Charles University

Faculty of Mathematics and Physics

BACHELOR THESIS

2017 Jiří Kolář

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

BACHELOR THESIS

Jiří Kolář

Utilization of latent semantic analysis in virtual screening

Department of Software Engineering

Supervisor of the bachelor thesis:	RNDr. David Hoksza, Ph.D.
Study programme:	Computer science
Study branch:	General computer science

Prague 2017

I declare that I carried out this bachelor 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 Sb., the Copyright Act, as amended, in particular the fact that the Charles University has the right to conclude a license agreement on the use of this work as a school work pursuant to Section 60 subsection 1 of the Copyright Act.

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

Acknowledgement

I would like to express my gratitude to my supervisor RNDr. David Hoksza, Ph.D., his constructive comments and pointed questions helped me to look at my problems from different angle. I want to also thank to Petr Škoda, who helped me to solve some technical and implementation issues.

Title: Utilization of latent semantic analysis in virtual screening

Author: Jiří Kolář

Department: Department of Software Engineering

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

Abstract: Aim of this thesis is to investigate utilisation of latent semantic indexing in Virtual screening. We have examined existing VS method called latent semantic structural indexing (LaSSI) and compared performance of different structural fingerprints. Additionally, we have developed a new model that compare fragments of molecules by usage of latent semantic indexing. Fragments are characterized by formula based counts and descriptors describing the physicochemical properties. Results of our methods are compared to VS techniques using directly standard fingerprints.

Keywords: virtual screening cheminformatics ligand-based fingerprints ECFP TT latent semantic analysis LaSSI

Contents

Introduction

Virtual screening has been widely adopted by scientists in the field of drug discovery. It can safe both time and financial resources especially in early stage of research. There is a wide range of different virtual screening techniques and we also present a brief overview in this thesis. Popular are similarity methods often joined with machine learning techniques.

Aim of this thesis is to examine utilization of latent semantic indexing (LSI) for Virtual screening (VS). LSI uses singular value decomposition (SVD) to identify hidden concepts in collections of documents. Hidden concepts can capture latent information in the text, which is used by searching algorithms. Similarly, molecule can be described by a set of features or fragments that can be viewed as a document containing terms. We test the idea, if usage of similarity methods on latent variables obtained by SVD can improve performance of Structure-based virtual screening.

We are going to study influence of different kind of sets of features or fragments on resulting performance. Additionally, we are going to compare results of LSI with each other and with standard similarity techniques directly using fingerprints (AP, TT, ECFP, MACCS).

First chapter describes representation of molecular structure in computer science and methods for analysis and estimation of molecular properties.

Second chapter is focused on virtual screening, its usage in drug-discovery process. We discuss principle and usage of protein-ligand docking, pharmacophore mapping, similarity searching and machine learning.

Third chapter summarize general information about developed application for VS based on LSI, we present used data set for benchmarking.

Last chapter includes description of implementation details for each developed VS method and comparison of results.

1. Molecules ”in silico”

At first, we are going to discuss how to represent molecules in computer science and how to describe their structural, chemical and physical properties. Subsequently, we are going to use these information in design of VS technique.

1.1 Molecule representation

Chemical molecules are usually represented by molecular graphs. A molecular graph is an abstract structure, which consists of nodes (chemical elements) and edges (chemical bonds), it can only describe the topological information of molecule. A molecule can be split into molecular fragments, which are represented by subgraphs, subsets of the nodes and edges of the graph.

A single graph has many different drawings, that might look very different so it is necessary to recognise same molecules, this problem is well known in mathematics and it is called graph isomorphism.

Two graphs, G and H are isomorphic if there is a bijective mapping, f between set of nodes in G and H, such that two nodes u and v of G are adjacent in G if and only if $f(u)$ and $f(v)$ are adjacent in H.

There are well-developed algorithms to determine if two graphs are isomorphic, further information can be found in Read and Corneil [1].

