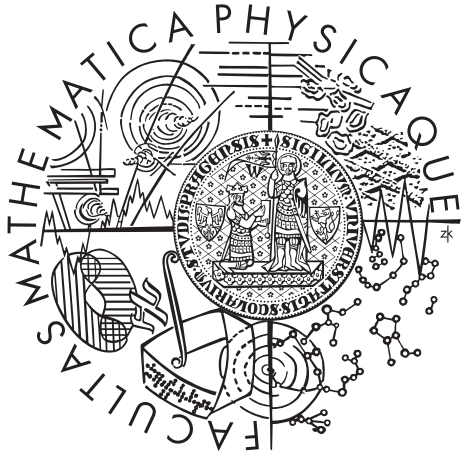
Charles University in Prague
Faculty of Mathematics and Physics

**MASTER THESIS**




Petr Škoda

# Návrh efektivní generické molekulární reprezentace

Department of Software Engineering

Supervisor of the master thesis: RNDr. David Hoksza, Ph.D.
Study programme: Informatics
Specialization: Software Systems

Prague 2014

I declare that I carried out this master thesis independently, and only with the cited sources, literature and other professional sources.

I understand that my work relates to the rights and obligations under the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular the fact that the Charles University in Prague has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 paragraph 1 of the Copyright Act.

In ..... date ..... signature of the author

Název práce: Návrh efektivní generické molekulární reprezentace

Autor: Petr Škoda

Katedra: Katedra softwarového inženýrství

Vedoucí diplomové práce: RNDr. David Hoksza, Ph.D, Katedra softwarového inženýrství

Abstrakt: Screening chemických knihoven je důležitou součástí výzkumu léčiv. Existující chemické knihovny obsahují miliony sloučenin. Screening v takovém rozsahu je velice nákladný, z tohoto důvodu je často používáno virtuálního screeningu. Existuje několik variant virtuálního screeningu, ligand-based virtual screening je jednou z nich. Využívá podobnosti screenovaných sloučenin ke známým sloučeninám. Na výsledky virutálního screeningu má vliv nejen použitá podobnostní metoda, ale také zvolená reprezentace chemických sloučenin. V této práci prezentujeme reprezentaci chemických sloučenin založenou na fingerprintech. Naše reprezentace využívá fragmentů chemické sloučeniny k její reprezentaci jakožto celku. Každý fragment je reprezentován fyzikálně-chemickými vlastnostmi. Reprezentace je vysoce parametrizovatelá a to zejména v oblasti výběru fyzikálně-chemických vlastností a jejich aplikace. Pro otestování naší reprezentace jsme využili existujícího frameworku pro benchmark virtuálních screeningů. Výsledky ukázaly, že naše reprezentace je srovnatelná s nejlepšími existujícími, navíc na některých datasetech dosáhla nejlepších výsledků.

Klíčová slova: fingerprinty, reprezentace, deskriptory

Title: Efficient molecular representation design

Author: Petr Škoda

Department: Department of Software Engineering

Supervisor: RNDr. David Hoksza, Ph.D, Department of Software Engineering

Abstract: The screening of chemical libraries is an important step in the drug discovery process. The existing chemical libraries contain up to millions of compounds. As the screening at such scale is expensive, the virtual screening is often utilized. There exist several variants of virtual screening and ligand-based virtual screening is one of them. It utilizes the similarity of screened chemical compounds to known compounds. Besides the employed similarity measure, another aspect greatly influencing the performance of ligand-based virtual screening is the chosen chemical compound representation. In this thesis, we introduce a fragment-based representation of chemical compounds. Our representation utilizes fragments to represent a compound. Each fragment is represented by its physicochemical descriptors. The representation is highly parameterizable, especially in the area of physicochemical descriptors selection and application. In order to test the performance of our method, we utilized an existing framework for virtual screening benchmarking. The results show that our method is comparable to the best existing approaches and on some datasets it outperforms them.

Keywords: fingerprint, representation, descriptors

# Contents

# Introduction

Chemical reactions are vital for all living organisms. A special role play protein-ligand reactions. Proteins can be considered as the building blocks for many life forms from simple prokaryotes to humans. As the protein-ligand reactions can modify the proteins behaviour they are not less important than the protein itself. Many diseases are caused by small organisms (viruses, bacteria, parasites). In order to treat a disease we can try to map its lifecycle and determine which chemical compounds are involved. If the suitable chemical compounds are identified and a proper ligand that changes the behaviour of that compound is found, then a treatment can be delivered for the given organism.

A process of testing whether chemical compounds react (is active) with other chemical compound (target) is called screening. Today's high-throughput screening technologies can test (screen) about thousands of compounds per day. With such throughput the problem is not in speed but in the availability of chemical compounds. Just managing stock of 100.000 compounds could be difficult and quite expensive.

The computer science solution of this problem is called virtual screening. Virtual screening is an in silico (computer based) version of screening and can easily screen millions of compounds in a short timespan. The drawback of virtual screening is the issue of quality of the screening. The possibility of obtaining false-positive and false-negative results is an experimental screening problem too, but for virtual screening this problem occurs significantly more often. There are basically two types of virtual screening, structure-based and ligand-based. In this thesis we focus on ligand-based virtual screening, specifically on a one part of the screening which - the design of a compounds representation and a similarity measure utilized for identification of active compounds. Given a set of known actives, the ligand-based virtual screening tries to identify other possibly active chemical compounds from given library of chemical compounds. The process of ligand-based virtual screening is simple. First the chemical compounds are loaded and converted into selected representation, after that the similarity methods are applied to compute the similarity of known actives and the screened chemical compounds. The similarity measure is used to prioritize the library with the motivation that similarity and activity correlate for well-designed similarity measure.

In this thesis we describe several existing approaches and propose our own. Our approach is basically a crossover of two existing approaches. We also provide implementation of our method in python, so it could be easily used by scientists. The recently published framework for benchmarking of ligand-based virtual screening methods has been utilized in the experimental evaluation of our method.

# 1. Chemical background

Proteins are macromolecules present in every cell and take an important role in many biological process. They control how cells communicate (signal function), speed up or inhibit chemical reactions (catalytic function), transport various compounds and they are also important for building tissues [1]. As the proteins play a key role in biochemical process, their study is essential for understanding the processes in living organisms.

Proteins mostly react with ligands by weak protein-ligand interactions (non-covalent and ion bond, hydrogen bridges). The term 'ligand' can be used for different chemical compounds, but it is commonly used to denote the molecule that reacts with given target (protein in our case). Ligands can be small molecules, macromolecules and even proteins. Protein-ligand reactions are highly specific and therefore a protein forms a complex (reacts) with only a small amounts of ligands that are present in a living organism.

A protein interacts with a ligand by weak interactions on atom basis so multiple weak interactions have to be formed to build a stable complex. This is only possible if the spatial distribution of chemical properties of a ligand and of the active site of a protein (cavity) are complementar. Therefore the structure of the protein-ligand active site restricts the ligand that can bond to this cavity. A single protein typically has more than one active site which is dependent on the whole protein structure (conformation). Thus, the protein conformation has a great effect on the protein's function and chemo-physical properties (charge distribution, hydrophobicity, ..). As a ligand forms a complex with a protein it actually may change the protein's conformation, the size of such change can vary based on the ligand and protein. Actually, during the reaction both the protein and the ligand can go through many conformations until both reach the equilibrium state While the protein function is dependent on its conformation and ligand can change the protein conformation, the ligand can change the protein's function . This fact is vital for many biochemical processes and can be used by scientist to modify these processes. As many forms of diseases can also be viewed as a sort of biological processes they can be targeted (cured) by the right ligands. An important fact to emphasize is that there the establishment of a protein-ligand bond is not affected by the structure by itself but also the physicochemical properties have to be taken into account. Therefore, two different proteins with different conformation can form a complex with the same ligand.

In order to target a disease on this level scientists have to identify the right protein, find its active sites and find a ligand that would bond and modify the function of the protein in a desired way.

There exist approaches that try to predict the different aspects of a chemical reaction. These approaches commonly use machine learning (utilization of existing knowledge) or try to simulate the whole reaction (docking). While the first method is based more on statics, the latter one focuses on simulation of the reactant interaction with the site where it reacts (docks). It may look like the docking approach is the ultimate tool since it models the process and should be therefore able to predict the output of a chemical reaction. However, docking relies on given 3D conformation (structure) of molecules which can be arranged

in many different conformations (poses) in the real environment. Even more importantly, the functions assessing the energy of a protein-ligand complex (given a specific pose) are only approximations of the real binding energy. So the best way how to verify a reaction of a given set of reactants under given conditions is the experimental verification.

## 1.1 Drug discovery

Innovations in chemistry and health science have dramatically changed the quality of life and the way how various diseases are treated. The discovery of penicillin in 1929 by Alexander Fleming opened the door to the new era in the treatment of bacterial infections. Biochemistry (the study of chemical processes within and relating to a living organism) is a source of innovation in the drug discovery process. Jürgen Drews [2] states that "The dominant concepts introduced by biochemistry were those of enzymes and receptors, which were empirically found to be good drug targets". This brings new possibilities into the drug discovery. Together with the knowledge of biological pathways the scientist can now research drugs with specific effects.

The scientist hope to understand the disease on the molecular level and find optimal macromolecular targets for drugs intervention . Jürgen Drews [2], et al. declare that in 2000 most of the drugs therapy has been based on 500 known molecular targets. The also state that "there are at least 10 times as many molecular targets that can be exploited for future drug therapy than are being used today." Jesse W.-H. Li, et al. work from 2009 claims at about 80% of the found drugs were of natural origin by 1900. But then slowly through the time the ratio of products that are based on natural compounds wall to the 50% and even 30% between 2001 and 2008. In their research they also identify two main sources of the difficulties in the drug discovery process.

- The prevailing paradigm for drug discovery in large companies.
- Technical limitations when identifying new compounds with desirable activity.

Hughes [3] describes the process of drug discovery in several phases 1.1.

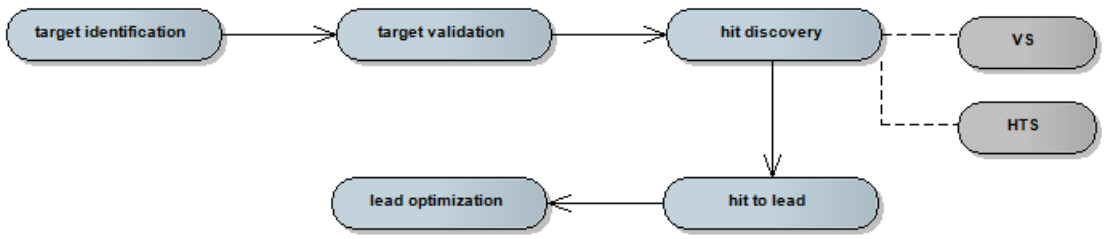



Figure 1.1:

Let's discuss some phases that are more interesting for us :

- *hit discovery* - This step involves testing of large amount of compounds (screening) for activity against chosen target.
- *hit to lead* - For each hit a library of similar small molecules is created. From this library the compounds with the best theoretical activity are used for further testing. The expert knowledge is commonly in use here. The selected compounds are called lead compounds.

Kim Sweeny [4] estimated the price of the aforementioned phases 1.1.

| | Cost US$m | Cost % | Time years |
|---|---|---|---|
| Biology | | | |
| Target Identification | 165 | 18.8 | 1.0 |
| Target Validation | 205 | 23.3 | 2.0 |
| Chemistry | | | |
| Screening | 40 | 4.5 | 0.4 |
| Optimisation | 120 | 13.6 | 2.7 |
| Development | | | |
| Preclinical | 90 | 10.2 | 1.6 |
| Clinical | 260 | 29.5 | 7.0 |
| Total | 880 | 100.0 | 14.7 |

Table 1.1: Drug Discovery and Development Process Boston Consulting Group, 2001

The phase of screening and optimisation form a significant part of the time and financial costs of the whole drug discovery process. Joseph A. DiMasi [5] estimated the price of 68 randomly selected drugs as "Capitalizing out-of-pocket costs to the point of marketing approval at a real discount rate of 11% yields a total pre-approval cost estimate of US$ 802 million (2000 dollars)."

## 1.2 Chemical informatics

The aim of the chemical informatics is to use computers and informational techniques in wide range of problems in the field of chemistry. Computational methods are called in silico methods as opposed to the in vivo (in the living organism) and in vitro (in artificial environment outside of the living organism) methods. QSAR, pharmacophores and other computational modelling approaches such as machine learning, data mining and neural network all those examples can be considered to be a methods that are used in chemical informatics as in silico methods. Use for computer methods can be found in capturing and processing large amounts of data. Those data then can be used in modelling to build models that can be used for making predictions of chemical compound interaction or it's structure features like for example binding sites. Bernard Testa [6] proposed a simple description of interactions as:

- Pharmacodynamics events what does the given compounds does to the biological system
- Pharmacokinetic events what does the system does to the given compound

This can be really important in process of drug discovering where the interaction between drug and human body are of crucial importance.

# 2. Libraries of chemical compounds

An important aspect not only in the drug discovery process is the source of chemical compounds. Specifically in drug discovery the libraries are sources of chemical compounds used for the screening. Therefore, the size and the quality of a library has a significant influence on the results. The researches may either utilize libraries of existing compounds or virtual libraries. The advantage of libraries of existing compounds consist mainly in the fact, that theses compounds are synthesizable. On the other hand, not all thinkable compounds have been synthesized yet. Another possibility is to utilize a virtual library. Virtual library contains computationally generated compounds (either pre-generated or generated on fly) and can thus offer much higher variety and amount of chemical compounds. However these compounds are not guaranteed to be synthesizable and their activity is not experimentally validated.

## 2.1 Libraries of existing compounds

Chemical libraries store not only the core information necessary for identification of a compound but also aim at accumulating as much knowledge about the chemical compounds as possible. The way how the libraries obtain data and how the data are processed and published make the main differences between the libraries. Here we give the overview of three main public data sources. It may be also interesting to mention that most of the them provide data access using the Resource Description Framework (RDF). Chemical libraries aim to provide information about biological activity of small molecules. The information about physicochemical molecular properties (molecular weight, number of rotatable bonds and calculated LogP) are also commonly provided. All presented databases also provide free web based access to the data. Tiejun Cheng [7] estimate the size of the chemical libraries 2.1.

### 2.1.1 Pubchem

Pubchem [8], the largest publicly available repository of small molecules, is maintained by the National Center for Biotechnology Information, which is part of the United States National Institutes of Health. Pubchem also provides a tool for uploading small molecules which can be used to upload new and update existing molecules in the database.

### 2.1.2 ChEMBL

ChEMB [9] [10] is a manually curated and maintained database of chemical compounds. The data are pooled primarily from published literature. Extracted data are examined for possible problems and normalized by given set of rules before loads database. As a manually curated database, ChEMBL provides more trustful data than, for example, Pubchem.

| Database | Type | No. of compoundsa | Website |
|---|---|---|---|
| PubChem | Public | 30 million | http://pubchem.ncbi.nlm.nih.gov |
| ChEMBL | Public | 1 million | https://www.ebi.ac.uk/chembldb/index.php |
| NCI Set | Public | 140,000 | http://dtp.nci.nih.gov/index.html |
| ChemSpider | Public | 26 million | http://www.chemspider.com |
| CoCoCo | Public | 7 million | http://cococo.unimore.it/tiki-index.php |
| TCM | Public | 32,000 | http://tcm.cmu.edu.tw |
| ZINC | Public | 13 million | http://zinc.docking.org |
| ChemBridge | Commercial | 700,000 | http://www.chembridge.com |
| Specs | Commercial | 240,000 | http://www.specs.net |
| Asinex | Commercial | 550,000 | http://www.asinex.com |
| Enamine | Commercial | 1.7 million | http://www.enamine.net |
| Maybridge | Commercial | 56,000 | http://www.maybridge.com |
| WOMBAT | Commercial | 263,000 | http://www.sunsetmolecular.com |
| ChemDiv | Commercial | 1.5 million | http://www.chemdiv.com |
| ChemNavigator | Commercial | 55.3 million | http://www.chemnavigator.com |

Table 2.1: Commonly Screened Chemical Databases, Source: Tiejun Cheng, 2012

### 2.1.3 Zinc

ZINC [11] [12] database differs from the previous too since it contains not only synthesizable compounds but also commercially-available compounds together with the information about where the compounds can be bought.. In order to support usage of ZINC for docking each molecule is accessible in 3D format so it can be readily used in docking programs.

## 2.2 Virtual libraries

Although the chemical libraries contain a large number of chemical compounds in comparison with the size of the chemical space their size is negligible. The chemical space is a space spanned by all possible chemical compounds and its size is virtually infinite. However, if we focus on compounds usable in drug discovery, we usually consider only compounds being subject to the Lipinsky's rule of five:

- log P $< 5$
- molecular weight $< 500$ g/mol
- $< 5$ hydrogen bond donors
- $< 10$ hydrogen bond acceptors

While these rules significantly restrict the chemical space size the size of the druggable subspace of the chemical space is still about $10^19$ molecules [13]. The way how we may tackle the disproportion between the sizes of chemical space and chemical libraries is to computationally generate new compounds leading to building virtual libraries. Basically, there are two main approaches. Both of them are based on generation of compounds obeying simple rules for chemical stability, synthetic feasibility, and medicinal chemistry.

- Neighbourhood generation focus on exhaustive molecule generation from neighbourhood of a given molecule. The neighbourhood is often defined by the number of possible structural changes like: atom addition, atom removal, bond addition, etc... Neighbourhood generation can be used in drug discovery in the phase of leads selection, where it can be utilized to generate a library from a single promising hit molecule.

- Path finding approach focuses rather on generation of chemical compounds relevant to given task under the hand. The goal is to identify a path in the chemical space between the input molecules [14]. The motivation comes from the fact that if two molecules share a target than molecules in their vicinity could do so too. Based on the utilized criteria, the exploration process can be guided by structural or physicochemical properties.

