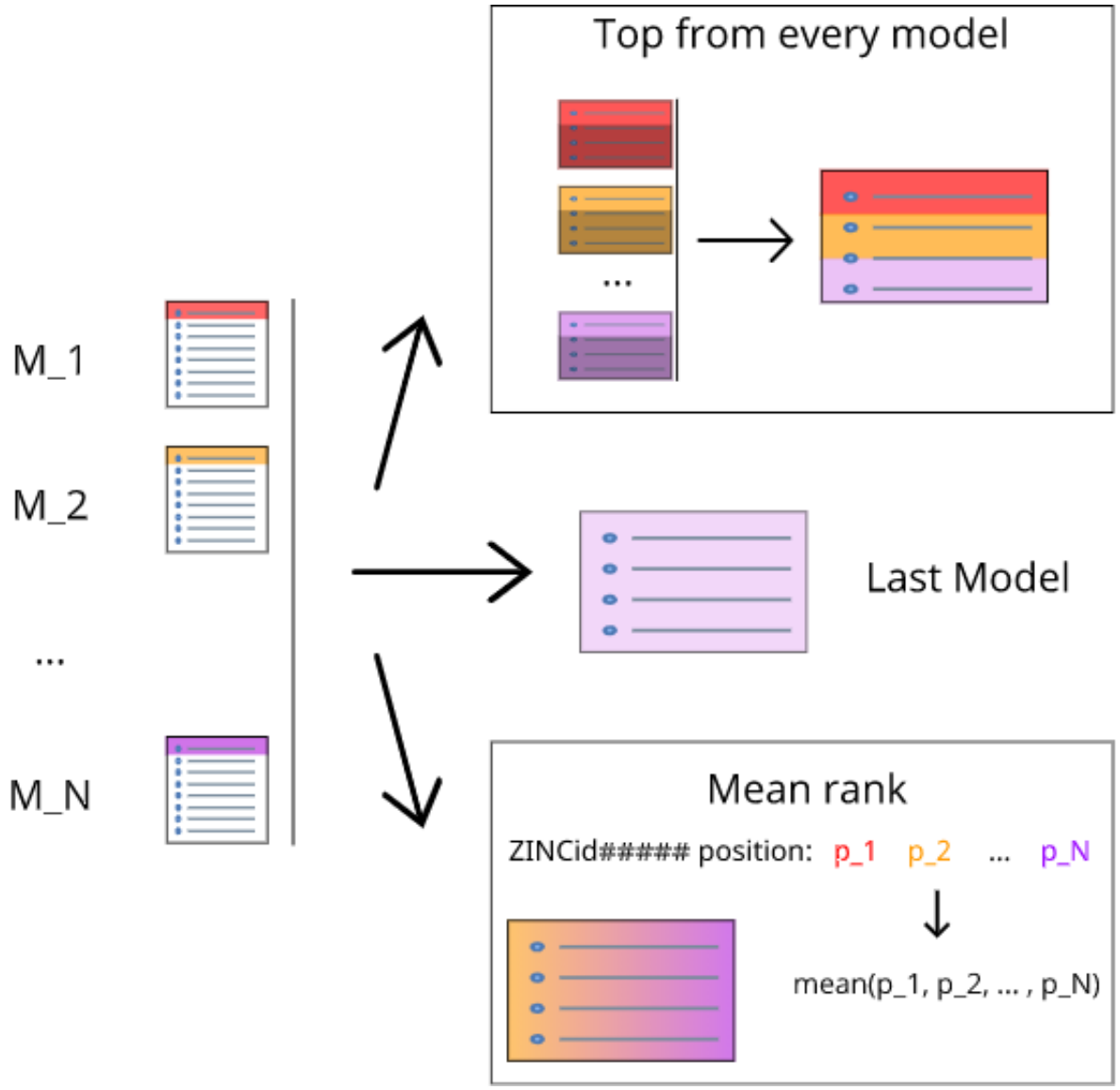


Figure 2.3: Types of complex model

2.5.2 Complex model creation

If there are several ML models available, it is possible to combine them somehow to create a complex model. Three types of complex models were employed in the algorithm (Fig. 2.3):

- **Last Model**: ML model from the latest iteration is treated as a complex model and used to predict docking outcomes. After the prediction, top-k% of predicted scores become hits;
- **Top from every model**: on the n^{th} iteration, each model makes a prediction and top-(k/n)% of predicted scores from each model constitute complex model's hits (repetitions are deleted);
- **Mean Rank**: for each compound, the positions in the ratings of all models are averaged; the k% of the compounds with the smallest average positions are handled as hits.

2.6 Linear model visualisation

If complex model is composed of linear regressions, the coefficients of single models can be visualised with help of t-SNE, t-distributed stochastic neighbor embedding. This visualisation can help to analyse whether single models in the iterative algorithm are diverse or not.

The coefficients of linear regression models were normalised so the sum of their squares was equal 1. TSNE algorithm from scikit-learn library was utilised to project the 2048-dimensional space of coefficients to 2D space.