1.1.1 Format of molecular graph

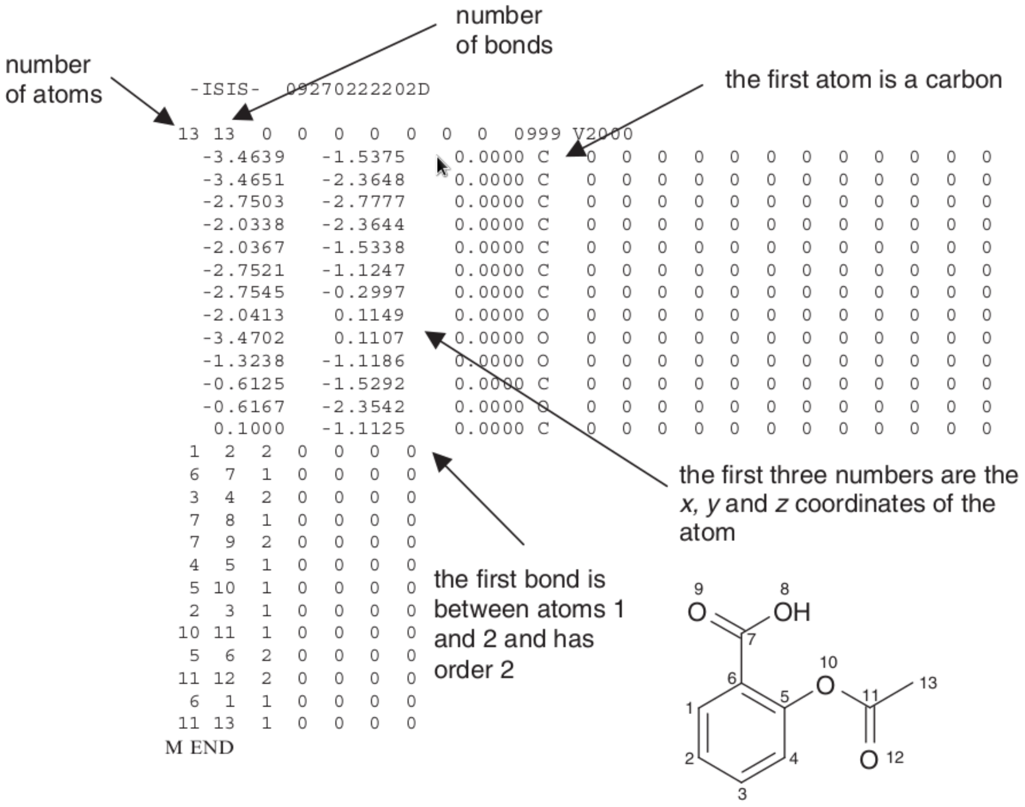
MDL information system

Connection table is a common type of representation, the representation of aspirin in MDL format can be seen in Fig. 1.1. The simplest type of connection table consists of two sections: first, a list of the atomic numbers of the atoms in the molecule and second a list of the bonds, specified as pairs of bonded atoms [2]. However there can be included more detailed information, for instance the hybridisation state, bond order or even spatial coordinates of the atoms to produce a standard chemical drawing.

Linear notation

On the other hand the most compact format use the linear notation. It uses alphanumeric characters to encode the molecular structure. The most wide spread linear notation is the Simplified Molecular Input Line Entry Specification (SMILES), see Fig. 1.3. Atoms are represented by their atomic symbols. Upper case symbols are used for aliphatic atoms and lower case for aromatic atoms. Hydrogen atoms are not normally explicitly represented. Double bonds are written by ”=” character, triple bonds by ”#”. Rings are dealt with by “breaking” one of the bonds in each ring; the presence of the ring is then indicated by appending an integer to the two atoms of the broken bond. Information about chirality, geometrical isomerism and aromaticity can be also encoded [2].

Number of different SMILES strings exist for a single molecule. Therefore, there were stated rules for a unique ordering of the atoms, which is called the



```
      -ISIS-  09270222202D

 13 13  0  0  0  0  0  0  0  0999 V2000
   -3.4639   -1.5375    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -3.4651   -2.3648    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -2.7503   -2.7777    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -2.0338   -2.3644    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -2.0367   -1.5338    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -2.7521   -1.1247    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -2.7545   -0.2997    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -2.0413    0.1149    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
   -3.4702    0.1107    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
   -1.3238   -1.1186    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
   -0.6125   -1.5292    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -0.6167   -2.3542    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    0.1000   -1.1125    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  1  2  2  0  0  0  0
  6  7  1  0  0  0  0
  3  4  2  0  0  0  0
  7  8  1  0  0  0  0
  7  9  2  0  0  0  0
  4  5  1  0  0  0  0
  5 10  1  0  0  0  0
  2  3  1  0  0  0  0
 10 11  1  0  0  0  0
  5  6  2  0  0  0  0
 11 12  2  0  0  0  0
  6  1  1  0  0  0  0
 11 13  1  0  0  0  0
M END
```

number of atoms
number of bonds
the first atom is a carbon
the first three numbers are the *x*, *y* and *z* coordinates of the atom
the first bond is between atoms 1 and 2 and has order 2
9
8
O
OH
7
10
6
5
O
11
13
1
2
4
3
O
12

Figure 1.1: The connection table for aspirin in the MDL format (hydrogen-suppressed form). The numbering of the atoms is as shown as in the chemical diagram [2].

canonical representation. Widely used algorithm for determining a canonical order of the atoms is the Morgan algorithm [3].