# 3. Virtual Screening

High-throughput screening (HTS) is a scientific method for evaluation of activity of multiple compounds against a given target. This method is mainly used in drug discovery in the first hits selection phase. As during this phase a large number of molecules needs to be tested, HTS is often an automatized process using robotics, advanced data processing, liquid handling devices, high-density microplates and sensitive detectors. Thanks to the combination of all these techniques, HTS offers the possibility to test tens of thousands of compounds in reasonable time. Some methods promise throughput about 100.000 compounds per second [15] by using the combinatorial libraries. It may be important to mention that as the throughput of screening increase it may also increase the possibility of an error.

Experimental screening has mainly two drawbacks. The first and main drawback is the necessity of having the test compound available. This can mean that for each experimental screen we may need hundreds of thousands or even millions of compounds. That would greatly increase the price of an experimental screen and moreover, to store such a large number of chemical compounds on stock could turn out to be impossible.

Tiejun Cheng [7] claims that the completion of the Human project has revealed a wealth of possible druggable targets. Kathrin Heikamp [16] goes even further and declare that "in the postgenome phase, there will not be the opportunity to experimentally screen everything". Virtual screening is basically in silicio version of HTS. Ekins [17] mention difference between HTS and virtual screening as: "In contrast to technology-driven HTS, virtual screening is a knowledge-driven approach that requires structural information either on bioactive ligands for the target of interest (ligand-based VS) or on the target itself (target-based VS)." It was the high cost and low hit rate of HTS that have simulated the development of computation alternative to HTS [7]. As we now have a lot of possible drug targets, we can expect increasing importance of virtual screening. The experiments using standard HTS will become time demanding and possibly even economically unfeasible. Klaus Pors [18] also mentioned that nowadays technologies provide much more than the in silico version of HTS. Todays methods can, for example, eliminate known substrates or inhibitors and so prevent future withdrawals of novel drugs in later drug discovery phases. The fact that virtual screening does not need real compounds can be an advantage as well as a disadvantage. The advantage consists in the fact that compounds do not have to be bought in order to test them. Virtual screening can be also easily used to perform screens over compounds that have been obtained from dynamic compound generation. This also gives rise to one of the disadvantage as the virtual screening may work with chemical compounds that can not be synthesized. Therefore, the synthesizability verification should be part of, or performed after, the virtual screening phase. However, much bigger disadvantage, when compared with biological screening, is its accuracy. As Tiejun Cheng [7] mentioned, the molecular targets are dynamic in real environment, which is often curtail for their biological functions. This increases the requirements on computational complexity and knowledge of the tested molecules' properties and behaviour. Unfortunately such information or computation power is not always accessible. This problem can be tackled by

various approaches. If we know at least the structure of the target we may screen large libraries and experimentally verify reaction of those molecules that show promising results in virtual screening. In this case we face the problem of virtual screening that it may produce false positives and false negatives. It is the incomplete information about the ligands/target, the complexity of reaction in real environment or the physicochemical laws that cause that we still do not have reliable method for virtual screening that would not generate false-positives and false-negatives. Brian K. Shoichet [19] presents an opinion that the "VS can be view from the same logic as HTS: as long as some interesting ligands are found, false-negatives are tolerated."

There exist multiple approaches to virtual screening each suitable for different setups [16]. In this thesis we will provide only a brief overview of the available approaches and methods for virtual screening.

### 3.0.1 Structure-based virtual screening (SBVS)

SBVS is also known as docking or target-based virtual screening [17]. SBVS requires detailed information about a target. Such information can be obtained either experimentally by solving given target or through the computational modelling. During the screening, every molecule in a given library is virtually docked into the target binding site using a docking program. With the resulting computed model of the target-ligand interaction we may then utilize a scoring function to evaluate the fitness value between the docked molecule and the target [7]. The docking program can be quite complex and during the docking simulation it can take into account factors like metal ions, water molecules and other environmental factors. However, the complexity can be quite computationally demanding. Therefore, the size of the compound library forms also a significant issue. Tested libraries often contain can contain from several thousands to millions compounds [7]. With such a huge libraries even the increasing performance of modern computers can not provide enough power to blindly dock every molecule in the library. For this reason it is a common practice to prefilter the libraries to be used for docking. The filters may include simple filters for physicochemical properties or filters employing similarity search with the goal to exclude compounds that may be significantly more computationally demanding for docking calculations than others. The similarity search can be also used to filter out molecules that are too similar and thus does not bring enough novelty. Or the filter can filter out structures that differ too much from the reference structure and thus there is a low chance that they will bind to the target. Although many SBVS tweaks have been proposed the single most grievous issue is the unreliable scoring procedure, that stems from the inability to consistently and correctly predict the free energy of protein-ligand binding [16]. As a result, the docking based evaluation often requires time-consuming human expert post processing of the result rankings.

### 3.0.2 Ligand-based virtual screening (LBVS)

The Ligand-based virtual screening (LBVS 3.0.2) is basically a process that ranks the molecules in given chemical library according to affinity with respect to a certain target. LBVS methods rely on the similarity-property principle which

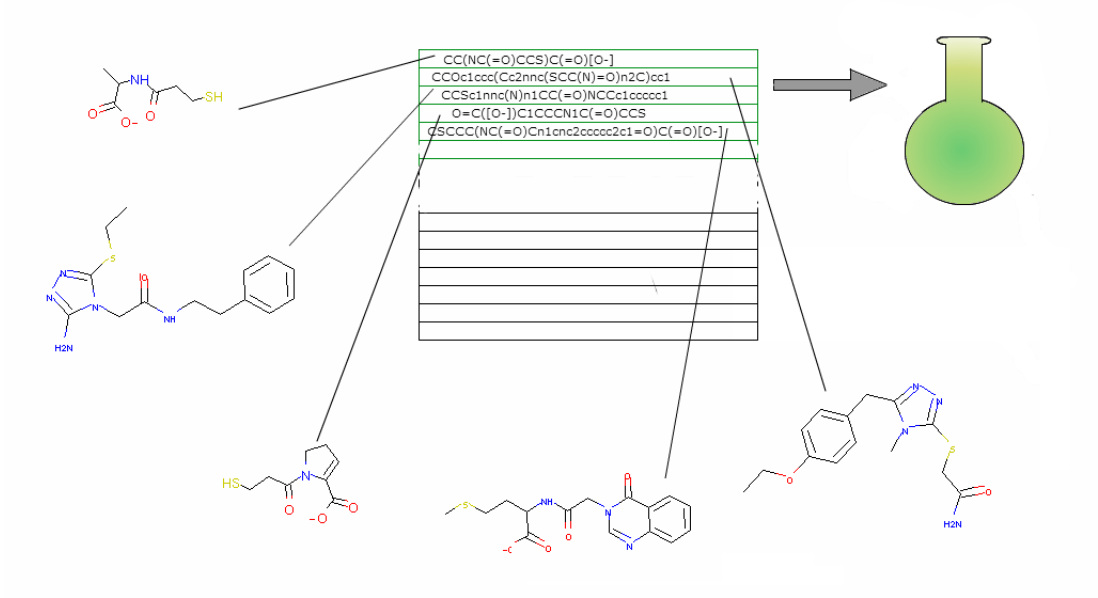



Figure 3.1:

states that similar molecules should exhibit similar properties. Therefore, LBSV assumes that similarity-based ranking of database compounds with respect to a target mirrors the probability of actually being active against given target. The methods mostly differ in the their computation demands and the type of information they use for the similarity scoring. The scoring is performed based on similarity to know actives (ligands) against given target, which is where the name comes from. Unlike SBVS, in LBVS the only required information is the one about known actives, not the target. Ekins [17] summarizes that the LBVS methods generally provide significant enrichments over the random selection of molecules in a database. The scientific problem in LBVS is the missing established reliable metric between the similarity and activity [20]. Different methods can provide different outcomes with uncertain result [21]. Kathrin Heikamp [16] notice that the LBSV provided compounds ranking in compare to SBVS does not provide an sufficient indicator of biological activity. This creates the need of a follow-up analysis of the LBVS results.

### Similarity-based

The similarity-based approach can be seen as a generalization of QSAR and pharmacophore-based approach. Those approaches works with a set of selected descriptors and similarity function that provides information how much are two molecules similar. With such function we may then calculate the similarities between known actives and the tested molecules and from this information create the ordering. In best case the ordering should represent the probability of activity. The fingerprints and byte-based similarity function are often used in this category.

### Quantitative structure-activity relationship(QSAR)

Quantitative structure-activity relationship (QSAR) and quantitative structure-property relationship (QSPR) are machine learning approaches. Both methods

in the learning phase construct a mathematical model (for example by using the regression analysis) based on the structure or physico-chemical information. The function can then be used to predict properties and activity of a molecule based on its descriptors. (Tiejun Cheng, 2012) consider QSAR/QSPR methods to have a great potential for modelling and design of novel compounds with robust properties. The belief comes stems from the QSAR/QSPR methods' ability to forecast physicochemical properties as a function of structural features.

### Pharmacophore-based virtual screening

A pharmacophore represents atoms and their mutual spatial arrangement in a molecule (chemical features) responsible for the drug's action. Common chemical features included in a pharmacophore are for example whether a position in a molecule is a hydrogen bond donor or acceptor. Pharmacophores that use the molecular structure basically try to represent a local arrangement of atoms which is responsible for certain molecular property or activity against a given target. A single pharmacophore will unlikely represent such complex information. On the other hand with the list of pharmacophores we may be able to catch what is needed. Pharmacophores basically provide a way how to describe rules for a molecular structure.

3D pharmacophores contain information about positions and relative orientations of chemical features on the level of a single atom. The positions of atoms are represented in the 3D space. Typically the 3D pharmacophores represents certain atom classes, such as hydrogen bond donors and acceptors, charged groups, aromatic rings. The obvious disadvantage of this approach is that we have to know a conformation of a molecule. It follows that in order for 3D pharmacophores to work we need to know the spatial information about every molecule we want to test. This makes use of the 3D pharmacophores difficult.

2D pharmacophores use a graph topological distance instead of euclidian distance. Tus, it does not have to know the 3D position of the atoms. Thus, 2D pharmacophores can be easily computed for a given chemical compound and therefore employed as pharmacophore fingerprints (PFP). In the PFP each bit states whether a given 2D pharmacophore is present or not. The alternative is that the fingerprint holds an information about how many times a certain pharmacophore is present in a molecule. We may then use desired distance function and calculate distance between two molecules. This allows to sort a list of molecules according to similarity to other molecule (or ensemble of molecules), which is all what we need to use pharmacophores as a method of LBVS.

# 4. Descriptors

Every chemical compound contains features/properties that can be captured and described. These captured information are called descriptors. A descriptor can hold various types of information (number, bit array, 3D function-map). There exist many different descriptors nowadays each suitable for a different purpose. For example, some descriptors may represent the physicochemical properties while others may be a result of a more complex computation. Therefore, the complexity of carried information may vary. In general, we can state that more complex descriptors are harder to compute but contain more information. For example the number of carbon atoms in a molecule does not convey much information about a molecule but it is fast to compute. By contrast, descriptors that are based on quantum mechanics may provide more accurate representation of properties, but their computation requires substantially more time. However, not all descriptors correspond to molecules physico-chemical characteristics. A common example is probably the most often utilized class of descriptors, the 2D fingerprints.

## 4.1 2D Fingerprints

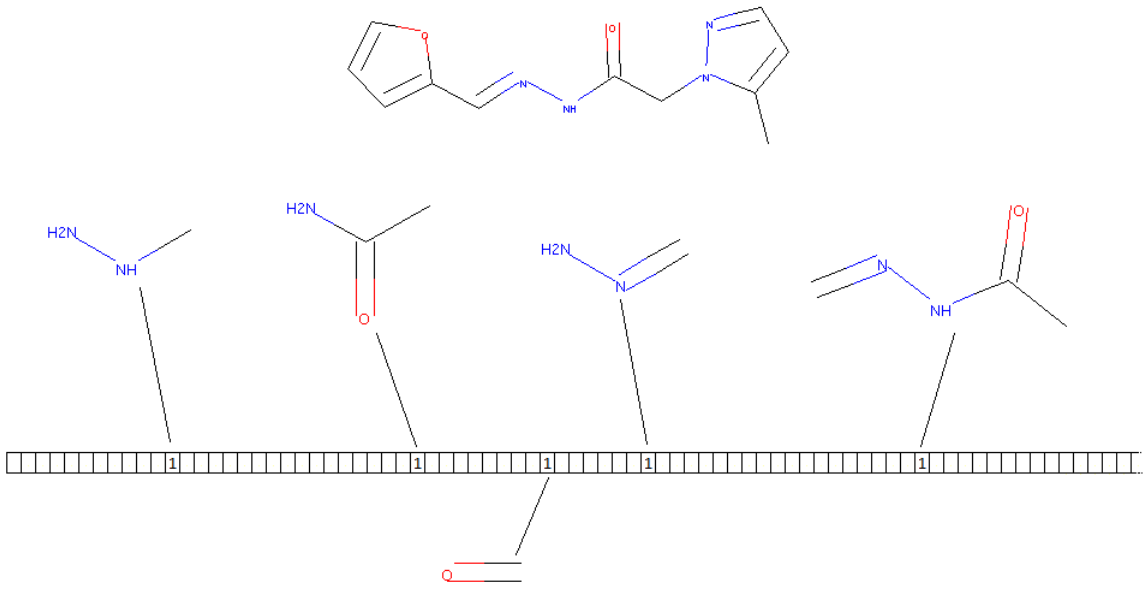



Figure 4.1:

The 2D fingerprint is a standard computationally efficient abstract representation of a molecular structure. Although the fingerprints were designed primary for efficient substructure searching in large chemical structure databases, they can be efficiently used also for similarity searching and thus for virtual screening. The fingerprints are often represented as bit strings which offer the possibility of a fast comparison. The main idea is to encode the existence of a given structural feature by a position in a bitstring 4.1. The existing fingerprints can be

categorized into three main classes:

- *Directory-based fingerprints* - The Directory-based fingerprints represent a chemical compound structure in terms of predefined substructures and their positions in a bitstring. If a molecular substructure present in a dictionary is also present in the chemical compound then the respective bit is set to one. The substructure-position mapping is 1:1. The disadvantage of directory-based fingerprints is the fixed size of the directory. This means only a subset of possible substructures can be captured.

- *Hashed fingerprints* - The hashed fingerprints are not dependent on a predefined directory. Instead they extract the substructures directly from the molecule (extracted structures depend on given fingerprint type). After the substructures are extracted they are labeled with a number which is subsequently hashed to a position in the bitstring. The number of bits in the bit string is usually much smaller than number of possible substructures. Thanks to the hashing the hashed fingerprints can encode any fragment in given chemical compound. It is often also not possible to map a bit position from fingerprint to a unique substructure and so it may not be easy to interpret the hashed fingerprints. Hashed fingerprints can be further divided into two subcategories:

  - Atom Pairs and Topological Torsions
  - Extended Connectivity Fingerprints

- *Pharmacophore based fingerprints* - These fingerprints may seems to stand on the edge between the directory and hashed based fingerprints. The pharmacophore fingerprints use a predefined directory of pharmacophores. In a given chemical compound all pharmacophores are identified and respective bit positions are turned on in the fingerprint. We can either store only information about the presence of a given pharmacophore or we can also store information about the number of how many times the pharmacophore is present in the molecule.

As the fingerprints were originally design for database similarity search they have direct correlation to the similarity of two chemical compounds. The main idea is, that if two chemical compounds are the same then their fingerprints are identical. If two fingerprints are different then the respective chemical compounds differ. If two fingerprints equal then it may happen that the respective molecules still differ, since the fingerprints are not designed to see such difference (fragment is not in a dictionary) or the information was lost in the representation (two fragments hashed to the same position in the bitstring).

### 4.1.1 Atom Pairs and Topological Torsions

Atom pair descriptors encode information about every pair of atoms in a compound. For each pair the information about the atoms and the length of the shortest past used [22]. Each atom is described by its type, number of non-hydrogen atoms to which it is bonded and its number of bonding electrons.

Kearsley [23] recognized that the properties of the atom may be more important than the specific atom type. A modified version utilizes this motivation as it uses seven groups to classify an atom: cations, anions, neutral hydrogen bond donors and acceptors, atoms which are both donor and acceptor, hydrophobic atoms and all others. While the AP fingerprints use atom pairs as subgraphs from which features are extracted, the Topological torsions (TT) fingerprints use paths of length four (quaternion). The information about types, non-hydrogen connections and number of pi-electrons is still used [24]. It's easy to use information about all presented pars(quaternion) in chemical compound and use them to build a fingerprint. We just map the pair (quaternion) into the bit string.

### 4.1.2 Extended Connectivity Fingerprints

In Extended Connectivity Fingerprints(ECFPs) and Functional Connectivity Fingerprints (FCFPs) an atom is described in terms of its neighbouring atoms up to a certain radius. Hert [25] shown that such descriptor can be effective in similarity searching applications. The extended connectivity of an atom is calculated using a modified version of the Morgan algorithm [26] where the atom code is combined with the codes of its neighbours to create the final atom description. The radius from which the neighbours are utilized is encoded in the fingerprint's name. For example, ECFP2 considers only the immediate neighbours; ECFP4 extends to the third-order neighbours, etc.

## 4.2 2D Fingerprints similarity

The common way how to compare two bit string (fingerprints) is to utilize a similarity or distance measures comparing the bit positions. Unlike similarity measures which return values from range ¡0,1¿, the distance methods return values between 0 and N where N is the max value usually dependent on the size of the bit string.

In this section we use the following notation. Let A and B be bit strings of the same lengths, then a, b denote the number of bits in A, B, c denotes the number of common bits in A and B, d is the number of bits turned on in A xor B. And bar is used for negation, i.e. $\bar{a}$ is number of bits off in the bit string A.

**Definition.** *Tanimoto:* $S_{AB} = \frac{c}{a+b-c}$

**Definition.** *Dice coefficient:* $S_{AB} = \frac{2c}{a+b}$

**Definition.** *Cosine similarity:* $S_{AB} = \frac{c}{\sqrt{ab}}$

**Definition.** *Russel:* $S_{AB} = \frac{c}{a}$

**Definition.** *Kulczynski:* $S_{AB} = \frac{c(a+b)}{2ab}$

**Definition.** *McConnaughey:* $S_{AB} = \frac{c(a+b)-ab}{ab}$

**Definition.** *Manhattan:* $S_{AB} = \frac{\bar{a}-d}{\bar{a}}$

**Definition.** *RogotGoldberg* $S_{AB} = \frac{c}{a+b} + \frac{\bar{a}-a-b-c}{2\bar{a}-a-b}$

# 5. Virtual screening benchmark

Benchmarking 5 of a virtual screening is performed by simulating the virtual screening process on labeled data and evaluating the result. Taking into account the topic of this theses, here we will focus on LBVS only. The inputs of the LBVS is the set of known actives and a chemical library to screen.