Chapter 3

Results & Discussion

3.1 Single model parameters

3.1.1 Choosing between regressors and classifiers

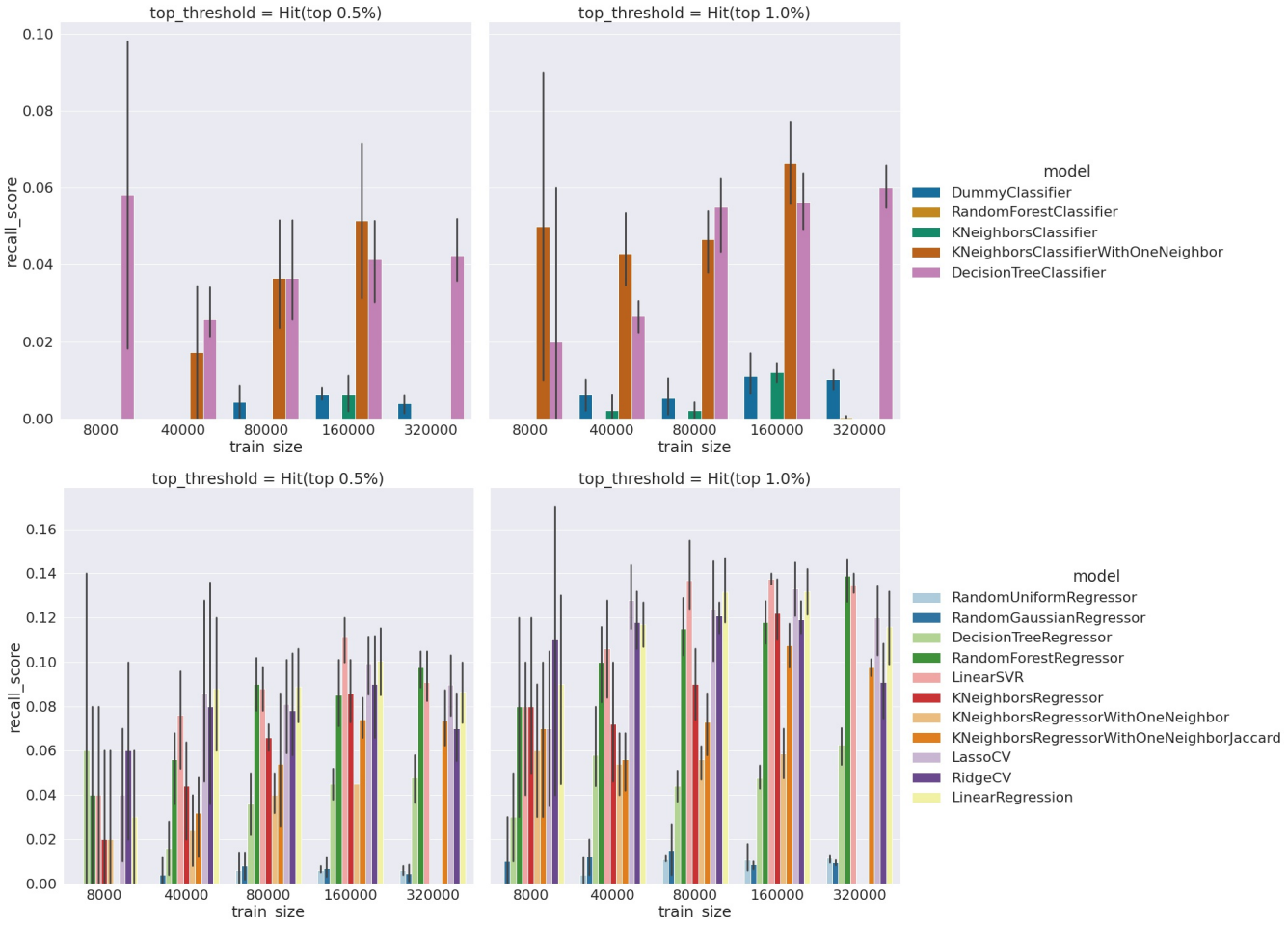


Figure 3.1: Comparison between classifiers and regressors trained on Morgan fingerprints with radius 2 and size 2048

Initially, Morgan fingerprints with a radius of 2 and a size of 2048 bits were taken as fingerprints for predicting docking hits. The first goal was to determine whether classifiers or regressors have better predicting ability. Fig. 3.1 shows that most regressors perform better than classifiers (baseline models, such as RandomUniformRegressor and RandomGaussianRegressor should not be included in consideration).