Morgan algorithm iteratively calculates the "connectivity values" to enable differentiation of the atoms. At first each atom is assigned a number of connected atoms. These numbers are recalculated in every iteration, until each atom is assigned a unique number. New "connectivity value" for each atom is equal to the sum of values assigned to neighbours in previous step. Several iterations are demonstrated on molecule of aspirin in Fig. 1.2. Atoms are listed in the descending order according to the "connectivity values". If there are multiple atoms that have the same value, additional properties are considered, such as bond order and atomic number respectively [2].

The resulting canonical ordering is then used to generate the unique SMILES string for the molecule [2], for example by CANGEN algorithm [4].

1.2 Molecular descriptors

Molecular descriptors have been designed in order to analyse and predict the properties of chemical structures. They provide basic information that can be used to predict biological activity. The numerical values of descriptors may represent the physicochemical properties of a molecule or only the statical about the number of atoms [2]. Widely used examples of the molecular descriptors from

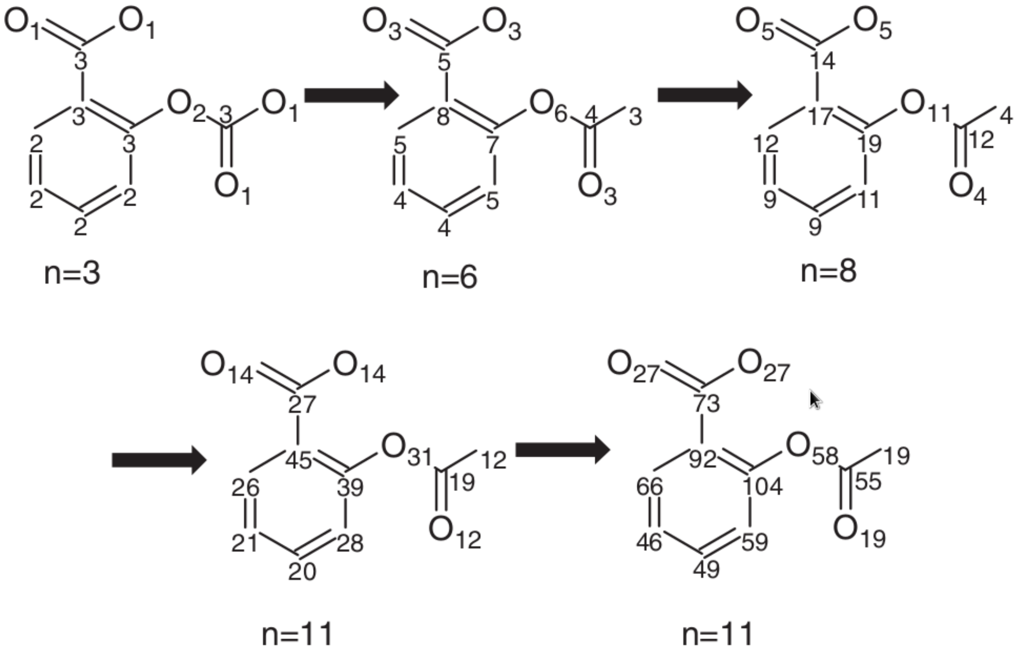
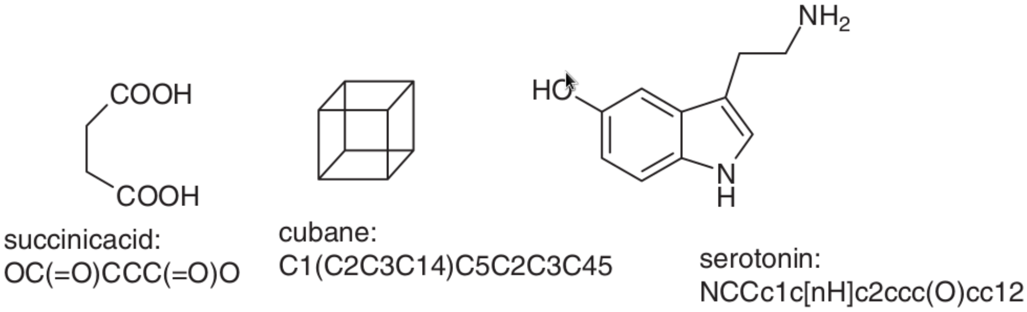


Figure 1.2: Illustration of the iterative construction of the atomic connectivity values during Morgan algorithm, n is the number of unique integer identifiers [2].



succinicacid:
OC(=O)CCC(=O)O

cubane:
C1(C2C3C14)C5C2C3C45

serotonin:
NCCc1c[nH]c2ccc(O)cc12

Figure 1.3: Illustration of some examples of the SMILES strings [2].

the 2-D structure can be found in the following text.