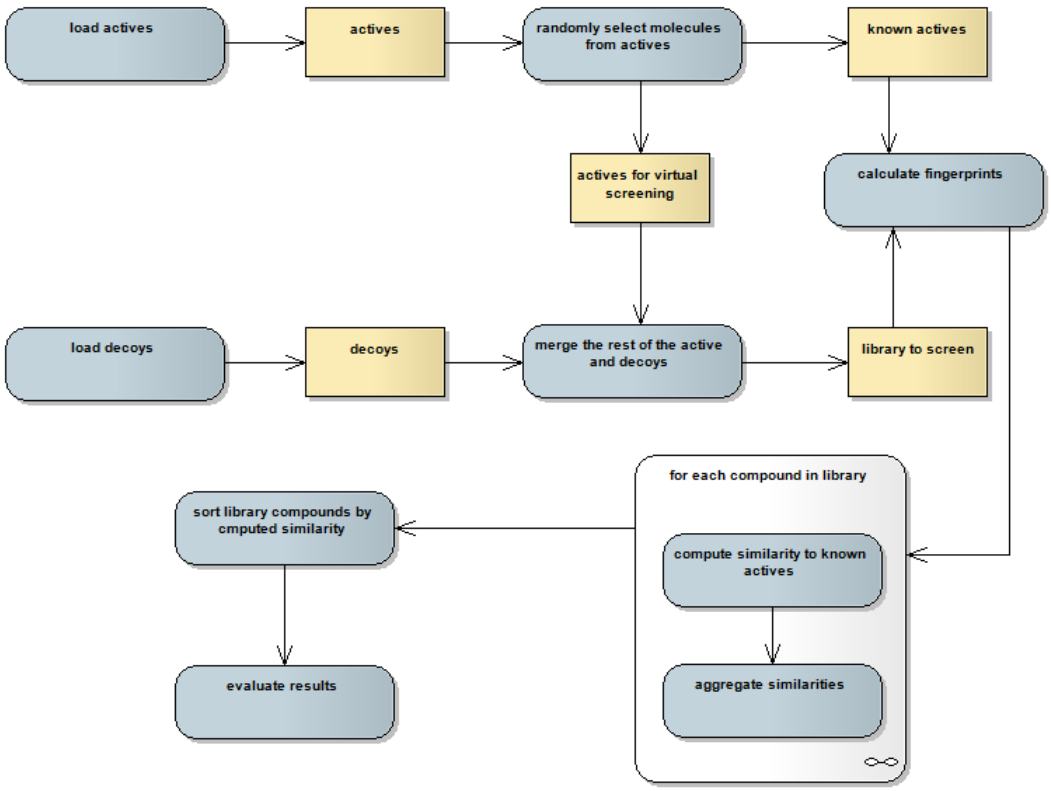



Figure 5.1: Process of virtual screening benchmark

In order to test a LBVS method we need a labeled dataset with known actives and inactives. Then we select the actives that will be used as known active ligands (query), the rest of the actives are pooled together with the decoys and the LBVS is carried out over this library. After the LBVS is done, the library compounds are sorted according to the similarity to the query molecules (known actives). If there are more than one query molecule the similarity scores are aggregated. The common approach for the aggregation is to use the highest similarity score as the similarity between a library compound and the query molecules. The sorted library is the result of the LBVS. Finally, the list with sorted compounds is scored using a LBVS quality measure.

The key part of this process is the selection of known actives which can greatly influence the benchmark results. A classical benchmark consist of multiple runs of LBVS where the actives are randomly selected, but the randomness can made the replication of such experiment hard as even for small number of actives there exists a number of combinations.

The problem with actives was a common problem related to dataset publications [27] [28] [29] [21] tackled this by problem by presenting the complex framework that use predefined random selections and so provide better reproducibility of the LBVS benchmarks.

## 5.1 VS quality measures

There are number of different fingerprints and similarity methods. So in case of LBVS there naturally emerge the need of selection of particular fingerprint and similarity method. Different fingerprints may perform differently based on task under the hand [21]

Fingerprints' performance is usually tested in benchmarking studies using datasets with known active and believed or known inactives (decoys). Such testing is essential not only for the developers but also for the end-user, as it may help them to decide which approach suits most their needs.

### 5.1.1 Receiver Operating Characteristic (ROC) curve

ROC is defined as the ratio of true positives to the ratio of false positive. ROC is mostly presented as a graph where where the x axis represents the ratio of false positives and the y axis the respective ratio of true positives. In context of virtual screening the number of true positive is equal to the number of actives at given rank divided by the total number of actives while the number of false negatives is equal to number of decoys at given rank divided by the total number of inactives. The closer the ROC gets to the left top corner of the graph the better. The graph diagonal, ie. $x = y$ represents the random selection. As the comparison of two graphs may not be easy, an area under the curve (AUC) was established to aggregate the information into a single number.

**Definition.** $AUC = \frac{1}{n*N} \sum_{i=2}^{N} A_i(I_i - I_{i-1})$

Where A is the comulative count of actives to rank i and I comulative count of inactives to rank i. The AUC is nonparametric and is bounded to 0,1. The AUC = 0.5 corresponds to random selection.

The AUC summarizes the information from ROC curves but since it is an aggregation it also loses much of the information. The same AUC value may represent different ROC curves. While the shape of the ROC curve may be vitally important for a comparison of virtual screening methods, as often only a small proportion of the dataset, which has been virtually screened, is used for further testing. The insufficiency of AUC sensitivity was also criticised by Jean-Francüois Truchon [30]. He gave the following example to illustrate the problem of early recognition "For instance, sample repositories of pharmaceutical companies commonly contain more than 1 million compounds. Identifying a set of a few thousand compounds to screen experimentally requires that the VS method perform well within the first 0.1% of the scored compounds! Even if the VS method is excellent at retrieving all actives for any target within the first half of the dataset, it is useless if the early performance is bad.".

The advantage of the AUC in comparison to some other methods is that it is non parametric. So it can be easily used to give a basic idea about the performance of tested method.

### 5.1.2 Enrichment Factor (EF)

The enrichment factor is popular for its simplicity and easy understanding. The EF reflects the enrichment in given fraction of data for actives in comparison to a random selection. The EF is parametric method, the parameter determines the considered fraction of data and is often given as a percentage. Unfortunately the 1 percent from small dataset can be enough while 1 percent from huge dataset can be too much.

**Definition.** $\delta(r_i) = \begin{cases} 0 & r_i > \chi^N \\ 1 & otherwise \end{cases}$

**Definition.** $EF(\chi) = \frac{\sum_{i=1}^{n} \delta(r_i)}{\chi^n}$

The ER is defined as the number of found actives in a given data fraction divided by the expected number of actives.

With proper parameter the EF can effectively measure the early recognition and thus give a valuable information about the measured method. The EF disadvantage is that it is dependent on the ratio of actives and inactives in dataset. This fact decrease the reliability of comparison of two methods without that additional information. There exist other approaches that tries to eliminate this disadvantage.

### 5.1.3 Robust Initial Enhancement (RIE)

Robert P. Sheridan [31] identified the problem of ER as follow "However, if the number of actives is small, this measure is not very robust because moving one or two compounds across the arbitrary boundary makes a large difference in IE", in the same article he also proposes the RIE that should overcome those problems.

**Definition.** $S = \sum_{i}^{actives} exp(\frac{-rank(i)}{a})$

**Definition.** $RIE = \frac{S}{<S>}$

Where $< S >$ represents the S computed for average random selection. In original article [31] propose usage of mean from 1000 random experiments The RIE is parametrized by the argument $a$, which should be significantly smaller then $N$ (total number of tested chemical componds) . In some literature the parameter $a$ is defined as $a = \frac{alpha}{N}$. Where the *alpha* is then parameter of the RIE evaluation and basically influence how sensitive the methods is to the new actives.

As can be seen from the definition the RIE = 1 represents the random selection.The meaning of the RIE is similar to the EF as it also compare the given performance with the random one. Jean-Franciois Truchon [30] computed the exact value of the random selections and propose the following formula:

**Definition.** $RIE(\alpha) = \frac{N}{n} \frac{\sum_{i=1}^{n} exp(\frac{-\alpha * r_i}{N})}{\frac{1-exp(-\alpha)}{exp(\frac{\alpha}{N})-1}}$

As can be seen this formula is parametrized by $n$, $N$ and $a$. RIE is unbounded and so based on the dataset there are different min and max values. This feature prevents from intuitive handling of the RIA value as to direct comparison of RIA on different datasets.

### 5.1.4 Boltzmann-Enhanced Discrimination of ROC (BEDROC)

Jean-Francüois Truchon [30] in his work proposed BEDROC as a method for measuring quality of. The BEDROC aims to solve the problem of RIE, which is that RIE is unbounded. The bounding to the $< 0, 1 >$ interval is achieved by linear transformation.

**Definition.** $RIE_{mix}(\alpha) = \frac{N}{n} \frac{1-exp(\frac{\alpha n}{N})}{1-exp(\alpha)}$

**Definition.** $RIE_{max}(\alpha) = \frac{N}{n} \frac{1-exp(-\frac{\alpha n}{N})}{1-exp(-\alpha)}$

**Definition.** $BEDROC(\alpha) = \frac{RIE(\alpha)-RIE_{min}(\alpha)}{RIE_{max}(\alpha)-RIE_{min}(\alpha)}$

Jean-Francüois Truchon interpret BEDROC as follows: "BEDROC metric is the probability that an active ranked by the evaluated method will be found before a compound that would come from a hypothetical exponential PDF with parameter R", where PDF stands for probability distribution function.

## 5.2 Datasets for benchmarking

Another criterion influencing the result of a benchmark, even more than the evaluation method, is the choice of the dataset. In fact the used dataset may have a great influence on the tested fingerprint performance. For example, if the dataset contains actives and decoys which differ significantly in 'simple' properties (like molecular weight, number of hydrogen bond donors/acceptors), then this fact may lead to a good performance of even a relatively poor method since the method can utilize the differences in the simple properties [32].

In the same study, a similar effect has been reported for docking benchmarking. In this case also the difference in simple property (size) between actives and decoys cause the docking enrichment to appear to be artificially good. This problem can be tackled by removing the disproportion of chemical properties between actives and decoys - in other words that data should be unbiased in as many properties as possible.

Another problem was also the lack of freely accessible standardized datasets that would enable comparison of benchmarks from different articles, as each article uses different dataset for benchmarking. Recently presented datasets(DUD 5.2.1, MUV 5.2.2, ChEMBL 5.2.3) try to solve this problem by providing freely accessible datasets for benchmarking.

### 5.2.1 DUD

Database of useful decoys(DUD) has been created by Huang [27] and is freely available. The DUD was created as benchmark datasets for structure-based virtual screening (docking). DUD contains 2950 annotated ligands for 40 targets. For each ligand it also contains 36 decoys chosen from ZINC. Every decoy has similar physical properties but dissimilar chemical structures with respect to its active counterpart. The targets were selected in such a way that they offer a reliable view of how the docking might perform on typical and interesting targets. The receiver operator characteristic (ROC) curves were used during the evaluation in order to avoid biases introduced in enrichments plot when the ratio of actives to decoys grows. The authors find that "there is probably no such thing as a perfect single decoy set for testing docking algorithms against all targets, but there certainly are better and worse ones, and the former offer better protection against artifactual performance to the unwary docker".

### 5.2.2 MUV

Maximum Unbiased Validation (MUV) [29] is a collection of datasets that were designed primary for benchmarks of LBVS, but can be as well used for SBVS methods. Each of the MUV datasets contains 15 actives, selected from PubChem, and 15000 decoys. In order to increase reliability of these datasets, activity of all used actives were required to be determined by low-throughput dose-response experiments. The goal of the MUV design is to generate sets with a spatially random distribution of actives and decoys in a simple descriptor space. The simple descriptors authors define as "vectorized form of the respective counts of all atoms, heavy atoms, boron, bromine, carbon, chlorine, fluorine, iodine, nitrogen, oxygen, phosphorus, and sulfur atoms in each molecule as well as the number of H-bond acceptors, H-bond donors, the logP, the number of chiral centers, and the number of ring systems". The author obtain a spatially random distribution by: 1) First the datasets of actives are adjusted to a common level of spread. 2) Decoy sets are selected with a common level of separation from the actives. These measures make the dataset hard for methods that use a simple descriptor space.

### 5.2.3 ChEMBL

A large-scale similarity search has been carried out on 266 compound activity classes extracted from the ChEMBL database by Kathrin Heikamp [28]. Two two-dimensional (2D) fingerprints were used for this analysis, MACCS structural keys [33] and the extended connectivity fingerprint refECFP with bond diameter four (ECFP4). The fingerprints were chosen to represent the opposite ends of the performance spectrum of 2D fingerprints. The results indicated that many activity classes can be well treated using 2D fingerprints, despite their relative simplicity. The set of 50 human targets extracted from ChEMLB for purpose of benchmarking of virtual screening were also presented in this work.

## 5.3 Benchmarking platform

Sereina Riniker [21]introduced a framework for benchmarking of LBVS approaches. The framework is written in Python 5.3.3 and uses RDKit 5.3.4 as the underlying chemical framework. It comes with a predefined set of fingerprints, similarity methods (Dice, Tanimoto, Cosine, Russel, Kulczynski, McConnaughey, Manhattan, RogotGoldberg) and quality measurement methods (ROC 5.1.1, EF 5.1.2, RIE 5.1.3, BEDROC 5.1.4). The framework simulates LBVS on three datasets DUD 5.2.1, ChEMBL 5.2.3, MUV 5.2.2, representing 88 targets in total. For each target a set of known actives is also available. As the framework aims to increase reproducibility of experiments it also contains a predefined random selection of actives and decoys. Thanks to that, the simulation of LBVS is deterministic.

### 5.3.1 Framework structure

The benchmark process in the Sereina Riniker's framework [21] basically consists of three core parts. Each part is represented by a directory with the respective python script.

- *Scoring* As the input this part loads a chemical compounds dataset, the data about the selected actives (query list), the list of fingerprints to test, the name of the similarity method to use and how many actives should be chosen as a query (options are 5, 10, 20). Using this information the LBVS is executed. For each dataset 50 iterations are done while in each iteration the predefined subset of actives is used as a query (known actives). The rest of the actives are pooled with inactives to form the library to screen. For each molecule in the library the fingerprints and similarities (distances) to all query compounds are computed. Only the highest similarity is used (MAX fusion method) as the final score for given compound. The result of each iteration is saved into a file (one file per dataset). For each tested chemical compound (from the screened library) the similarity, id and information if its active (true if the chemical compound is considered active, false otherwise) is stored.

- *Validation* In the validation phase, the file resulting file of the scoring phase is loaded as a list. The list is sorted according to the similarity and the performance is evaluated by using multiple quality measure functions. The employed quality measure functions can be selected by the user. In this phase the results are still kept separated for each of the 50 iterations. So we can get performance for every iteration and we also know the actives that has been selected as targets. This information could be utilized for detail analysation of LBVS performance on in every repetition.

- *Analysis* The last part consists in aggregation of the obtained results. The average performance for each method is calculated over each of the datasets. It's also possible to compute the performance for specific fingerprints or evaluation methods.

### 5.3.2 Results

Sereina Riniker [21] presented not only the platform for LBVS benchmarking but also compared number of existing FPs, similarity and quality measurement methods. Those experiments were done on the above described benchmarking platform. The results showed that there one can see differentiate easy (DHFR, $ER_agonist$, GR) and difficult (12911) targets (datasets). Moreover, the MUV proved to be the hardest ot the three tested datasets. For the hard targets, the results of some methods was even close to a random method. The benchmark leads to the following conclusions about fingerprints "Except for the baseline fingerprints, the performance of all fingerprints is generally similar". The correlation between the evaluation methods was also found. Based on that, the usage of AUC 5.1.1 and EF 5.1.2 or BEDROC 5.1.4 is encouraged for benchmarking evaluation.

### 5.3.3 Python

Python [34] is a dynamic, typed and interpreted language. A once written code can be easily used on multiple operating systems, where the python interpreter is installed. This can be valuable for scientific experiment since they can be easily reproduced on different machines and operating systems.

### 5.3.4 RDKit

In our experiments we decide to use RDKit [35] , an open-source computational chemistry platform. RDKit can be used with python as well with C++. Since RDKit provides a lot of easy-to-use to use tools, it is simple enough to prototype and develop an cheminformatics application. From the list of RDKit features we mainly utilize loading a molecule from the SMILE format and fragment analysis. Computation of many different descriptors is also offered by RDKit but we decide to utilize another tool for this task because it provides a wider scale of available descriptors.

# 6. Vector Fingerprints

In this work we introduce vector fingerprints (VectorFPs), a new approach to the representation of chemical compounds and their comparison. VectorFPs utilize structure fragments to represent a chemical compound. The fragments are represented in terms of their physicochemical properties. Therefore unlike standard representations, with VectorFP a compound is represented by physicochemical properties of individual fragments. VectorFP is basically an array, where each cell represents one or more (in case of a collision) fragments.

Computation of VectorFP of a molecule consist of four main steps:

- 1. Extract fragments from the given chemical compound and compute indexes for those fragments.
- 2. For each fragment compute its physicochemical descriptors.
- 3. Convert each fragment's descriptor into a bitstring. The bitstring is used as a fragment representation.
- 4. Create a fragment representation from the descriptor representations.
- 5. Combine the individual fragments representations (bitstrings) to assemble the representation of the chemical compound. The representations are stored at positions computed in step 1.

Extract fragments from the given chemical compound and compute indexes for those fragments. For each fragment compute its physicochemical descriptors. Convert each fragment's descriptor into a bitstring. The bitstring is used as a fragment representation. Create a fragment representation from the descriptor representations. Combine the individual fragments representations (bitstrings) to assemble the representation of the chemical compound. The representations are stored at positions computed in step 1.