3.1.2 Morgan fingerprints

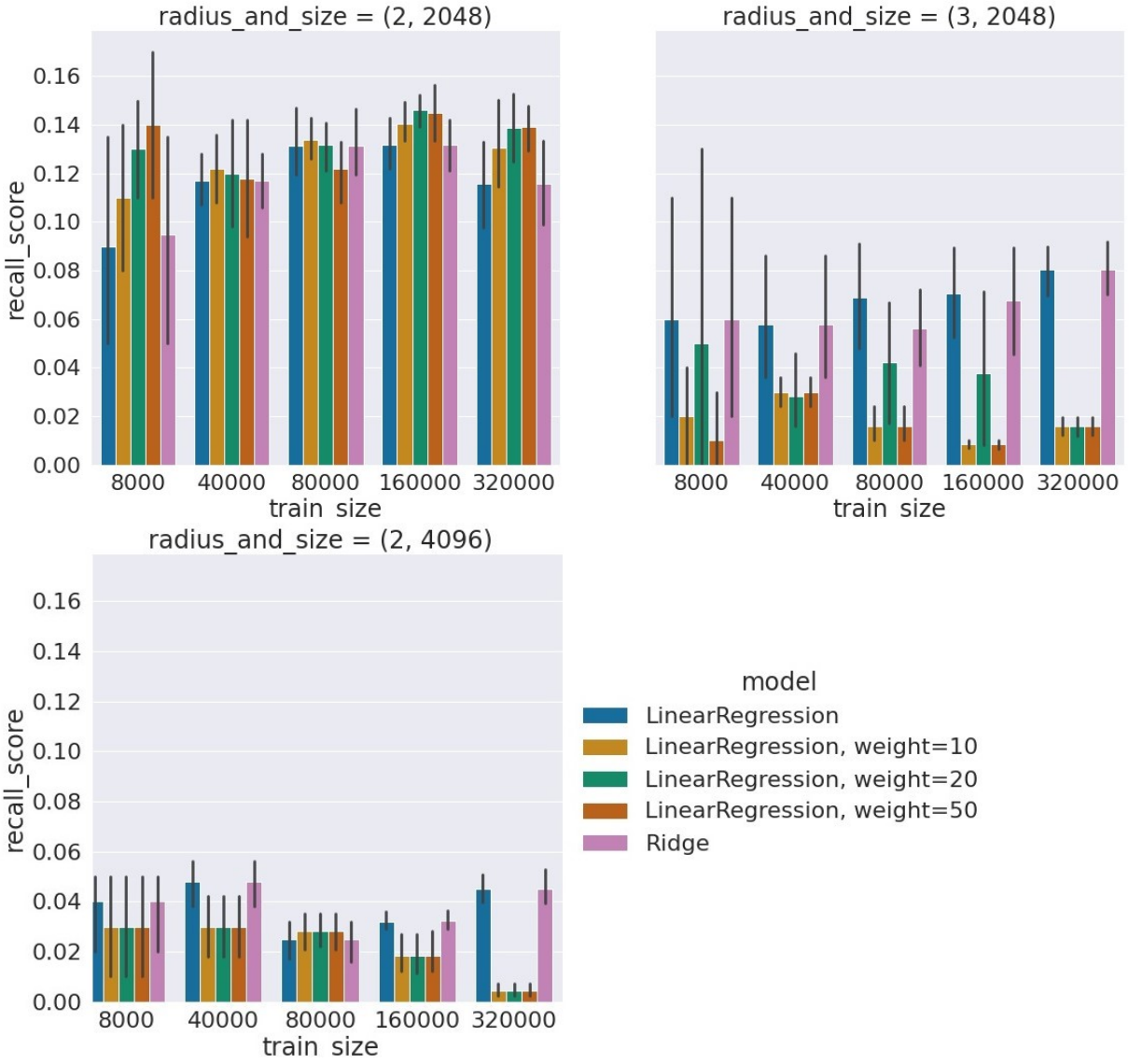


Figure 3.2: Comparison of models trained on Morgan fingerprints

Default radius and size for Morgan fingerprints in RDKit/chemfp are 2 and 2048. However, it can be assumed that another combination of parameters may contribute to more accurate prediction result. For example, increase in the size of the fingerprint allows to distinguish more

precisely between encoded molecular structures. On the other hand, multiplication of the number of features can deteriorate the quality of the machine learning model trained on the set of the constant size. Changing the radius of the fingerprints means encoding larger substructures in the same number of bits. Analysis of the quality of predictors trained on the Morgan fingerprints with radius 2 or 3 and size 2048 or 4096 revealed, that default parameters are still the best for prediction.

3.1.3 Type of fingerprints

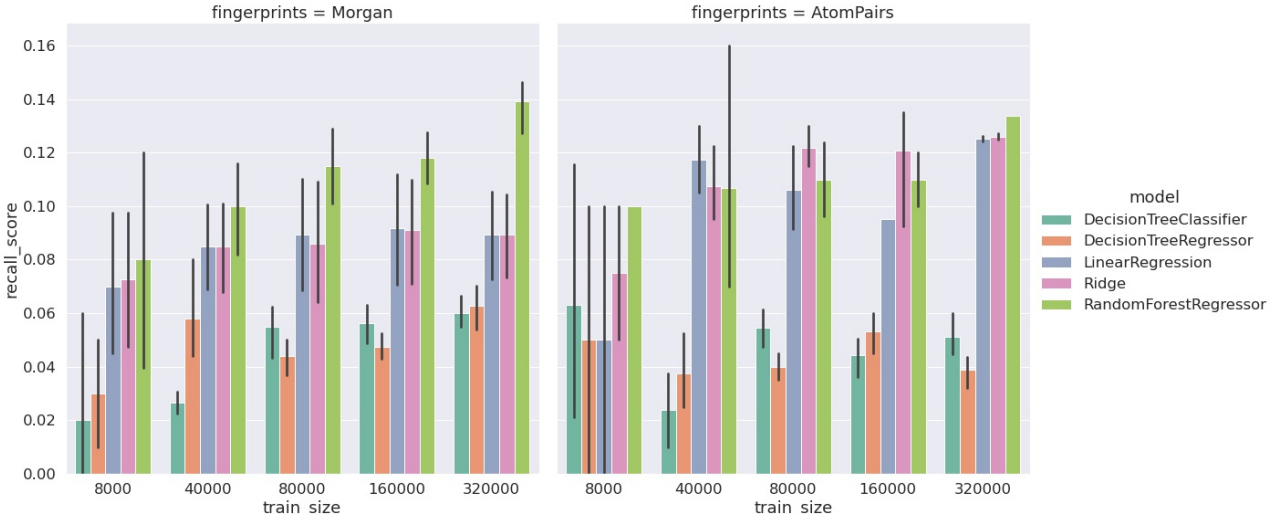


Figure 3.3: Comparison of some models trained on Morgan and Atom pair fingerprints

As long as Morgan fingerprints are not the only one utilized in QSAR modelling, another type of fingerprints could be used for prediction. Thereby, atom pair fingerprints were generated for the dataset and used for training. Models, which were trained on Morgan and atom pair fingerprints, have showed comparable results (Fig. 3.3), and further work has been done using Morgan fingerprints.

3.1.4 Comparing with docking

Linear regression has been chosen as a single model for an iterative algorithm. It was one of the best performing models during the analysis, it was the quickest one in training and prediction, and results given by the linear regression can be interpreted: according to the weight of each bit of the fingerprints it is possible to suggest which chemical substructures play significant role in attraction/repulsion between the binding site and a small molecule. Also, the quality

of the linear regression can be improved at no cost by adding weights to molecules which are considered as docking hits.

Recall of the linear regression has already been compared with "dummy" regressors which give a random number from a unifrom or a gaussian distribution. The upper estimate should also be done through the use of the independent round of docking with effort=1 (Fig. 3.4).

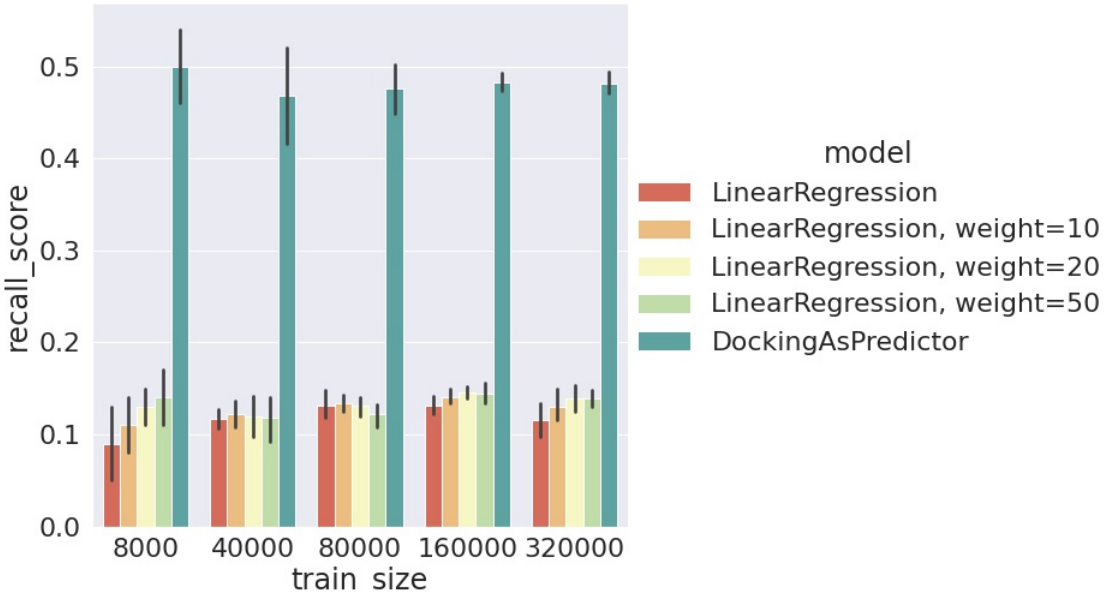


Figure 3.4: Models based on linear regression compared with docking

3.2 Complex model

3.2.1 Determination of the best parameters

Iterative algorithm with "add" and "noadd" train set augmentation strategies, "LastModel", "MeanRank" and "TopFromEveryModel" complex model types was tested. Firstly, independent round of docking was utilized as a single model. The percentage of hits discovered by the algorithm based on docking can give the upper bond for the effectiveness of the technique: no one machine learning model can distinguish between hits and non-hits better than docking. Hereby, iterative algorithm reaches values of about 80% with 25% of docked molecules when using docking as a single model, as seen at Fig 3.6.