As the VectorFP size is limited the index computed in step 1 must be modified by application of the modulo operation (hashing). As a consequence collision may occur. A collision means that two or more fragments are mapped to the same position, ie. the hashed indexes have the same value. VectorFP utilizes the bitwise logical or to merge representations of multiple fragments together.

## 6.1 Representation

In order to maintain compatibility with existing fingerprint methods we decided to choose the bit string as the final representation for VectorFP. The Advantage is that we can utilize existing well-established similarity measures and benchmarking platforms in order to get comparison of our method to the other fingerprints.

## 6.2 Molecular fragments

Every chemical compound can be expressed by a 2D chemical formula. From the chemical formula, a graph representation of the chemical compound can be

constructed (edges represent bonds, vertices represents atoms). A continuous subgraph then represents a molecular fragment. Since the subgraph is continuous, a fragment is also a chemical compound. There exist different types of molecular fragments. The following list enumerates several types commonly used in contemporary fingerprint representations:

- Atom pair fingerprints 4.1.1 work with pairs of atoms. The atoms do not have to be connected.But still we can view such pair as a fragment.
- Topological torsion (TT) fingerprints 4.1.1 define paths in a chemical compound's graph. In TT the path length is four.
- Extended connectivity fingerprints (ECFP) 4.1.2 utilize a modified Morgan algorithm [26] for fragment extraction. In ECFP, the fragments are neighborhoods [1] of given atom. The size of used neighbourhood can be set, so the fragments of different sizes can be obtained.
- Pharmacophore fingerprints are build on a directory of pharmacophore definitions. Each pharmacophore represents a (believed)significant chemical features and their relative position in chemical compound. This representation directly use only part of the information of the fragment.

After the fragments are extracted they are further processed based on the fingerprint type. The resulting fragment representation always carries less information with respect to the original information. There are multiple places where the information can be lost. In some cases (pharmacophores) the information is lost during the extraction of the fragment. Some information is always lost when a fragment is being encoded into a bit string as the whole fragment is represented just by a single bit (position). Also multiple different fragments can be represented by the same bit. There exists a simple way how we can prevent the loss of the information in the encoding phase. We can simply utilize a list as a storage. Thus there would be no collisions and we can compare the fragments directly. This could be a suitable modification for similarity search, however the problem here is the speed that would be lost by such representation. Lossless representation of the structure information may not be effective for LBVS at all. The problem of such precise representation is lack of generalization ability, ie. ability to identify chemical compounds that are structurally similar but not the same. In LBVS we are trying to find chemical compounds that have potential to react with a certain target. If the LBVS identifies only actives that are structurally highly similar to known actives, then such result may be not optimal for some situations. For example in drug discovery, if the newly identified molecules are too similar to known actives they can be dropped as they do not bring enough new structure information.

## 6.3 Fragments representation

Some fingerprint approaches (TT, ECFP) directly convert the extracted fragments into a position in a bit string. In pharmacophores fingerprint approach the

[1] All subgraphs that contain given atom and all atoms that are reachable with the given number of steps (neighbourhood size)

fragments are represented in terms of pharmacophore features. For each pharmacophore feature there is a respective position in the fingerprint. As multiple fragments can suite the same feature the information about the number of feature occurrences is lost. So the representation is more about whether a feature is present at all rather than how many times and where.

We can view differences between TT, ECFP, .. and pharmacophore fingerprints in terms of used fragment representation. While the TT, ECFP work with the fragment, the pharmacophore fingerprints convert the fragments into pharmacophore features and then work with the pharmacophore features. The difference between those two approaches is the one extra step when the information about a single fragment is generalized -conversion from fragment into pharmacophore.. Another representation is used in QSAR or even in some types of pharmacophores. Here, the physicochemical descriptors are utilized to represent fragments or whole molecules. This kind of representation in fact can have some additional value against approaches that works only with fragments themselves (no information is derived). The descriptors extract and provide explicit access to information that is implicitly encoded in the chemical structure. Our goal is to combine in one fingerprint structure-based representation with the physicochemical properties-based representation as both have shown good results in some LBVS setups.

## 6.4 Descriptors conversion

We may classify the physicochemical descriptors into two categories based on their value types as 1) integers 2) floats. It may seem that the conversion of integers into bit strings is simple and straightforward, but it presents similar challenges as conversion of floats into bit strings. We first introduce two approaches to the conversion and then we discuss the respective problems.

- *In memory bit representation of a number.* This approach copies the representation of a number in memory into the fingerprint bit string. Such approach may not be suitable for floats as in floats representation some bits have different meaning than others. As a consequence, the number of differing bits may not properly reflect the difference between two floats.

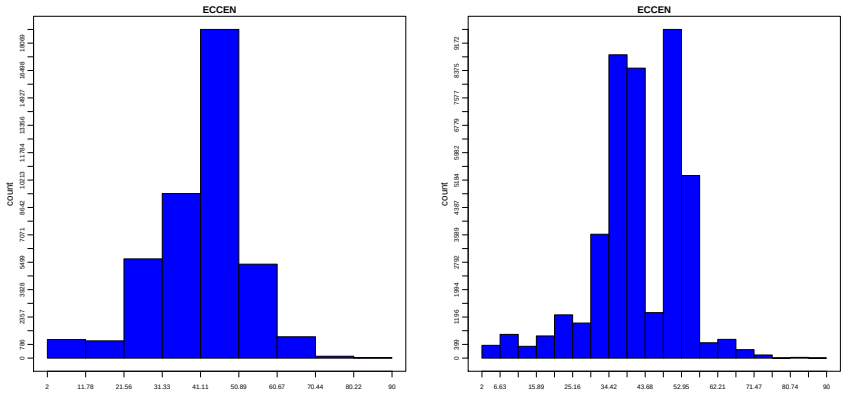
- *Binning.* By binning we understand the formation of a set of disjoint intervals (bins) that represent the possible values. A unique combination of bits is then assigned to each of the interval representing the resulting descriptor value. A simple example can be that the value of an integer determines the number of ones in the bit string.

Both approaches suffer from the not well bounded values of physicochemical descriptors. For many physicochemical descriptors we may try to estimate the min and max values or use representation where the values outside out interval will be intentionally represented by the same bit string. We can examine the data and find out the min and max safely, the possible problem is that some outliers may increase the needed range significantly and so we may allocate unnecessary amount of bits. We should try to keep the size of final fingerprints bit string

reasonably small. The overestimation may cost bits that could be used to represent different physicochemical descriptor, also the total fingerprint size can be a problem as the descriptors allocate space in segments and the final representation (fingerprint) of the whole chemical compound may consists from hundreds to thousands of segments. Typically we have not apriori knowledge which descriptors are useful and which not, same as we do not know which values are important. This problem is much more obvious if the binning is utilized. In case of binning we have to decide where the ranges of the bins will be.

The proper choosing of ranges may greatly improve the results. If we can select right ranges, we may save space (positions in bit string) and achieve good representation. In case of overestimation ie. choosing too many bins, it's possible that some of them would not be used at all. If there is lack of ranges or they are improperly placed we may loose possibly important information about the fragment. It may look like that more bins is better then less,unfortunately even with a lot of bins it's possible to miss the important values and aggregate data into bins in a bad way. This can lead to loosing of much more information than with smaller number of bins, but with better range selection. On the other hand proper aggregation (binning) may support the generalization ability and increase the performance. The aggregation can be viewed as intentional information loss that can support generalisation of fingerprint.

To demonstrate the importance of proper binning we select the ECCEN descriptor. The first histogram 6.1(a) use 10 ranges while the second 6.1(b) 20. As can be seen the middle of the graph change dramatically if more bins are used. While on the first graph the values between 40 and 50 form a single group, on the second graph we can see there are in fact at least two groups 39-43 and 48-53. This is generally the issue of range selection for binning, when the level of detail (number of bins) and ranges must be chosen.




## 6.5 Collisions handling

There are two types of collisions:

- *Intra-collisions* happens inside the fingerprint, where two fragment hashed on the same position in a bit string.

- *Inter-collisions* are between two fingerprints, when a position is turned on in both bit strings.

### 6.5.1 Intra-collisions

The collisions of the first type are generally not wanted. It means that two fragments share the same position. The result of such a collision is that some information is lost or that the information about at least two fragments must fit into space designed for a single fragment. In case of collision we use logical 'or' to merge the data from the conflicting fragments. The advantage of this solution is the simplicity and consistency. If two chemical compounds contain two conflicting fragments then the result is always the same irrespective on the order in which the fragments are extracted. This stems from the fact that bitwise 'or' is a commutative and associative operation. Different solution than using logical 'or' could be to use only the first fragment, but if the ordering of fragments is not given, we can generate two different fingerprints for the same molecule. In order to not require ordering of fragments we decide to not use this approach now and leave it for further testing.

### 6.5.2 Inter-collisions

In case of a simple binary representation the inter-collision is not a real collision. It means that two fragments that were hashed on the same position are present in both fingerprints. This fact is utilized by most similarity methods so every method that is designed to be used with such similarity methods must respect that. The following list discusses possible collision types, their meaning and how they are handled in VectorFP. Let us have fragments F1 and F2 from compounds C1 and C2 being has at the same position:

- If F1 = F2 then the collision simply states that the C1 and C2 contain a common fragment. In this case VectorFP behaves like classical fingerprints.
- If $C1 <> C2$ and C2 contains other fragment (F2') that takes the same position as F2, then we compare F1 to F2, F2'. In classical hashed fingerprints (ECFP, TT, ..) there is no difference in this situation and situation from the first case. As presence of fragments F2, F2' is represented by single bit, there is no space to represent that there are two fragments hashed to the same position. If the representation is more complex than the result depends on the representation and merge strategy (ie. how intra-collisions are handled). VectorFP has no optimization for this case, which leave us space for further improvements.
- If $C1 <> C2$ and there was no intra-collisions type in neither of the respective fingerprints and also $F1 <> F2$. In classical binary approach the fragments will be seamed to be same ie. F1 = F2, which is not true. In VectorFP however, we work with physicochemical descriptors and we store fragment using this descriptors. So in this case we compare the descriptors - basically compare the fragments in more detail. This is important as the

fragments on same positions may not be the same. Thanks to the comparison on fragment's descriptor level we believe, that we are able to provide more accurate information about the similarity of whole molecules.

## 6.6 VectorFP calculation

Based on our analysis of fingerprint construction we design a process 6.6 that should capture all significant phases and enables us to modify their specifics.

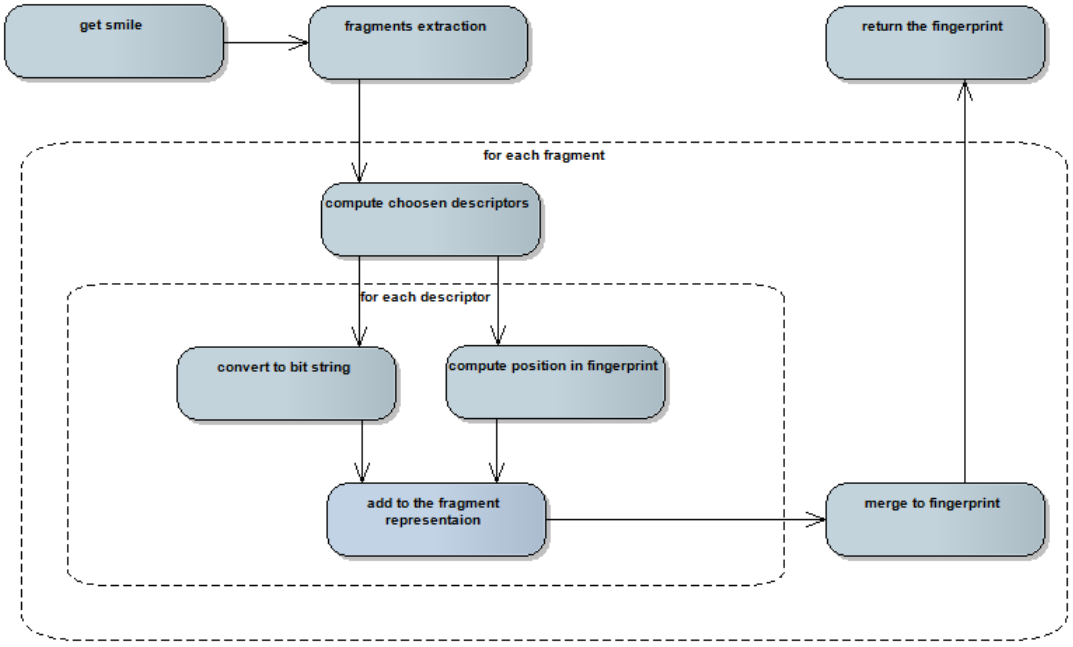



Figure 6.1:

- 1) Fragments extraction
  The fragments are extracted based on chosen method, in our case we utilize the RDKit algorithm which is based on Morgan (Morgan) algorithm. But any method that extracts fragments can be utilized.
- 2) For each fragment
  - a) Descriptor computation
    Chosen descriptors are calculated.
  - b) For each descriptor
    - I. Convert the descriptor into a bit string
      We may decide to use different methods and parameterizations to convert different descriptors based on types (integer, float) or distribution information. The number of bits available for each descriptor can also be chosen.

- c) Assemble fragment representation for bit strings
  In this phase we just copy the bit strings from previous phase one after another into the fragment representation.
- d) Compute fragment position in fingerprint
  In our implementation we utilize the existing method that comes with the RDKit, and provide position in atom (root) and index (position in fingerprint) for each fragment. The index is an integer so before its use we have to fit it into the fragment size.
- e) Merge(save) bitstring representation of fragment to the fingerprint
  If no fragment has been stored at the given position we just copy the bit string (fragment representation from step 2.c) into the fingerprint on given position
  * I. Handle the collisions
    In our case we ignore collisions and just use the logical 'or' to to merge the results. If two chemical compounds have the same fragments then they still get the same fingerprints.

# 7. Vector fingerprint - implementation

VectorFP is implemented in python and utilizes RDKit as the chemical framework. We decided to use those two technologies as the benchmarking platform we utilize also use them. Thus, we can easily plug our method in and evaluate it. During the implementation we tried to keep the code modular so the modules of code can be replaced and the whole functionality modified. Special attention was given to the conversion of descriptors into bit strings and this part is easily configurable and extendable.

## 7.1 Architecture

There are two main packages - descriptors and fingerprints. The descriptors package contains functionality for computing physicochemical descriptors. The package fingerprints contains functionality for decomposing a chemical compound into fragments and for calculation VectorFP from fragments and descriptors. Not all the methods are located in classes therefore a class diagram can not fully present the functionality. But still it may be used to deliver the general idea about code architecture. The code is split into multiple files. Each file provides different VectorFP functionality.

- descriptors.cache.Cache.py
  We utilize PaDEL 7.2 for descriptor computation. PaDEL is standalone java program for descriptor computation. As invoking an PaDEL as an external process is slow, package descriptor.cache contains class Cache that can be used as a cache for descriptors so once a descriptor given fragment is computed it can be reused in the future if needed. We can also preload some data into Cache and prevent invocation of PaDEL at runtime. Such optimization can have impact on performance and is highly recommended.

- descriptors.source.PaPED.py
  The Padel class is an implementation of the DescriptorSource interface. PaDEL is utilized as an external process in order to compute the value of required descriptor.

- descriptors.source.Source.py
  The DescriptorSource is an interface for a service that can provide value of given name to given fragment (represented as a SMILES string).

- fingerprints.Convertors.py
  The methods for converting descriptors into bit strings are located here.

- fingerprints.MoleculeRep.py
  This file contains classes that carry the representation of chemical compounds as a set of fragments. The functions that build the representation are also located in this file.

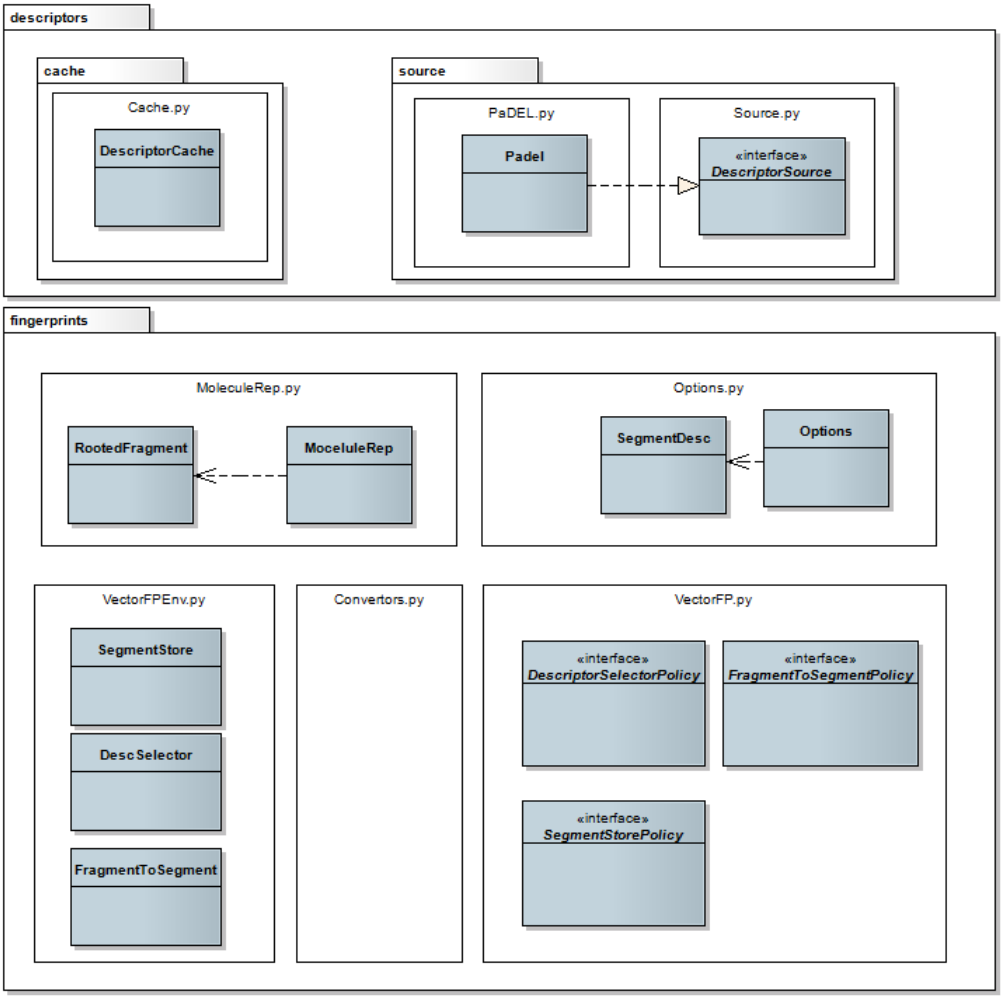



Figure 7.1:

- fingerprints.VectorFP.py
  The method for computing the VectorFP is located here. It can be parametrized by several options (DescriptorSelectorPolicy, FragmentToSegmentPolicy, SegmentStorePolicy).