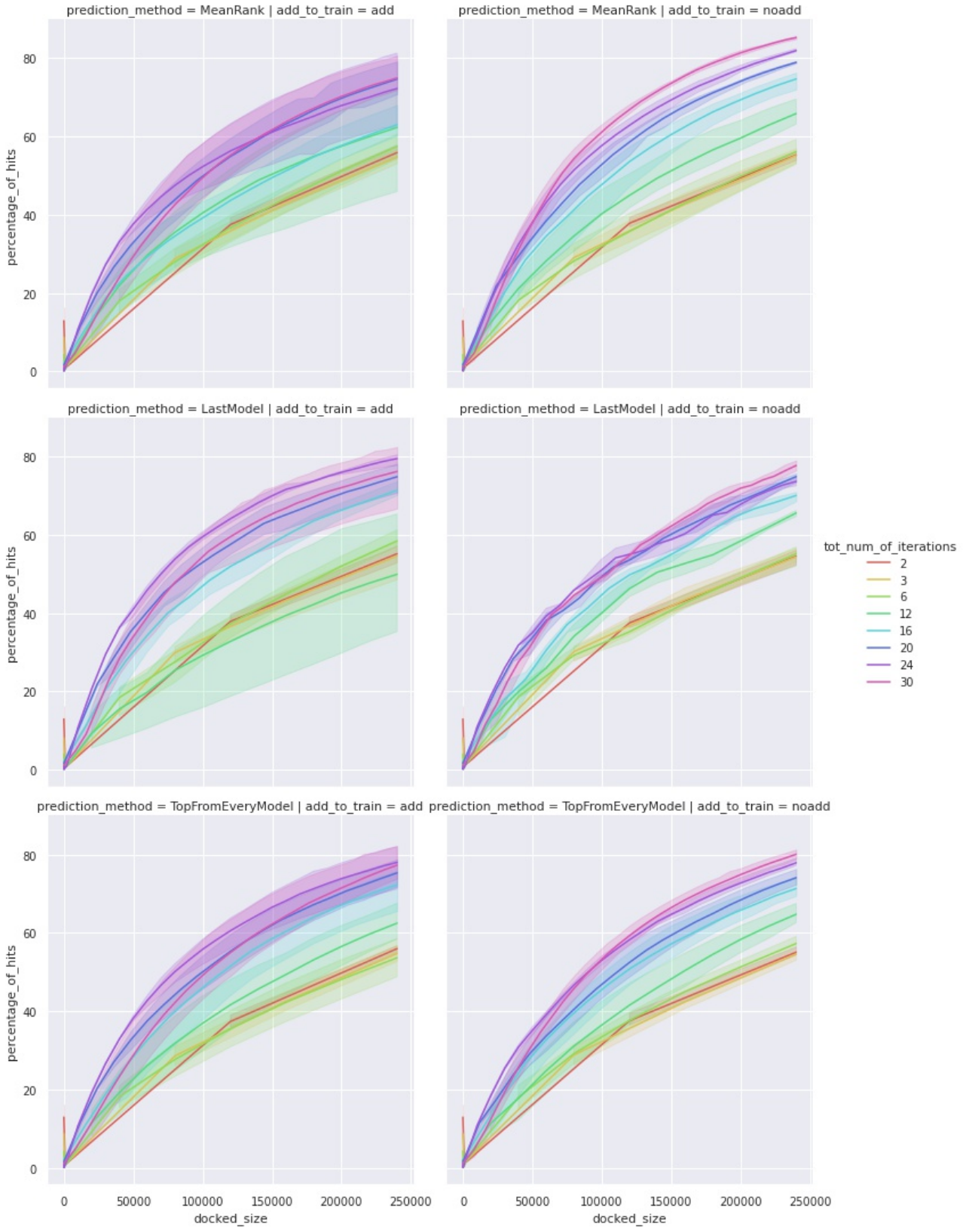


Figure 3.5: The portion of docking hits discovered by iterative algorithm depending on its parameters, the number of iterations and the amount of docked molecules

Thus, variation of the algorithm based on the linear regression can be considered as effective if the fraction of discovered hits will not differ much from 80% with 25% of docked molecules. Examination of all variations of the iterative algorithm has led to the conclusion that best performance gives the algorithm with "noadd" train set acquisition stratedy and "MeanRank" complex model.

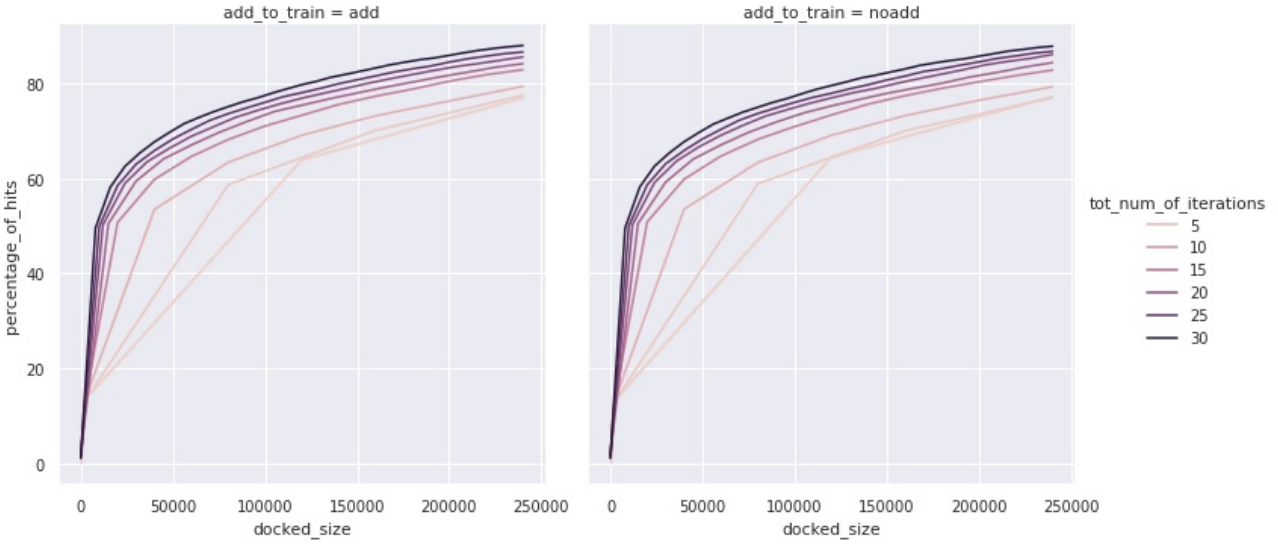
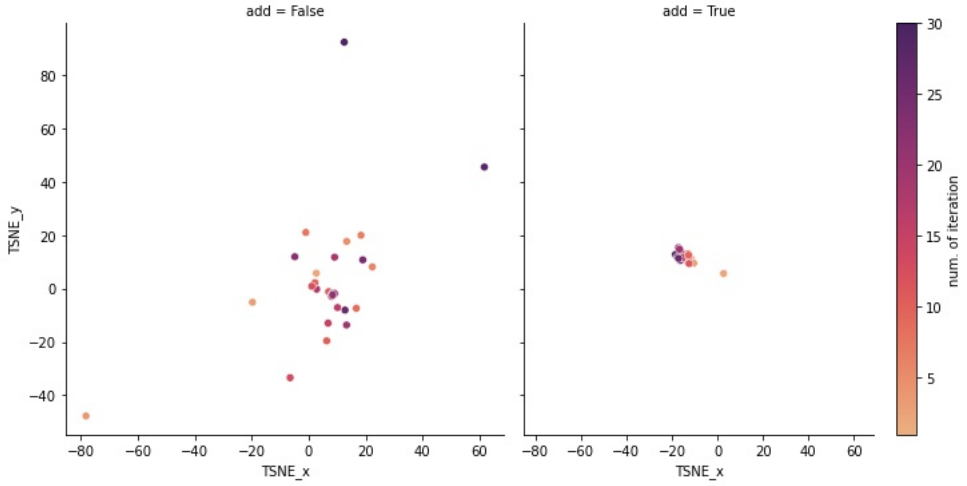


Figure 3.6: Performance of the iterative algorithm with docking as a single model