- fingerprints.Options.py
  Contains classes that can be used to configure the VectorFP computation process, especially the descriptors that should be used and the way in which they should be converted into the bit strings.

- fingerprints.VectorFpEnv.py
  Provides the default implementation of VectorFP settings. The implementation uses settings from OPTIONS. The OPTIONS is a global variable in VectorFPEnv file that configures the way the fingerprints are computed. It is also the main file that should be imported into a project that would use the VectorFP.

## 7.2 Descriptor source

In order to deliver as many descriptors as possible we decide to utilize the PaDEL [36] descriptor generator. PaDEL is a free Java tool, that computes various descriptors (2D, 3D, fingerprints) for given molecule. In comparison with RDKit, PaDEL offers more 2D descriptors. The disadvantage is that PaDEL has to be executed as an external process. This causes a serious performance problem. As a solution we decided to cache the descriptors. But even with the use of a cache PaDEL can be executed too often in a standard virtual screening procedure, so we decided to completely precompute the cache. This step however requires knowledge of all chemical compounds in the library to be screened since for each of them VectorFP need to be computed. Unfortunately the used benchmarking platform only asks for fingerprints which it needs at the moment, and does not provide a initialization step. We tackle this problem by extracting all chemical compounds in the benchmarking datasets and precomputing the fragments and the respective descriptors. A file that contains this informations is then loaded into our fingerprint's cache so the external execution of PaDEL is not needed anymore. The list of computed descriptors can be obtained directly from the header of the PaDEL output file or is located in PaDEL.py python file. This list is valid if the PaDEL is executed with same options as are used in PaDEL.py.

## 7.3 Descriptor to bit string conversion

The methods for conversion of numerical descriptors into bit strings are located in Convertors. py file. Currently, 3 conversion method are available.

- intCopy
  Copy the representation of an integer. If used for float, the conversion to integer is performed first.
- intUnaryCoding
  If the number is float then it is converted to an integer. The descriptor's value determines how many bits will be turned on.
- intIntoBins
  Can be used for integers and float without conversion. The user must define used ranges (bins).

All three methods also require length of the resulting bit string as a parameter.

## 7.4 Sample usage and configuration

The VectorFP implementation aims to be easily usable, but still some settings have to be done before first usage. The following list contains possible options.

- PaDel.PADEL
  Path to the PaDEL.
- PaDel.TEMP_DIR
  Path where temporary files can be stored.

- VectorFPEnv.USE_CACHE
  If true then data from cache file are loaded, when the VectorFPEnv is imported.

- VectorFPEnv.CACHE_FILE
  Path to the cache file.

If those properties are set, then the following example illustrate the usage of VectorFP.

```
# create mol representation of given smile
from rdkit import Chem
mol = Chem.MolFromSmiles(smile)

from cz.skodape.chem.fingerprints import Options, VectorFPEnv
// set VectorFP
VectorFPEnv.OPTIONS = Options(1024,
        [SegmentDesc("nHeavyAtom" , 8, Convertors.intCopy)])
// calculate the fingerprint
fp = VectorFPEnv.calculateFpAsBitVec(mol)
```

In this example only nHeavyAtom descriptor is used to describe fragment. This descriptors use 8 bits, the conversion is done by copying the 8 lower bits of representation. The fingerprints has 1024 segments, and $1024 * 8 = 8192$ bits.

# 8. Experiments

VectorFP is highly parameterizable and its performance is heavily dependent on the chosen parametrization. Moreover, what is an optimal parameterization is dependent on the actual dataset (actives, decoys). Selection of optimal parametrization can be viewed as a problem of feature selection, which is a different problem and it's solution is not part of this thesis. The experiments run over 88 datasets, where each dataset represents a different target. For each dataset the VS has been repeated 50 times. We decided to use 5 actives from each dataset as known actives. To evaluate the results we use AUC and BEDROC(0.20). There are two possible ways of VectorFP performance evaluation:

- Comparison with other existing fingerprints. Here we face the problem with parametrization as in order to get the best from the VectorFP we should first investigate the optimal parametrization for the dataset.
- Comparison of different VectorFP's parametrization. In this case we can also compare to utilized underlying fingerprint. The VectorFP utilize third party algorithm (Morgan, RDKit) to generate fragments and positions in fingerprint. As the base fingerprint (BASE), we will mark fingerprint that use the same fragment extraction algorithm and just set ones based on fragment's position.

The main difference between first and second case is, that in first case we focus on best performance of VectorFP, while in the second case we research influence of parameterization on the performance.

## 8.1 Descriptor selection

In order to descriptor values we create a fragment set. The fragment set consist fragments from all chemical compounds from all data sets. PaDEL generates about 771 2D descriptors, that can be easily utilized in VectorFP. Neitherless to say that about 3/7 of the descriptors are useless for our purpose. A brief examination found that only 407 descriptors are not constant on the fragment set. Also the distribution of the descriptors' value vary significantly. Based on this finding a certain level of preprocessing should be introduced to the process of descriptor selection. Selection of constant descriptor does not improve the performance of the VectorFP, also the bits allocated for given descriptor could just consume memory. Some types of descriptors are safe: molecular weight, number of hydrogen donor-acceptor bonds, count descriptors. In most cases those descriptors will probably not be constant, the size of the chemical compound or presence of certain molecular feature (sulphur, oxygen, ..) will also probably be important for the chemical functionality.

The count descriptors represent number of occurrence of certain molecular feature, like acidic group, hydrogen, oxygen, etc. The possible advantage of this descriptors is integral values and relatively small ranges. Values of such descriptors often equal a single value, mostly zero. This behaviour is acceptable, however the problem may arise if the descriptor is loaded too much. For example

in the nAcid descriptor, about 96,7% of fragments have zero value. This does not leave much space for other values, and basically divide all fragments into just few categories (practically 3 for nAcid).

As we illustrated on the previous lines the proper descriptor selection (feature selection) is hard problem, which is not the part of this thesis. During the research we try to utilize few descriptors based on our previous analysis. Those parametrization does not prove to be better then the one used in the experiment section in this thesis. The reason could probably be, that our analysis focus more on overall descriptor usability (if they are constant, if they have more than two values ..) then on value distribution in terms of active and decoys.

At the end we decide to use following descriptors:

- *nHeavyAtom* - represents the number of heavy atoms ie. all atoms except hydrogens. This basically represent the size of the fragment.
- *nHBDon* - represents the number of hydrogen donor bonds.
- *nHBAcc* - represents the number of hydrogen acceptor bonds.
- *nAromBond* - represents the number of bonds with aromatic character.
- *nRotB* - represents the number of rotatable bonds. This could be viewed as the potential of fragment to change the structure.

We have chosen them as we believe that they represent an easily imaginable chemical features, have a reasonable value distribution 8.1 and perform reasonably well. The used descriptors could be seen as simple descriptors. They prove to be useful since on some datasets they separate well the the space formed by the active and decoy molecules.

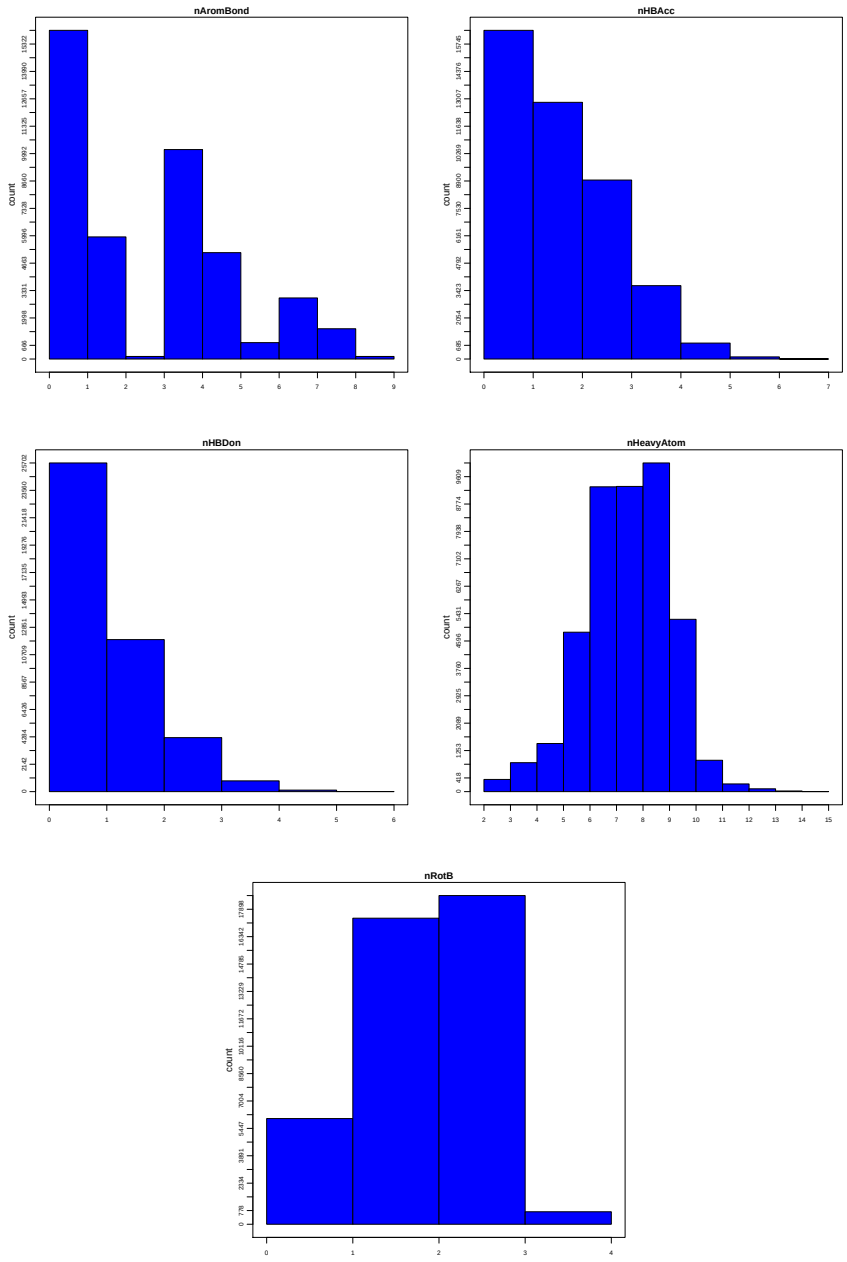
The MUV dataset has been optimized against simple descriptors which include count of heavy atoms, sulfur, etc. as well as the number of hydrogen donor, acceptors, chiral centers. This means that the active and decoys are unbiased in the space formed by simple descriptors. So with our descriptor selection we should not expect good performance over the MUV datasets.

| descriptor name | min | max | var |
|---|---|---|---|
| nAromBond | 0 | 9.0 | 6.402 |
| nHeavyAtom | 2 | 15.0 | 2.626 |
| nHBAcc | 0 | 7.0 | 1.518 |
| nHBDon | 0 | 6.0 | 0.999 |
| nRotB | 0 | 4.0 | 0.683 |

Table 8.1: Summary for selected descriptors

## 8.2 Experiment design

The experiments were designed rather to compare performance of different descriptors conversion methods than to achieve the best possible results. We focus on comparison of different conversion methods on basic descriptors. All the experiments have utilized the AUC and BEDROC as a performance metric. The




primary used metric in comparison was the AUC. The BEDROC is provided in appendix in form of a summary table for each experiment.

To give the general overview we use tables with summary information containing:

- mean - AUC mean
- max - max reached AUC
- min - min AUC
- #best - number of datasets on which given methods gives the best performance
- improvement - sum over AUC differences between given method and first

method in a table. If positive then the current method performs better than the first one.

improvement - sum over AUC differences between given method and first method in a table. If positive then the current method performs better than the first one.

## 8.3 Descriptor conversion comparison

In the first series of experiments we focus on comparison of different descriptor conversion methods for selected descriptors and general descriptor performance:

- nHeavyAtom
- nHBDon
- nHBAcc
- nRotB

In order to get comparison of different conversion methods, we decide to use only one descriptor in each parametrization. As each parametrization (fingerprint) utilize exactly one descriptor we use the descriptor name to denote the fingerprint. We also provide performance of the BASE fingerprint to give comparison with the non modified version.

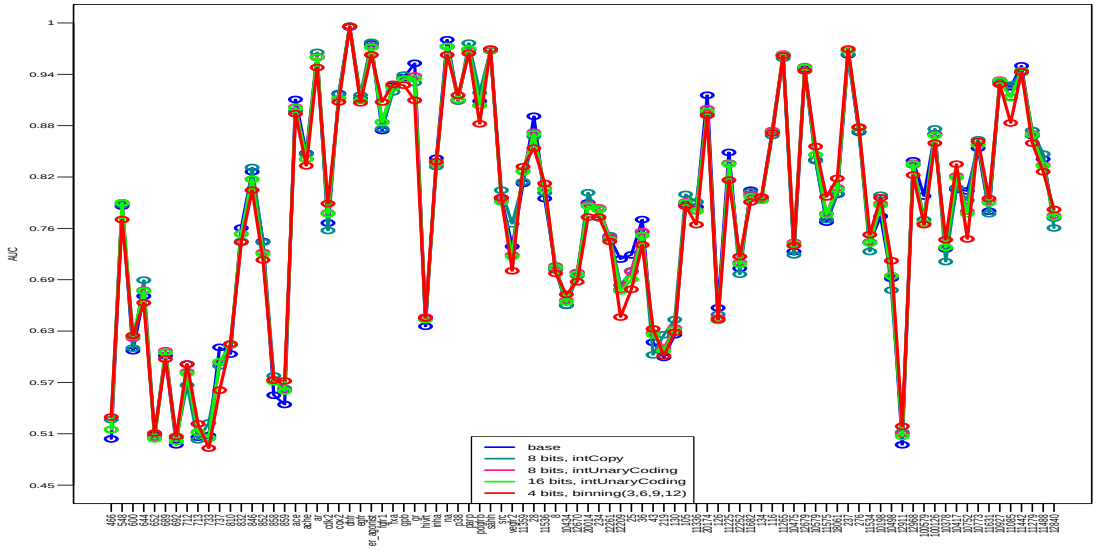



Figure 8.1: AUC of nHeavyAtom

As the results 8.3 8.3 8.3 8.3 9 9 9 show, the BASE fingerprint in average outperforms all parametrization. Interesting fact is, that the BASE fingerprints do not outperform our fingerprints by the same amount on all the datasets (MUV, DUD, ChEMBL), those performance differences are not negligible.

We decide for further analyzation of nHeavyAtom's performance over different datasets. The nHeavyAtom is in general 8.3 9 outperformed in terms of AUC (improvement). But on MUV 8.3 the BASE is outperformed in AUC from

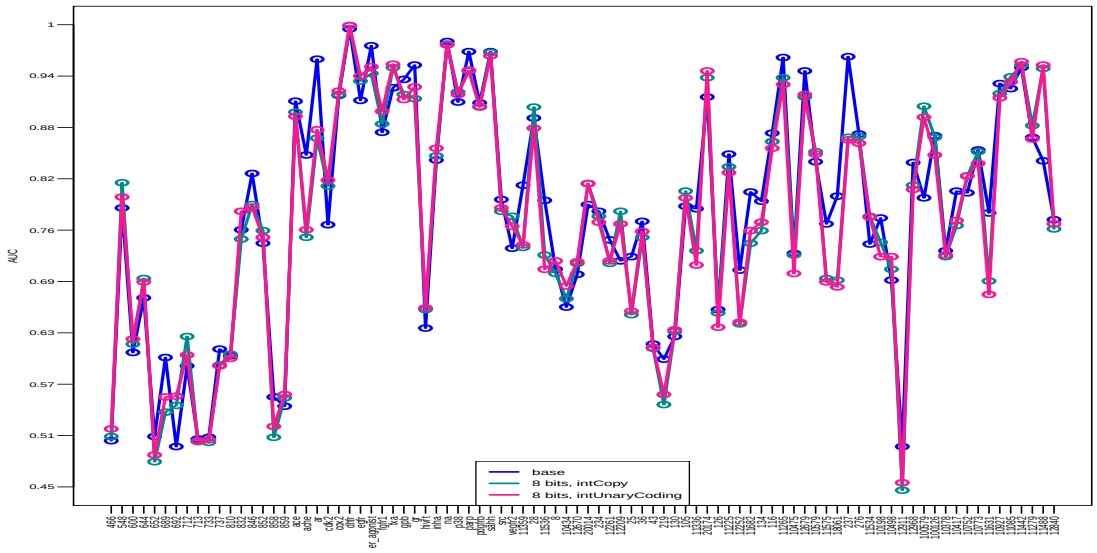



Figure 8.2: AUC of nHBAcc

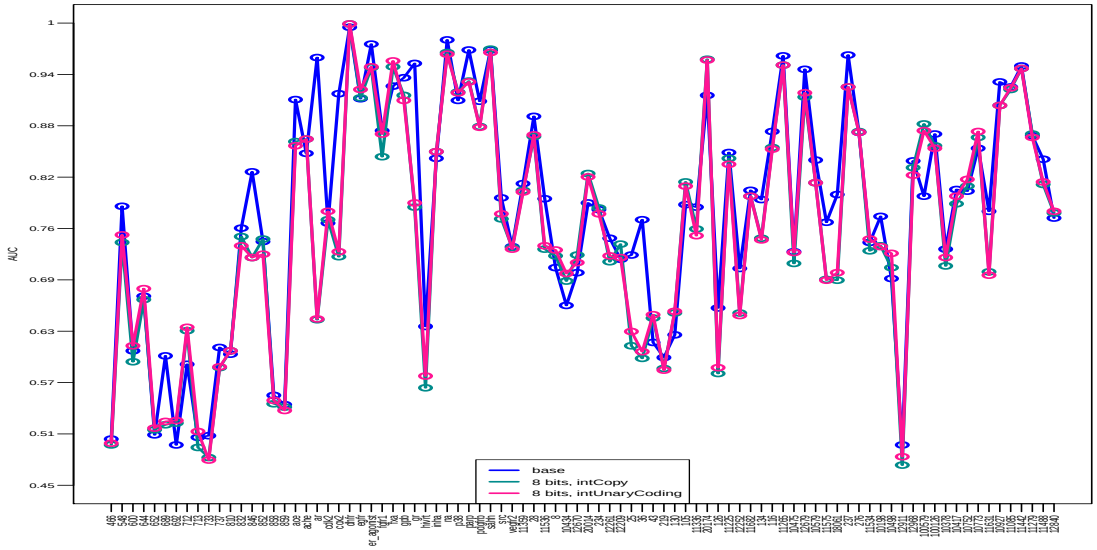



Figure 8.3: AUC of nHBDon

nHeavyAtom. On the DUD 8.3 and ChEMBL 8.3 dataset the BASE generally outperformed parametrization in AUC.