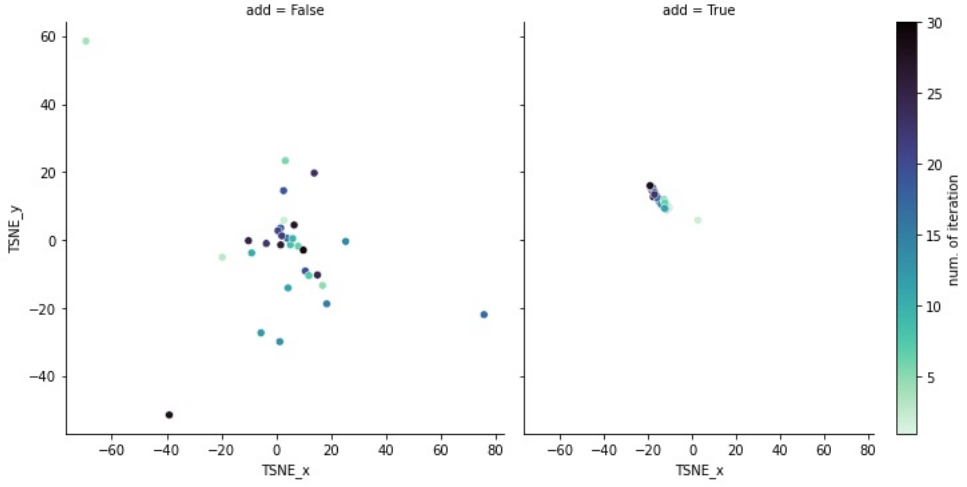
Besides, it it worth to notice difference common for all complex models depending on "add" or "noadd" strategy they use: when a certain number of iterations is reached, the quality of the algorithm with "noadd" strategy begins to deteriorate. This can be explained by lower variety of single models in the iterative algorithm: when training set does consists partly of "old" molecules, model on next step does not differ from the one from the previous step. The fewer molecules are added during the iterations, the less significant is the difference between models. Thus, when number of iterations is big enough, models start to pick molecules in the same area of chemical space, ommiting higher part of the hits. On the other hand, when all models are taught on the independent subsets of docked molecules, they show higher diversity. That means that models prioritize separate fields of chemical space, which allow to gather more docking hits.

3.2.2 Exploring the variety of the models in the algorithm

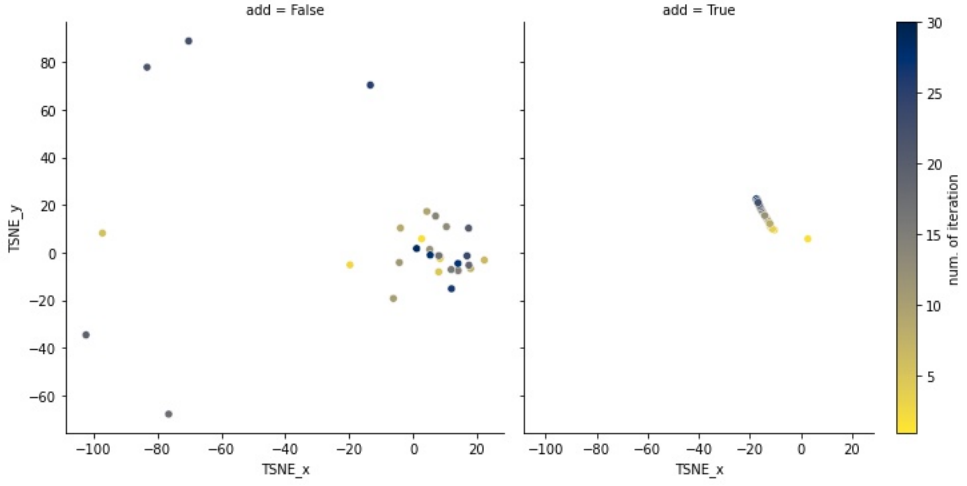
The statement about higher variety of "noadd" models can be proved thanks to simplicity of a linear regression. The 2048-dimensional space of coefficients of all trained models in the



(a) "LastModel"



(b) "TopFromEveryModel"



(c) "MeanRank"

Figure 3.7: Comparison of varieties of model obtained when using "add" and "noadd" strategies

algorithm can be projected on plane using t-sne. Result of projection is shown on 3.7. It allows to confirm the assumption about greater variety of models trained in iterative algorithm with "noadd" train set augmentation strategy.

3.2.3 Comparison of the iterative algorithm with exhaustive docking

As already mentioned above, iterative algorithm shows best performance with "noadd" train set acquisition strategy and "MeanRank" complex model. The comparison between the iterative algorithm, with best parameters, with docking or linear regression as a simple model, is shown on Fig. 3.8.

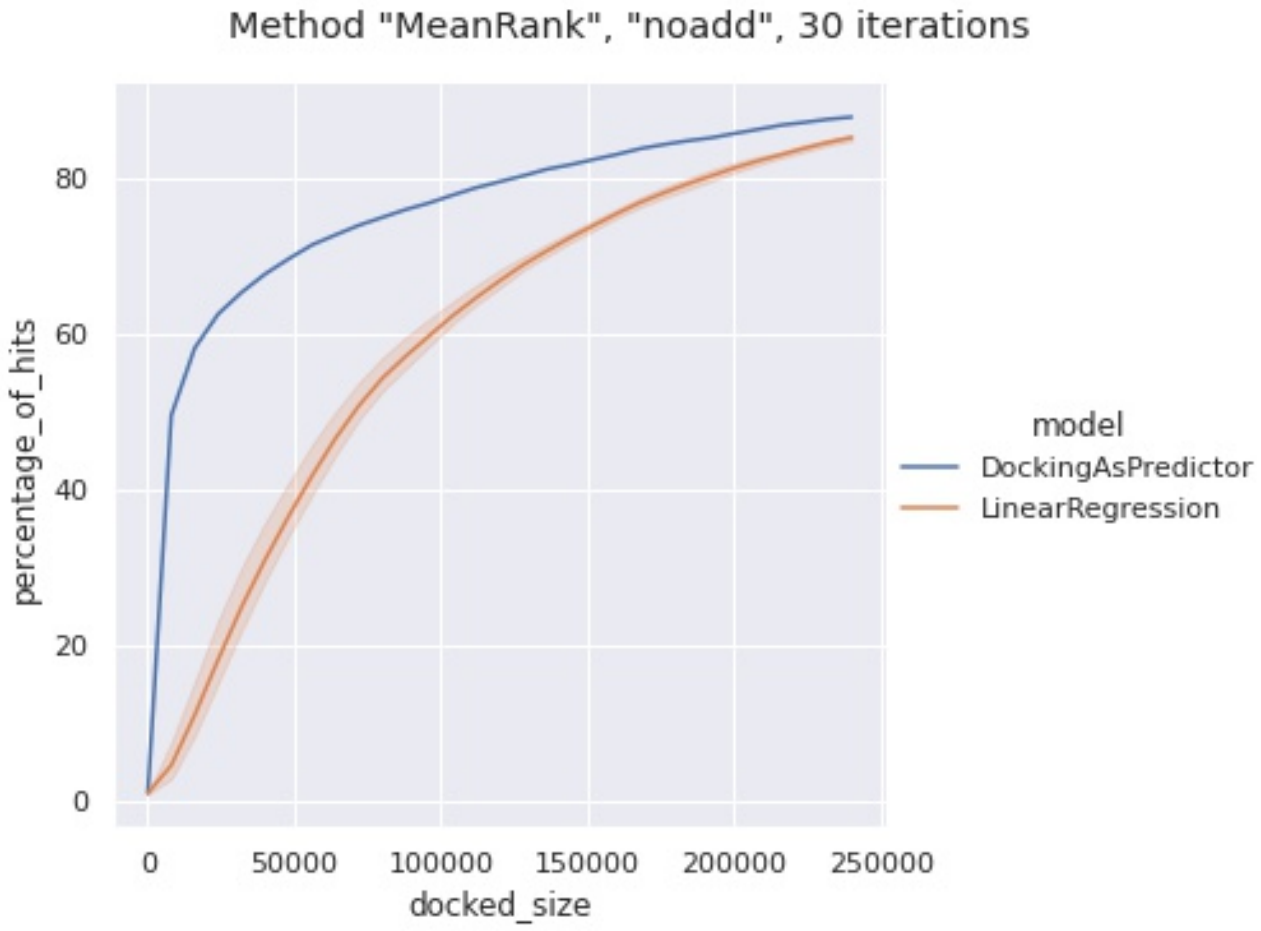


Figure 3.8: Best performing variation of iterative algorithm compared with docking

The graph shows that if the size of docked molecules reaches 25% of the set, then the number of hits predicted by linear regression and docking differs by less than 3%, what can be considered as an insignificant loss of predictive ability. It is also possible to compare the efficiency and spent time for docking and an iterative algorithm based on linear regression. Since the prediction time by linear regression is negligible compared to the docking time, it can be neglected when estimating the time spent. Hence, 4-fold reduction of time allows to receive 85% of docking hits and 10-fold reduction - 60% of them.

Chapter 4

Conclusion

The work aimed to reduce the time required for molecular docking by throwing out the molecules which are less prone to binding to a target. The results of this work are listed below:

- Performance of regressors and classifiers trained on the Morgan fingerprints with default parameters have been analysed. Results have indicated that higher recall was achieved when working with regressors.
- Different parameters (radius, size) for the Morgan fingerprint have been utilized in predictions. Best recall have showed regressors trained on Morgan fingerprints with default parameters.
- Morgan fingerprints with default parameters have been compared with atom pairs fingerprints. Models trained on these fingerprints have had comparable results, so further work have been done using Morgan fingerprints with radius=2 and size=2048.
- Iterative algorithm with various approaches to train set augmentation ("add" or "noadd") and complex model creation ("LastModel", "MeanRank", "TopFromEveyModel") has been developed.
- The percentage of hits discovered by the iterative algorithm with linear regression as a single model has been evaluated for all variations of the algorithm depending on the amount of iterations. The algorithm with "MeanRank" complex model and "noadd" train set augmentation approach has turned out to be the one with the highest fraction of discovered hits.
- Iterative algorithm has been compared to docking. The number of hits discovered with 4-fold reduction in time compared to exhaustive docking has been estimated as 80%, with 10-fold reduction - 60%.

Bibliography

[1] Blass, B. E. Basic Principles of Drug Discovery and Development / Benjamin E. Blass. — 2015. — P. 1–574.

[2] Maveyraud, L. Protein X-ray crystallography and drug discovery / Laurent Maveyraud, Lionel Mourey // Molecules. — 2020. — Vol. 25, no. 5.

[3] Graff, D. E. Accelerating high-throughput virtual screening through molecular pool-based active learning / David E. Graff, Eugene I. Shakhnovich, Connor W. Coley // arXiv. — 2020.

[4] Bohacek, R. S. The art and practice of structure-based drug design: A molecular modeling perspective / Regine S. Bohacek, Colin McMartin, Wayne C. Guida // Medicinal Research Reviews. — 1996. — Vol. 16, no. 1. — P. 3–50.

[5] Gilson, M. K. An Introduction to Protein-Ligand Binding for BindingDB Users / Michael K Gilson. — 2010. — P. 1–12.

[6] ZINC20 - A Free Ultralarge-Scale Chemical Database for Ligand Discovery / John J. Irwin, Khanh G. Tang, Jennifer Young et al. // Journal of Chemical Information and Modeling. — 2020. — Vol. 60, no. 12. — P. 6065–6073.