Tested parametrization perform best on MUV dataset, DUD and ChEMBL show up to be harder. This is unexpected as the MUV dataset has been designed "against" simple descriptors, that we use in our experiment. This observation also confirm the importance of benchmarking dataset selection. As the selection of dataset can have significant impact on the VS benchmark results.

On the nHeavyAtom we test the different conversion methods:

- 8 bits, intCopy (*8b, copy*)
- 8 bits, intUnaryCoding (*8b, unary*)
- 16 bits, intUnaryCoding (*16b, unary*)

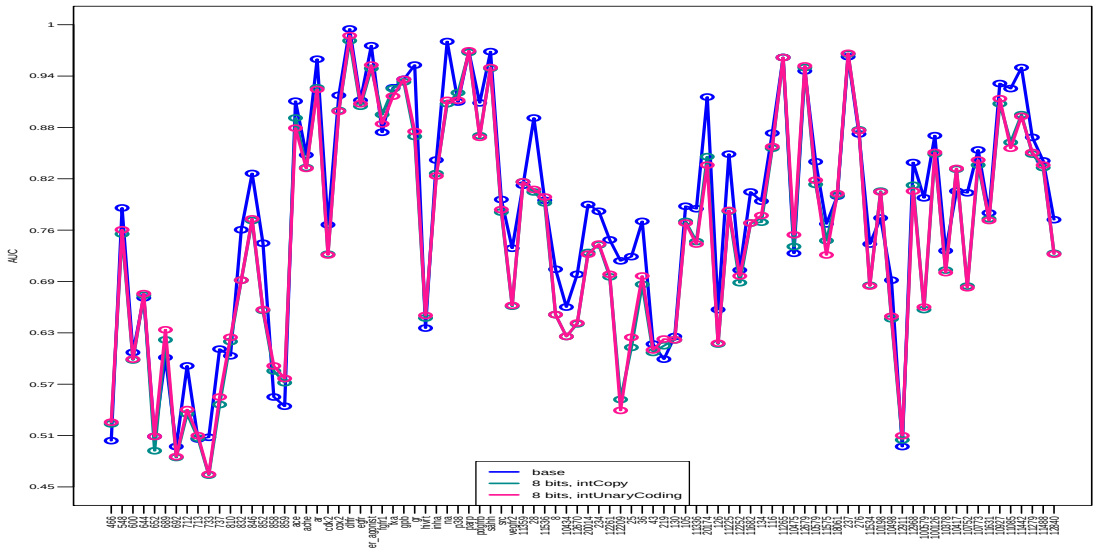



Figure 8.4: AUC of nRotB

| | BASE | 8b, copy | 8b, unary | 16b, unary | binning |
|---|---|---|---|---|---|
| mean | 0.7786 | 0.7779 | 0.7782 | 0.777 | 0.7744 |
| max | 0.995 | 0.996 | 0.996 | 0.996 | 0.996 |
| min | 0.498 | 0.504 | 0.502 | 0.502 | 0.494 |
| #best | 23 | 25 | 13 | 2 | 32 |
| improvement | 0 | -0.059 | -0.032 | -0.137 | -0.367 |

Table 8.2: AUC of nHeavyAtom in compare to BASE on MUV, DUD and ChEMBL

- 4 bits, binning(3,6,9,12) (*binning*)
  The binning parameterization represents binning with ranges $(-inf, 3), < 3, 6), < 9, 12), < 12, +inf)$

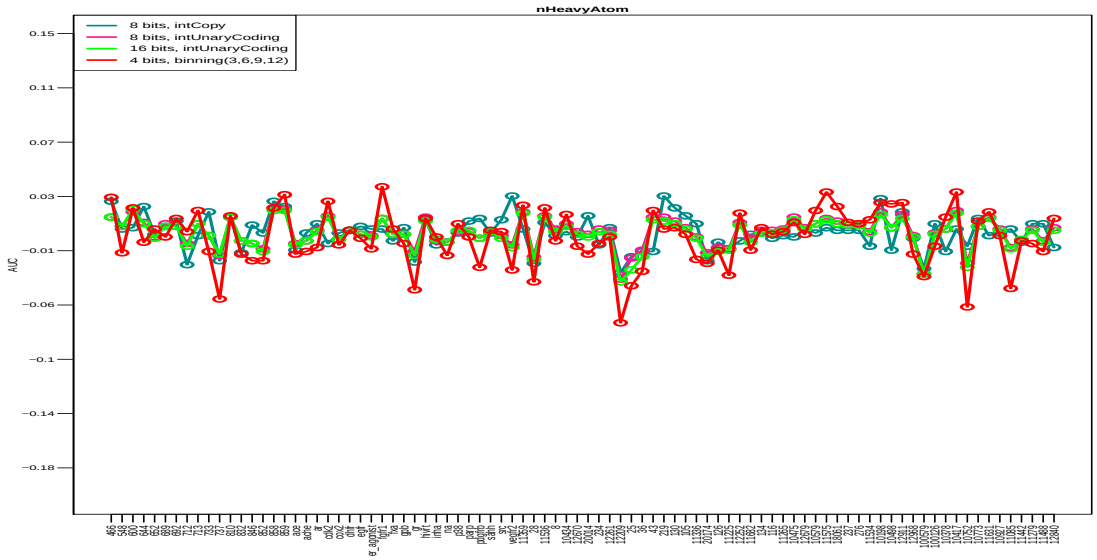



Figure 8.5: Difference in AUC of nHeavyAtom to BASE

| | BASE | 8b, copy | 8b, unary | 16b, unary | binning |
|---|---|---|---|---|---|
| mean | 0.89 | 0.8903 | 0.8879 | 0.8874 | 0.8838 |
| max | 0.995 | 0.996 | 0.996 | 0.996 | 0.996 |
| min | 0.639 | 0.648 | 0.65 | 0.647 | 0.649 |
| #best | 6 | 9 | 2 | 1 | 6 |
| improvement | 0 | 0.006 | -0.044 | -0.055 | -0.13 |

Table 8.3: AUC of nHeavyAtom in compare to BASE on DUD

| | BASE | 8b, copy | 8b, unary | 16b, unary | binning |
|---|---|---|---|---|---|
| mean | 0.7877 | 0.7852 | 0.7873 | 0.7855 | 0.7835 |
| max | 0.962 | 0.963 | 0.967 | 0.967 | 0.969 |
| min | 0.498 | 0.513 | 0.511 | 0.508 | 0.52 |
| #best | 13 | 12 | 9 | 0 | 18 |
| improvement | 0 | -0.128 | -0.024 | -0.11 | -0.212 |

Table 8.4: AUC of nHeavyAtom in compare to BASE on ChEMBL

The nHeavyAtom were more suitable for those experiment then other descriptors as the nHeavyAtom have greater range of possible values. In case of nHBDon, nHBAcc, nAromBond and nRotB the values are in range $< 0, 9 >$. If the 16 bits, intUnaryCoding would be used on nHBDon, nHBAcc, nAromBond or nRotB, the result will not differ from 8bits, as the higher bits are not used. So the overestimation on max value in case of intUnaryCoding do not influence the final performance.

As can be seen from results 8.3 8.3 the 8 bits, intUnaryCoding perform best of the tested methods. The chosen value of 8 maybe performs better as it could aggregate all bigger (8+ heavy atoms) molecules into a single group. But to state this hypothesis to be true, we would need to closer examine the datasets, which is not a part of this thesis. The worst AUS performance is recorded for binning approach, but there are two important things to notice. First, it use only 4 bins, which is only half of the space that is used by 8bit, intUnaryCoding. Second, the performance of binning fully depends on chosen ranges. In fact the intUnaryCoding can be also view as a special case of binning. On the other hand in terms of number of best results the binnig get highest score.

The difference between average AUC and the number of best results show up to be common during our experiments. So for that reason we decide to provide information about both. In authors' opinion it's not possibly to generally declare which of them is more important.

## 8.4 Two-descriptor parametrization

In these experiments we decide to test performance of parametrizations that utilize two descriptors. Against expectations, adding two descriptors together does not bring expected improvement 8.4 8.4 9. We design four parametrization for these tests. The first three consist of combination of the descriptors from the first experiment except the nHeavyAtom.

- nHBAcc, nRotB (*HR*)

| | BASE | 8b, copy | 8b, unary | 16b, unary | binning |
|---|---|---|---|---|---|
| mean | 0.6139 | 0.6176 | 0.616 | 0.6155 | 0.6124 |
| max | 0.823 | 0.828 | 0.814 | 0.814 | 0.801 |
| min | 0.498 | 0.504 | 0.502 | 0.502 | 0.494 |
| #best | 4 | 4 | 2 | 1 | 8 |
| improvement | 0 | 0.063 | 0.036 | 0.028 | -0.025 |

Table 8.5: AUC of nHeavyAtom in compare to BASE on MUV

- nAromBond, nRotB ($AR$)
- nHBAcc, nAromBond ($HA$)
- nHeavyAtom, $ETA_dAlpha_A$ ($HE$)

The only conversion method used is 8bits, intUnaryCoding. In this experiment we also compare the results with BASE fingerprints.

In general, all methods are outperformed by BASE fingerprint. But again as in previous experiments the results vary based on the data set. Tested parametrization again perform best on MUV data set. Interesting performance difference between HR and AR, HA could be observed on some targets (gr, 12209,100579). Although all the tested methods follow the same trends (get worse performance on this datasets) the difference between HR and AR is more than 0.15 AUC. For example on 100579 dataset the HR perform better than BASE method, while other two methods perform significantly worse. Those results show how important the descriptor selection is.

There are also significant performance differences between parameterizations. As the HR perform better then AR and HA, there could arise a question: if the nAromBond description does not influence the performance in a bad way. On the other hand the performance difference between AR and HA is quite large, so there must be other factor then just nAromBond.

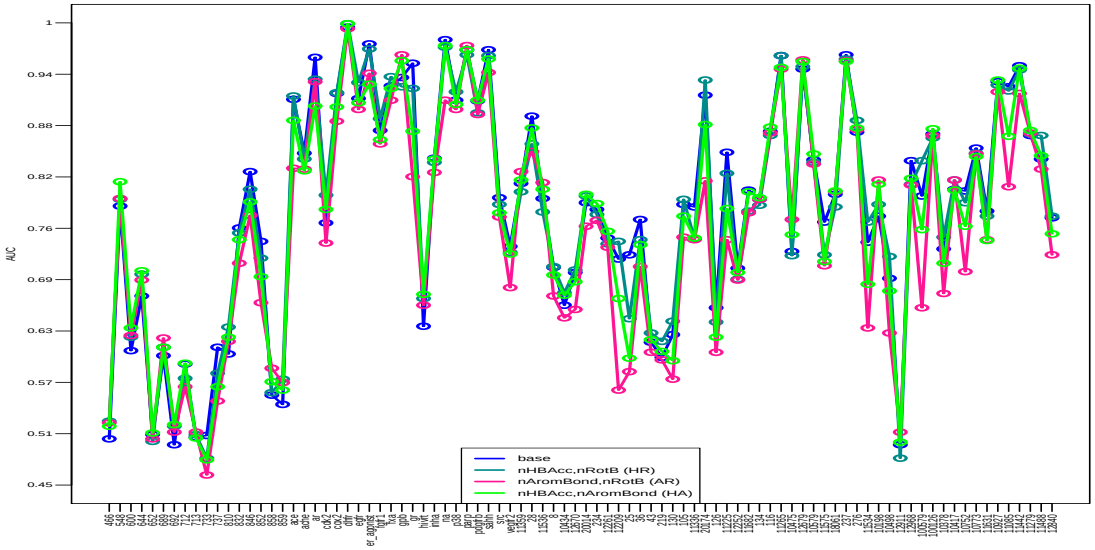



Figure 8.6: AUC of HR, AR and HA

| | BASE | HR | AR | HA |
|---|---|---|---|---|
| mean | 0.7786 | 0.7771 | 0.752 | 0.7685 |
| max | 0.995 | 0.999 | 0.993 | 0.999 |
| min | 0.498 | 0.482 | 0.462 | 0.48 |
| #best | 31 | 29 | 12 | 19 |
| improvement | 0 | -0.126 | -2.34 | -0.886 |

Table 8.6: AUC of HR, AR, HA in compare to BASE

In order to investigate the influence of single descriptor on whole parametrization we compare performance of HR and AR with performance of used descriptors. In order to present result of this comparison we decide to use diff graph, where the HR (AR) is selected as a base method. The graphs then shows difference of given method to the base method. As can be seen from the graphs 8.4 8.4 8.4 9 8.4 9

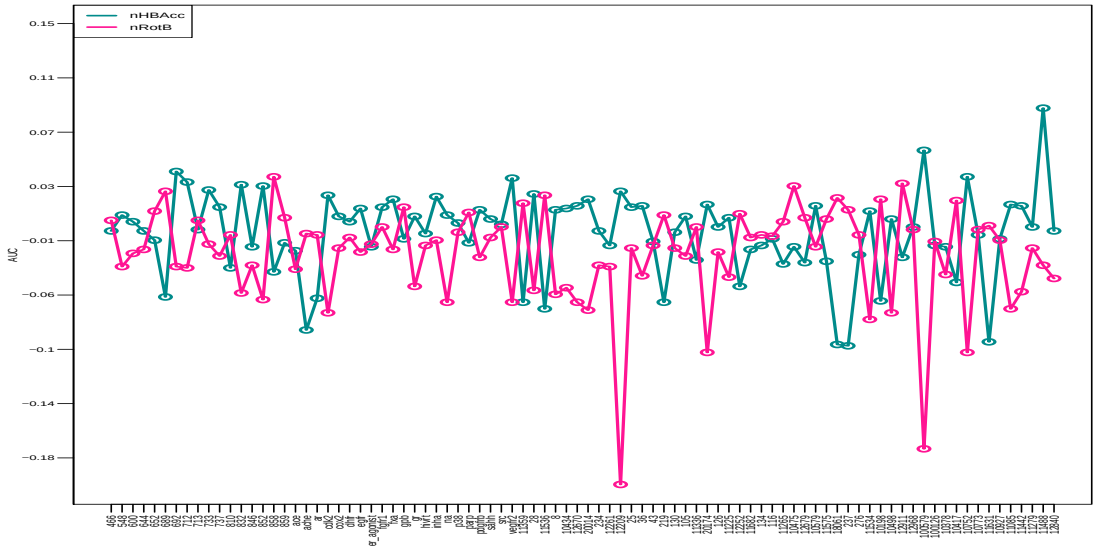



Figure 8.7: Difference in AUC of nHBAcc and nRotB to HR

there are significant performance differences of single descriptor parametrization on different datasets. Before we continue with the discuttion of the results, we have to make a note to the performance of nAromBond. The possible poor performance of nAromBond on some data sets (ache, cdk2, fxa, gr, inha, p38, sahh) could be caused by bad distribution of values in those datasets. Still results on those datasets can be usefull. They demonstrate the even one of the descriptor perform poorly, the other can still ensure reasonable performance of the whole parametrization. Similar situation can be observed on some datasets (12209, 100579) for HR as well.

The overall thrend from diff-graphs 8.4 8.4 is that the final performance is in between the performance of the respective used descriptors. This is however not definitely true, in terms of the average AUC or number of best results In case of HR we can see that the average AUC performance is better then the nHBAcc and nRotB, while in terms of number of best AUC the HR take second place. Also the worst AUC of HR is better then the worst AUC of nHBAcc or nRotB. Based on that the HR seem to be more stable that the nHBAcc or nRotB.