[7] Sterling, T. ZINC 15 - Ligand Discovery for Everyone / Teague Sterling, John J. Irwin // Journal of Chemical Information and Modeling. — 2015. — Vol. 55, no. 11. — P. 2324–2337.

[8] Polanski, J. Receptor Dependent Multidimensional QSAR for Modeling Drug - Receptor Interactions / Jaroslaw Polanski // Current Medicinal Chemistry. — 2009. — Vol. 16, no. 25. — P. 3243–3257.

[9] Hansch, C. ρ-σ-π Analysis. A Method for the Correlation of Biological Activity and Chemical Structure / Corwin Hansch, Toshio Fujita // Journal of the American Chemical Society. — 1964. — Vol. 86, no. 8. — P. 1616–1626.

[10] Debnath, A. Quantitative Structure-Activity Relationship (QSAR) Paradigm - Hansch Era to New Millennium / Asim Debnath // Mini-Reviews in Medicinal Chemistry. — 2005. — Vol. 1, no. 2. — P. 187–195.

[11] Todeschini, R. Methods and Principles in Medicinal Chemistry / Roberto Todeschini, Viviana Consonni. — 2007. — P. 438–438.

[12] Baskin, I. Introduction in chemoinformatics (in russian) / Igor Baskin, T Majitov, A Warneck. — 2020. — P. 296.

[13] QSAR modeling: Where have you been? Where are you going to? / Artem Cherkasov, Eugene N. Muratov, Denis Fourches et al. // Journal of Medicinal Chemistry. — 2014. — Vol. 57, no. 12. — P. 4977–5010.

[14] Rogers, D. Extended-Connectivity Fingerprints / David Rogers, Mathew Hahn. — 2010. — P. 742–754.

[15] Weininger, D. SMILES. 2. Algorithm for Generation of Unique SMILES Notation / David Weininger, Arthur Weininger, Joseph L. Weininger // Journal of Chemical Information and Computer Sciences. — 1989. — Vol. 29, no. 2. — P. 97–101.

[16] Ultra-large library docking for discovering new chemotypes / Jiankun Lyu, Sheng Wang, Trent E. Balius et al. // Nature. — 2019. — Vol. 566, no. 7743. — P. 224–229. — http://dx.doi.org/10.1038/s41586-019-0917-9.

[17] Insights into protein–ligand interactions: Mechanisms, models, and methods / Xing Du, Yi Li, Yuan Ling Xia et al. // International Journal of Molecular Sciences. — 2016. — Vol. 17, no. 2. — P. 1–34.

[18] Totrov, M. Flexible ligand docking to multiple receptor conformations: a practical alternative / Maxim Totrov, Ruben Abagyan // Current Opinion in Structural Biology. — 2008. — Vol. 18, no. 2. — P. 178–184.

[19] Liu, J. Classification of current scoring functions / Jie Liu, Renxiao Wang // Journal of Chemical Information and Modeling. — 2015. — Vol. 55, no. 3. — P. 475–482.

[20] Li, J. An Overview of Scoring Functions Used for Protein–Ligand Interactions in Molecular Docking / Jin Li, Ailing Fu, Le Zhang // Interdisciplinary Sciences: Computational Life Sciences. — 2019. — Vol. 11, no. 2. — P. 320–328. — http://dx.doi.org/10.1007/s12539-019-00327-w.

[21] An open-source drug discovery platform enables ultra-large virtual screens / Christoph Gorgulla, Andras Boeszoermenyi, Zi Fu Wang et al. // Nature. — 2020. — Vol. 580, no. 7805. — P. 663–668.

[22] Deep Docking: A Deep Learning Platform for Augmentation of Structure Based Drug Discovery / Francesco Gentile, Vibudh Agrawal, Michael Hsing et al. // ACS Central Science. — 2020. — Vol. 6, no. 6. — P. 939–949.

[23] Efficient iterative virtual screening with Apache Spark and conformal prediction / Laeeq Ahmed, Valentin Georgiev, Marco Capuccini et al. // Journal of Cheminformatics. — 2018. — Vol. 10, no. 1. — P. 4–11. — https://doi.org/10.1186/s13321-018-0265-z.

[24] State of the art iterative docking with logistic regression and Morgan fingerprints: Rep. / The University of Sydney, Brain and Mind Centre, The Lambert Initiative for Cannabinoid Therapeutics; Executor: Lewis J Martin. — Sydney, NSW, Australia: 2021. — https://chemrxiv.org/articles/preprint/State_of_the_Art_Iterative_Docking_with_Logistic_Regression_and_Morgan_Fingerprints/14348117.

[25] Lean-Docking: Exploiting Ligands' Predicted Docking Scores to Accelerate Molecular Docking / Francois Berenger, Ashutosh Kumar, Kam Y. J. Zhang, Yoshihiro Yamanishi // Journal of Chemical Information and Modeling. — 2021. — no. Dd.

[26] Efficient Exploration of Chemical Space with Docking and Deep-Learning Efficient Exploration of Chemical Space with Docking and Deep-Learning / Ying Yang, Kun Yao, Matthew P Repasky et al. — 2021. — no. 2.

[27] Dalke, A. The chemfp project / Andrew Dalke // Journal of Cheminformatics. — 2019. — Vol. 11, no. 1. — P. 1–21. — https://doi.org/10.1186/s13321-019-0398-8.

List of Figures

List of Tables

Acronyms

ADMET absorption, distribution, metabolism, excretion and toxicity. 4

DD Deep Docking. 12

DL deep learning. 13

HTS high throughput screening. 5

MPN message-passing neural network. 13

PDB Protein Data Bank. 17

QSAR quantitative structure-activity relationship. 5–8, 12

SF scoring functions. 11

SMILES simplified molecular-input line-entry system. 7, 16, 17