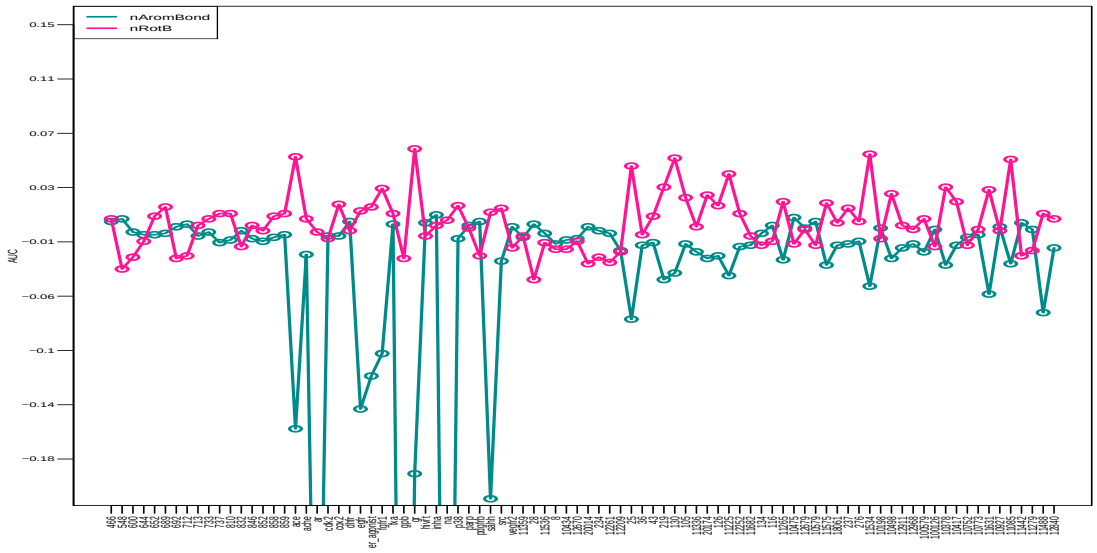



Figure 8.8: Difference in AUC of nAromBond and nRotB to AR

| | HR | nHBAcc | nRotB |
|---|---|---|---|
| mean | 0.7771 | 0.7685 | 0.7523 |
| max | 0.999 | 0.999 | 0.987 |
| min | 0.482 | 0.455 | 0.465 |
| #best | 31 | 39 | 21 |
| improvement | 0 | -0.761 | -2.187 |

Table 8.7: AUC of nHBAcc and nRotB in compare to HR

The situation in case of AR is different to the HR as the nAromBond perform poorly in general. On the other hand it gets the higher max AUC then AR, nRotB. The interesting fact here is, that the nRotB is able to save the performance in case that the nAromBond failed. In this case the AR does not outperform the nRotB in terms of AUC like in case of HR. But the difference is so small, if we consider the possible impact of nAromBond, that we can assume that the behaviour is almost the same as in case of AR.

Based on the experiments we can do several observation of two-descriptor based parametrization behaviour:

- Usage of two descriptors lead to the improvement in terms of AUC.
- If one of the descriptors perform poorly and the other not, then the other could compensate that.
- For some datasets the utilization of two-descriptor parametrization can improve the results.

In the second phase of this experiment, we try to use two descriptors nHeavyAtom,ETA_dAlpha_A (*HE*) to form a parametrization. The only conversion method used is 8bits, intUnaryCoding. The important aspect of this setup is that the ETA_dAlpha_A descriptor is (on used datasets) smaller then 0.6, as we use intUnaryCoding, the final value of this descriptor is always the same. The aim of this experiment is to verify that the usage of constatnt descriptor will not

| | AR | nAromBond | nRotB |
|---|---|---|---|
| mean | 0.752 | 0.706 | 0.7523 |
| max | 0.993 | 0.994 | 0.987 |
| min | 0.462 | 0.19 | 0.465 |
| #best | 40 | 8 | 43 |
| improvement | 0 | -4.042 | 0.027 |

Table 8.8: AUC of nAromBond and nRotB in compare to AR

influence the performance of final reprezentation. The experiment 8.4 9 prove our hypothesis. Thanks to this feature the constatn descriptor could be removed from the parametrization without any impact on the parametrization performance.

| | nHeavyAtom | nHeavyAtom,ETA_dAlpha_A (HE) |
|---|---|---|
| mean | 0.7782 | 0.7782 |
| max | 0.996 | 0.996 |
| min | 0.502 | 0.502 |
| #best | 88 | 88 |
| improvement | 0 | 0 |

Table 8.9: AUC of HE in compare to nHeavyAtom

## 8.5 Three-descriptors parameterization

In this series of experiments we continue in addition of descriptors into the representations. We utilized already used descriptors: nHeavyAtom, nHBAcc, nRotB and nAromBond. We assemble three fingerprints, the shortcuts were assigned to each setting for better identification as well:

- nHeavyAtom, nHBAcc, nRotB (*HHR*)
- nHeavyAtom, nAromBond, nRotB (*HAR*)
- nHeavyAtom, nHBAcc, nAromBond (*HHA*)

We reuse the combinations from second experiment and add nHeavyAtom descriptor into each. The 8bits, intUnaryCoding conversion method is used for all the descriptors. Only the HHR outperform 8.5 8.5 9 the BASE in AUC, while

| | BASE | HHR | HAR | HHA |
|---|---|---|---|---|
| mean | 0.7786 | 0.7798 | 0.7725 | 0.7779 |
| max | 0.995 | 0.998 | 0.995 | 0.998 |
| min | 0.498 | 0.502 | 0.493 | 0.498 |
| #best | 32 | 26 | 17 | 20 |
| improvement | 0 | 0.113 | -0.535 | -0.055 |

Table 8.10: AUC of HHR, HAR, HHA in compare to BASE

the AUC outperform all tested method in terms of best AUC resutls. In fact the

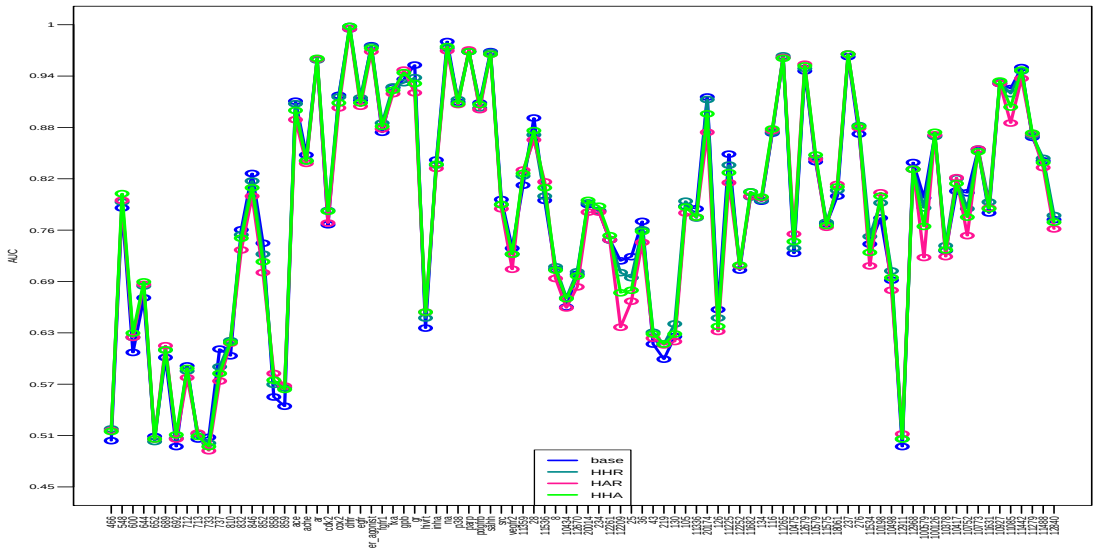



Figure 8.9: AUC of HHR, HAR, HHA

HR also outperform the three-descriptor parametrization. So the three descriptor basically bring improvement only in area of average AUC and in case of HHR also in the minimal reached AUC (0.502).

We choose HHR, due it's best performance, to compare the three-desciptor parametrization with the two(HR) and single(nHeavyAtom,nHBAcc,nRotB) descriptor parametrization. HHR performs best in term of average AUC, on the oth-

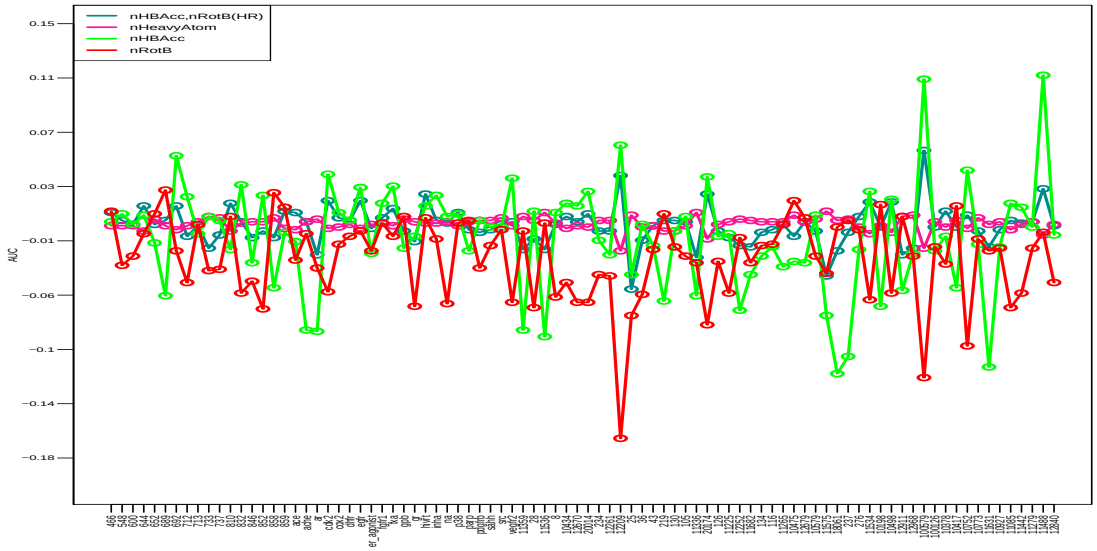



Figure 8.10: Difference in AUC of nHeavyAtom,nHBAcc,nRotB and HR to HRR

er hand single descriptor fingerprints (nHeavyAtom, nHBAcc) outperform HHR in terms of number of best AUC. From the diff-graph 8.5 we can see that the performance of HHR is similar to the nHeavyAtom performance. This observation is also confirmed by the smallest difference in average AUC 8.5 9. The results of this experiments are same as in case of two-descriptor parametrization. The usage of more descriptor lead to improvment in average AUC. This can be seen as some kind of stabilization process. The HHR's AUC is somewhere in between

| | HHR | HR | nHeavyAtom | nHBAcc | nRotB |
|---|---|---|---|---|---|
| mean | 0.7798 | 0.7771 | 0.7782 | 0.7685 | 0.7523 |
| max | 0.998 | 0.999 | 0.996 | 0.999 | 0.987 |
| min | 0.502 | 0.482 | 0.502 | 0.455 | 0.465 |
| #best | 18 | 11 | 31 | 32 | 13 |
| improvement | 0 | -0.239 | -0.145 | -1 | -2.426 |

Table 8.11: AUC of nHeavyAtom,nHBAcc,nRotB and HR in compare to HHR

the the AUCs of the nHeavyAtom,nHBAcc,nRotB and HR. This fact also leads to decrease in term of best AUC results.

## 8.6 Four descriptors

In this experiment we utilize four descriptors (nHeavyAtom, nHBAcc, nHBDon, nRotB) to form parametrizations. The difference between the parametrization in in number of used bits while one representation use 4bits for each descriptor ((4bit)) the other use 8bits (*8bit*). All descriptors use the same conversion method, intUnaryCoding. We also provide comparision to the BASE parametrization. In this case we decide to provide only the diff-graph 8.6, as the performance of BASE is well known from the previous graphs and the diff-graph give better view on the performance differences. As the results 8.6 9 show 4bit

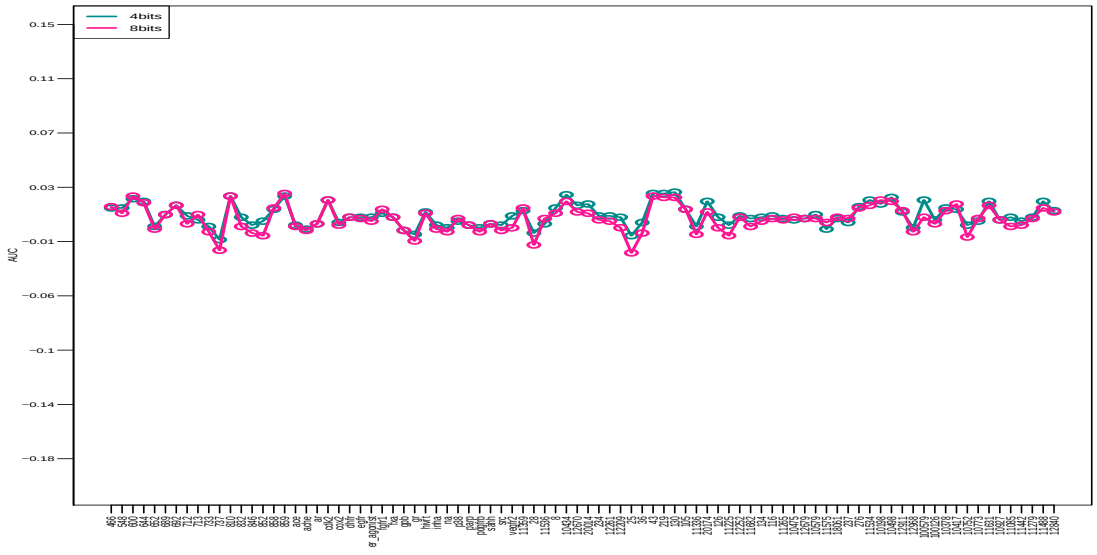



Figure 8.11: AUC of 4bit and 8bit

and 8bit outperform BASE method in terms of average AUC as well in number of best AUC. The 4bits get is best in terms of AUC for more than half datasets (50 from 88). There are two interesting facts: 1) 4bits perform better than 8bit 2) 4bits and 8bits reach good max AUC.

The first fact is probably caused by the chosen data, as the 4bits suite better than 8bits. The important notice is that the 4bits outperform 8bits on DUD, MUV and ChEMBL. The performance was most similar on MUV where 4bits

| | BASE | 4bits | 8bits |
|---|---|---|---|
| mean | 0.7786 | 0.7839 | 0.7814 |
| max | 0.995 | 0.999 | 0.999 |
| min | 0.498 | 0.506 | 0.502 |
| #best | 24 | 50 | 26 |
| improvement | 0 | 0.473 | 0.251 |

Table 8.12: AUC of 4bits and 8bits in compare to BASE

and 8bits perform same in number of best AUC, only the 4bits gets better average AUC. Further analyzation of this fact can lead to better understanding of VectorFP behaviour, however such analyzation is not part of this thesis.

In the experiments with two and three-descriptor parametrization we observe, that the addition of descriptors cause the stabilization of the final parameterization. This should lead to better average AUC but to decrease of number of best AUC. Both 4bits and 8bits got high max AUC and both cross the 0.5 value with respective minimal AUC. We do not expect such max values, as addition of descriptor in two,three experiments looks to have normalization effect on extreme results. To get a better picture we decide to compare 4bits with HHR, nHeavyAtom, nHBAcc and nRotB. The fact that 4bits outperformed all oth-

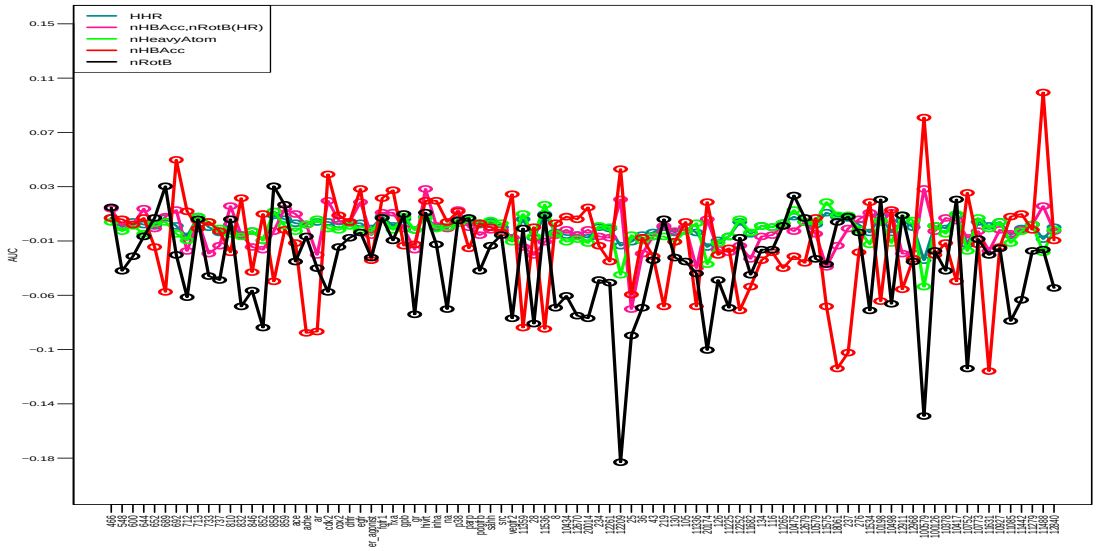



Figure 8.12: Difference in AUC of HHR, HR, nHeavyAtom, nHBAcc and nRotB to 4bits

er methods in terms of average AUC in not suprising. In experiments with two, three-based representation the complex(two, three-descriptor) representation was always outperformed in terms of best AUC results 8.4 8.4 8.5, the more comlex the representation was the better perform single based reprezentation. However 4bits outperform 8.6 8.6 9 single based representation in number of best fingerprints.

This is important as it shows, that utilization of more descriptors can lead not only to better AUC but also to better number of best AUC.

| | 4bits | HHR | HR | nHeavyAtom | nHBAcc | nRotB |
|---|---|---|---|---|---|---|
| mean | 0.7839 | 0.7798 | 0.7771 | 0.7782 | 0.7685 | 0.7523 |
| max | 0.999 | 0.998 | 0.999 | 0.996 | 0.999 | 0.987 |
| min | 0.506 | 0.502 | 0.482 | 0.502 | 0.455 | 0.465 |
| #best | 32 | 3 | 8 | 14 | 29 | 13 |
| improvement | 0 | -0.36 | -0.599 | -0.505 | -1.36 | -2.786 |

Table 8.13: AUC of HHR, HR, nHeavyAtom, nHBAcc and nRotB in compare to 4bits

## 8.7 Existing fingerprints comparison

In the last experiment we decide to compare 4bit fingerprint with other standard fingerprints. We choose three fingerprints to compare with.

- Extended connectivity fingerprints (*ECFP4*)
- MACCS (*MACCS*)
- Topological torsion (*TT*)

All three fingerprints were used in the benchmarking study [21], under the same names. The Topological torsion fingerprints were generally ranked among the top fingerprints. In terms of average AUC our method has been outper-

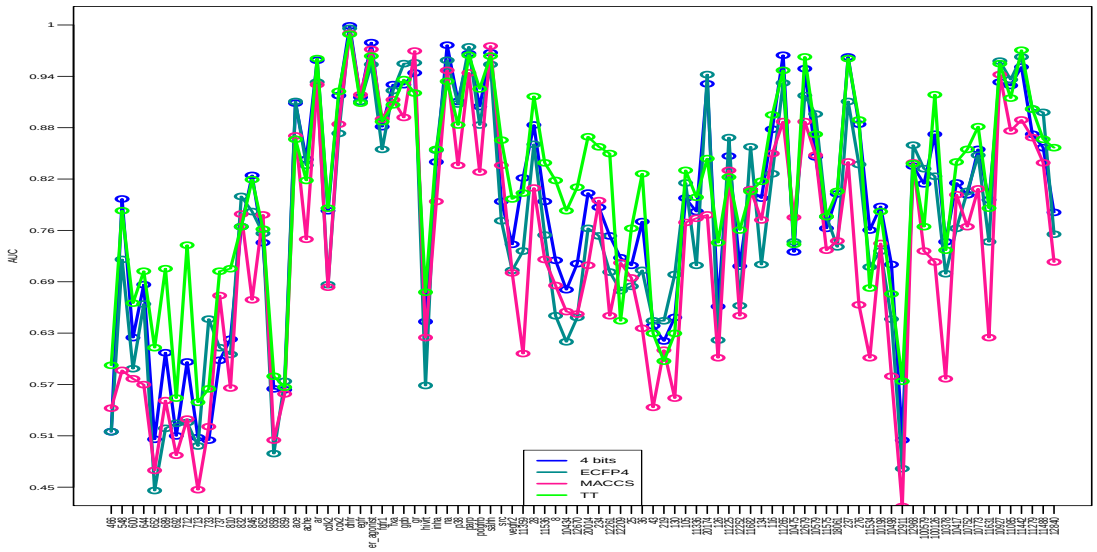



Figure 8.13: AUC of 4bits, ECFP4, MACCS and TT

formed 8.7 8.7 9 only by the TT fingerprints. In case of number of best results 4bits take third place after TT and ECFP4.

| | 4bits | ECFP4 | MACCS | TT |
|---|---|---|---|---|
| mean | 0.7839 | 0.7643 | 0.7333 | 0.8034 |
| max | 0.999 | 0.996 | 0.99 | 0.989 |
| min | 0.506 | 0.446 | 0.427 | 0.551 |
| #best | 17 | 20 | 6 | 47 |
| improvement | 0 | -1.725 | -4.453 | 1.715 |

Table 8.14: AUC of ECFP4, MACCS, TT in compare to 4bits

# 9. Conclusion

The aim of this thesis is to design new fingerprint-like representation for chemical compounds. The representation should be designed to be used in ligand-based virtual screening.

We proposed fingerprint based representation called VectorFP. The representation is highly parameterizable. The physicochemical descriptors of fragments are used to represent the whole chemical compound. The representation was tested in contemporary benchmarking framework and compared with standard 2D fingerprints.

In experiments we try several parametrization to determine the influence of the parametrization on the VectorFP performance. Our findings show that the chosen parametrization, (mainly used physicochemical descriptors but even the number of bit used per single descriptor) play significant role in the final performance.

In compare to the other 2D fingerprints our method recorded average results in general. However on some datasets we accomplished the best results between the tested 2D fingerprints. The experiments show that the proper parametrization is key to success. The right parametrization differ for each dataset. Data mining techniques could be a solution for the parametrization selection. The investigation of parametrization selection should be the next step in VectorFP development.

# Bibliography


[1] A. Bruce, J. Alexander, L. Julian, R. Martin, R. Keith, and W. Peter, *Molecular Biology of the Cell, Fourth Edition*. Garland, 2002.

[2] J. Drews, "Drug discovery: A historical perspective," *Science*, vol. 287, pp. 1960–1964, 2000.

[3] J. Hughes, S. Rees, S.and Kalindjian, and K. Philpott, "Principles of early drug discovery," *British Journal of Pharmacology*, vol. 162, p. 1239–1249, 2011.

[4] K. Sweeny, "Pharmaceutical industry project equity, sustainability and industry development working paper series," 2002.

[5] J. DiMasi, R. Hansen, and G. HG, "The price of innovation: new estimates of drug development costs," *Journal of Health Economics*, vol. 22, pp. 151–185, 2003.

[6] T. Bernard and D. K. Stefanie, "The biochemistry of drug metabolism–an introduction: part 1. principles and overview," *Chemistry & Biodiversity*, vol. 3, pp. 1053–1101, 2006.

[7] T. Cheng, Q. Li, Z. Zhou, Y. Wang, and S. Bryant, "Structure-based virtual screening for drug discovery: a problem-centric review," *The AAPS Journal*, vol. 14, no. 1, pp. 133–141, 2012. [Online]. Available: http://dx.doi.org/10.1208/s12248-012-9322-0

[8] "Pubchem," apr 2014. [Online]. Available: https://pubchem.ncbi.nlm.nih.gov/

[9] A. Gaulton, L. Bellis, A. Bento, M. Chambers, J. anb Davies, A. Hersey, Y. Light, S. McGlinchey, D. Michalovich, B. Al-Lazikani, and J. Overington, "hembl: a large-scale bioactivity database for drug discover," *Nucleic Acids Research*, vol. 40, 2012.

[10] "Chembl," apr 2014. [Online]. Available: https://www.ebi.ac.uk/chembl/

[11] J. J. Irwin and B. K. Shoichet, "Zinc - a free database of commercially available compounds for virtual screening," *Journal of Chemical Information and Modeling*, vol. 45, no. 1, pp. 177–182, 2005, pMID: 15667143. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci049714%2B

[12] "Zinc," apr 2014. [Online]. Available: http://zinc.docking.org/

[13] J. Overington, "Are there around 1019 lipinski-like small molecules?" 2012, cite 28.3.2014. [Online]. Available: http://chembl.blogspot.com/2012/05/are-there-around-10-19-lipinski-like.html

[14] D. Hoksza and D. Svozil, "Exploration of chemical space by molecular morphing," in *Bioinformatics and Bioengineering (BIBE), 2011 IEEE 11th International Conference on*, Oct 2011, pp. 201–208.

[15] B. Battersby and M. Trau, "Novel miniaturized systems in high-throughput screening," *Trends in Biotechnologi*, vol. 20, pp. 167–173, 2002.

[16] K. Heikamp and J. Bajorath, "The future of virtual compound screening," *Chemical Biology & Drug Design*, vol. 81, no. 1, pp. 33–40, 2013. [Online]. Available: http://dx.doi.org/10.1111/cbdd.12054

[17] S. Ekins, J. Mestres, and B. Testa, "In silico pharmacology for drug discovery: methods for virtual ligand screening and profiling," *British Journal of Pharmacology*, vol. 152, pp. 9–20, 2007.

[18] K. Pors, *Drug Discovery into the 21st Century, Drug Discovery and Development - Present and Future*, D. I. Kapetanović, Ed. InTech, 2011.

[19] B. K. Shoichet, "Virtual screening of chemical libraries," *Nature*, vol. 432, p. 862–865, 2004.

[20] M. Vogt and J. Bajorath, "Chemoinformatics: A view of the field and current trends in method development," *Bioorganic & Medicinal Chemistry*, vol. 20, no. 18, pp. 5317 – 5323, 2012, chemoinformatics in Drug Discovery. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0968089612002209

[21] S. Riniker and G. Landrum, "Open-source platform to benchmark fingerprints for ligand-based virtual screening," *Journal of Cheminformatics*, vol. 5, no. 1, p. 26, 2013. [Online]. Available: http://www.jcheminf.com/content/5/1/26

[22] R. E. Carhart, D. H. Smith, and R. Venkataraghavan, "Atom pairs as molecular features in structure-activity studies: definition and applications," *Journal of Chemical Information and Computer Sciences*, vol. 25, no. 2, pp. 64–73, 1985. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci00046a002

[23] S. K. Kearsley, S. Sallamack, E. M. Fluder, J. D. Andose, R. T. Mosley, and R. P. Sheridan, "Chemical similarity using physiochemical property descriptors†," *Journal of Chemical Information and Computer Sciences*, vol. 36, no. 1, pp. 118–127, 1996. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci950274j

[24] R. Nilakantan, N. Bauman, J. S. Dixon, and R. Venkataraghavan, "Topological torsion: a new molecular descriptor for sar applications. comparison with other descriptors," *Journal of Chemical Information and Computer Sciences*, vol. 27, no. 2, pp. 82–85, 1987. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci00054a008

[25] J. Hert, P. Willett, D. J. Wilton, P. Acklin, K. Azzaoui, E. Jacoby, and A. Schuffenhauer, "Comparison of topological descriptors for similarity-based virtual screening using multiple bioactive reference structures," *Organic and Biomolecular Chemistry*, vol. 2, pp. 3256–3266, 2004. [Online]. Available: http://dx.doi.org/10.1039/B409865J

[26] H. L. Morgan, "The generation of a unique machine description for chemical structures-a technique developed at chemical abstracts service." *Journal of Chemical Documentation*, vol. 5, no. 2, pp. 107–113, 1965. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/c160017a018

[27] N. Huang, B. K. Shoichet, and J. J. Irwin, "Benchmarking sets for molecular docking," *Journal of Medicinal Chemistry*, vol. 49, no. 23, pp. 6789–6801, 2006. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/jm0608356

[28] K. Heikamp and J. Bajorath, "Large-scale similarity search profiling of chembl compound data sets," *Journal of Chemical Information and Modeling*, vol. 51, no. 8, pp. 1831–1839, 2011. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci200199u

[29] S. G. Rohrer and K. Baumann, "Maximum unbiased validation (muv) data sets for virtual screening based on pubchem bioactivity data," *Journal of Chemical Information and Modeling*, vol. 49, no. 2, pp. 169–184, 2009. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci8002649

[30] J.-F. Truchon and C. I. Bayly, "Evaluating virtual screening methods: Good and bad metrics for the "early recognition" problem," *Journal of Chemical Information and Modeling*, vol. 47, no. 2, pp. 488–508, 2007, pMID: 17288412. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci600426e

[31] R. P. Sheridan, S. B. Singh, E. M. Fluder, and S. K. Kearsley, "Protocols for bridging the peptide to nonpeptide gap in topological similarity searches," *Journal of Chemical Information and Computer Sciences*, vol. 41, no. 5, pp. 1395–1406, 2001. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci0100144

[32] M. L. Verdonk, V. Berdini, M. J. Hartshorn, W. T. M. Mooij, C. W. Murray, R. D. Taylor, and P. Watson, "Virtual screening using protein-ligand docking: Avoiding artificial enrichment," *Journal of Chemical Information and Computer Sciences*, vol. 44, no. 3, pp. 793–806, 2004, pMID: 15154744. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/ci034289q

[33] "Maccs structural keys," 2011, accelrys: San Diego, CA.

[34] "Python," apr 2014. [Online]. Available: https://www.python.org/

[35] "Rdkit: Cheminformatics and machine learning softwares," 2014. [Online]. Available: http://www.rdkit.org

[36] C. W. Yap, "Padel-descriptor: An open source software to calculate molecular descriptors and fingerprints," *Journal of Computational Chemistry*, vol. 32, p. 1466–147, 2011.

# List of Abbreviations

QSAR - quantitative structure-activity relationship
HTS - high-throughput screening
VS - virtual screening
SBVS - structure-based virtual screening
LBVS - ligand-based virtual screening
PFP - parmacophore fingerprints
FP - fingerprints
AP - atom pairs fingerprints
TT - topological torsions fingerprints
ECFP - extended connectivity fingerprints
ROC - receiver operating characteristic
AUC - area under the curve
EF - enrichment factor
RIE - robust initial enhancement
BEDROC - Boltzmann-enhanced discrimination of ROC

# Appendix

## CD content

- *benchmark* - Contains benchmarking framework. The scoring/fingerprint_lib.py file contains aditional functionality to invoke configuration from VectorFP.
- *PaDEL* - Contains PaDEL.
- *vectorFP* - Contains implementation of VectorFP used in this thesis.

CD does not contains RDKit library nor Python. Both of them are required in order to run the benchmarking framework or VectorFP. Also execution of benchmarking platform require presence of path to vectorFP/src in system property PYTHON_PATH.

## BEDROC summary tables

| | BASE | 8 bits, intCopy | 8 bits, intUnaryCoding |
|---|---|---|---|
| mean | 0.7704 | 0.7609 | 0.7602 |
| max | 0.995 | 0.999 | 0.999 |
| min | 0.481 | 0.432 | 0.44 |
| #best | 50 | 18 | 22 |
| improvement | 0 | -0.843 | -0.902 |

Table 9.1: BEDROC of nHBAcc in compare to BASE

| | BASE | 8 bits, intCopy | 8 bits, intUnaryCoding |
|---|---|---|---|
| mean | 0.7704 | 0.7479 | 0.7498 |
| max | 0.995 | 0.999 | 0.999 |
| min | 0.481 | 0.459 | 0.467 |
| #best | 55 | 16 | 24 |
| improvement | 0 | -1.987 | -1.816 |

Table 9.2: BEDROC of nHBDon in compare to BASE

| | BASE | 8 bits, intCopy | 8 bits, intUnaryCoding | 16 bits, intUnaryCoding |
|---|---|---|---|---|
| mean | 0.7704 | 0.7696 | 0.7699 | 0.7687 |
| max | 0.995 | 0.995 | 0.995 | 0.995 |
| min | 0.481 | 0.489 | 0.486 | 0.486 |
| #best | 23 | 25 | 13 | 2 |
| improvement | 0 | -0.071 | -0.049 | -0.15 |

Table 9.3: BEDROC of nHeavyAtom in compare to BASE

| | BASE | 8 bits, intCopy | 8 bits, intUnaryCoding |
|---|---|---|---|
| mean | 0.7704 | 0.7424 | 0.7435 |
| max | 0.995 | 0.98 | 0.985 |
| min | 0.481 | 0.45 | 0.451 |
| #best | 66 | 8 | 20 |
| improvement | 0 | -2.469 | -2.371 |

Table 9.4: BEDROC of nRotB in compare to BASE

| | BASE | HR | AR | HA |
|---|---|---|---|---|
| mean | 0.7704 | 0.7688 | 0.743 | 0.7599 |
| max | 0.995 | 0.999 | 0.993 | 0.999 |
| min | 0.481 | 0.467 | 0.446 | 0.464 |
| #best | 33 | 29 | 12 | 19 |
| improvement | 0 | -0.144 | -2.419 | -0.926 |

Table 9.5: BEDROC of HR, AR, HA in compare to BASE

| | AR | nAromBond | nRotB |
|---|---|---|---|
| mean | 0.743 | 0.6969 | 0.7435 |
| max | 0.993 | 0.994 | 0.985 |
| min | 0.446 | 0.189 | 0.451 |
| #best | 40 | 7 | 43 |
| improvement | 0 | -4.05 | 0.048 |

Table 9.6: BEDROC of nAromBond and nRotB in compare to AR

| | nHR | nHBAcc | nRotB |
|---|---|---|---|
| mean | 0.7688 | 0.7602 | 0.7435 |
| max | 0.999 | 0.999 | 0.985 |
| min | 0.467 | 0.44 | 0.451 |
| #best | 31 | 39 | 21 |
| improvement | 0 | -0.758 | -2.227 |

Table 9.7: BEDROC of nHBAcc and nRotB in compare to HR

| | nHeavyAtom | nHeavyAtom, ETA_dAlpha_A (HA) |
|---|---|---|
| mean | 0.7699 | 0.7699 |
| max | 0.995 | 0.995 |
| min | 0.486 | 0.486 |
| #best | 88 | 88 |
| improvement | 0 | 0 |

Table 9.8: BEDROC of HE in compare to nHeavyAtom

| | BASE | HHR | HAR | HHA |
|---|---|---|---|---|
| mean | 0.7704 | 0.7716 | 0.7639 | 0.7697 |
| max | 0.995 | 0.998 | 0.995 | 0.998 |
| min | 0.481 | 0.487 | 0.478 | 0.483 |
| #best | 33 | 26 | 17 | 19 |
| improvement | 0 | 0.102 | -0.578 | -0.069 |

Table 9.9: BEDROC of HHR, HAR, HHA in compare to BASE

| | HHR | HR | nHeavyAtom | nHBAcc | nRotB |
|---|---|---|---|---|---|
| mean | 0.7716 | 0.7688 | 0.7699 | 0.7602 | 0.7435 |
| max | 0.998 | 0.999 | 0.995 | 0.999 | 0.985 |
| min | 0.487 | 0.467 | 0.486 | 0.44 | 0.451 |
| #best | 16 | 10 | 30 | 32 | 13 |
| improvement | 0 | -0.246 | -0.151 | -1.004 | -2.473 |

Table 9.10: BEDROC of nHeavyAtom,nHBAcc,nRotB and HR in compare to HHR

| | BASE | 4bits | 8bits |
|---|---|---|---|
| mean | 0.7704 | 0.7759 | 0.7732 |
| max | 0.995 | 0.999 | 0.999 |
| min | 0.481 | 0.49 | 0.487 |
| #best | 25 | 49 | 23 |
| improvement | 0 | 0.479 | 0.244 |

Table 9.11: BEDROC of 4bits and 8bits in compare to BASE

| | 4 bits | HHR | HR | nHeavyAtom | nHBAcc | nRotB |
|---|---|---|---|---|---|---|
| mean | 0.7759 | 0.7716 | 0.7688 | 0.7699 | 0.7602 | 0.7435 |
| max | 0.999 | 0.998 | 0.999 | 0.995 | 0.999 | 0.985 |
| min | 0.49 | 0.487 | 0.467 | 0.486 | 0.44 | 0.451 |
| #best | 31 | 2 | 8 | 13 | 30 | 13 |
| improvement | 0 | -0.377 | -0.623 | -0.528 | -1.381 | -2.85 |

Table 9.12: BEDROC of HHR, HR, nHeavyAtom, nHBAcc and nRotB in compare to 4bits

| | 4 bits | ECFP4 | MACCS | TT |
|---|---|---|---|---|
| mean | 0.7759 | 0.756 | 0.7234 | 0.7953 |
| max | 0.999 | 0.996 | 0.99 | 0.988 |
| min | 0.49 | 0.432 | 0.413 | 0.535 |
| #best | 18 | 20 | 6 | 47 |
| improvement | 0 | -1.747 | -4.615 | 1.706 |

Table 9.13: BEDROC of ECFP4, MACCS, TT in compare to 4bits