1.2.1 Simple structural formula based counts

The simplest molecular descriptors are counts of features such as hydrogen bond donors, hydrogen bond acceptors, ring systems (including aromatic rings), rotatable bonds and molecular weight. However these descriptors offer only a very limited information so they are often combined with the others [2].

1.2.2 Physicochemical properties

Very important property is the hydrophobicity. The molecule's hydrophobicity can affect how tightly it binds to a protein and its ability to pass through a cell membrane. It is commonly modelled by the usage of the logarithm of the partition coefficient between n-octanol and water, *SlogP* descriptor [5].

1.2.3 Molar Refractivity

The refractive index, descriptor *SMR*, term accounts for the polarisability of the molecule and does not vary much from one molecule to the another, the molar refractivity additionally depends on the molecular weight and density [2].

1.2.4 Topological indices

The topological indices are single-valued descriptors calculated from the 2-D graph representation of the molecule.

They characterise structures according to their size, complexity of branching and shape in general. Most popular are the chi molecular connectivity indices developed by Kier and Hall [6]. They used values of sigma, pi and lone pair electrons. An arbitrary atom i in the molecule is described by the simple delta value δ_i and the valence delta value δ_i^ν as

$$\delta_i = \sigma_i - h_i, \tag{1.1}$$

$$\delta_i^\nu = Z_i^\nu - h_i, \tag{1.2}$$

where σ_i is the number of sigma electrons, h_i is the number of hydrogen atoms bonded to the atom i, Z_i^ν is the total number of valence electrons. Zero chi index for a molecule is then computed as:

$$^0\chi = \sum_{atoms} \frac{1}{\sqrt{\delta_i}} \quad {}^0\chi^\nu = \sum_{atoms} \frac{1}{\sqrt{\delta_i^\nu}}. \tag{1.3}$$

There are also higher order chi indices which are computed over bonds or their sequences.

1.2.5 Kappa shape indices

The Kappa shape indices were designed by Hall and Kier [7] to characterize the molecular shape by comparing the molecule with the extreme shapes depending on various order of the indices. For the first order, kappa index is defined as:

$$^1\kappa = \frac{2^1 P_{max}^1 P_{min}}{(^1P)^2}, \tag{1.4}$$

where $^1P_{max}$ corresponds to the number of edges in the completely connected graph, $^1P_{min}$ is the number of bonds in the linear molecule and ^1P is the number of bonds in the specific molecule. The second-order kappa index is determined by the count of two-bond paths etc. There are also modification that also consider the orbital hybridisation of the atoms.

1.2.6 Electrotopological state indices

The Electrotopological state (E-state) indices are determined for each atom so the output value is rather a vector or a bitstring. However individual atomic E-states can be combined into the molecular descriptor by calculating the mean-square value over all atoms.

Intrinsic state of the atom i encodes its electronic and topological characteristics

$$I_i = \frac{(\delta_i^\nu + 1)}{\delta_i}, \tag{1.5}$$

and E-state, S_i of the atom i is given by the sum of the intrinsic state and all perturbations between other atoms

$$S_i = I_i + \Delta I_i = I_i + \sum_j^{\text{atoms}} \frac{I_i - I_j}{r_{ij}^2}, \tag{1.6}$$

where r_{ij} is the path length between atoms i and j [8].

1.2.7 Fingerprints

The structural fingerprints are originally used for searching tasks. It enables to extract information about similar molecules from the database efficiently. Fingerprint is a sequence of zeros and ones. Each bit indicates if particular structural feature is present or not. As there are many structural features it can be represented as a fragment dictionary or the information is hashed to hash key of a given length. In the case of hash key collisions must be handled, so multiple fragments corresponds to a single bit.

However molecular properties and biological activity correlate to the structural features, therefore there are attempts to use fingerprints as molecular descriptors, despite they were not designed for it. Instead of simple boolean value an integer frequency of the structural pattern can be used.

Atom pairs

The atom pair descriptors encode all pairs of atoms in a molecule together with the length of the shortest bond-by-bond path between them, and kind of bond [9]. For example the substructure $-CH_2\text{-}CH_2-$, is coded as "CX2-(2)-CX2", where X$n$indicates presence of n non-hydrogen neighbouring atoms.

The atom pairs were later modified by Kearsley et al. [10] in the sense that property of atoms are often more important than the specific element type. Atoms are identified as belonging to the one of seven binding property classes: cations, anions, neutral hydrogen bond donors and acceptors, atoms which are both donor and acceptor, hydrophobic atoms and all others.

Topological torsions

The topological torsions encode sequences of connected atoms together with their types, number of non-hydrogen connections and number of pi-electrons [11]. Typically, sequences have length of four atoms. However, the longer sequences can be also used.

Extended Connectivity Fingerprints

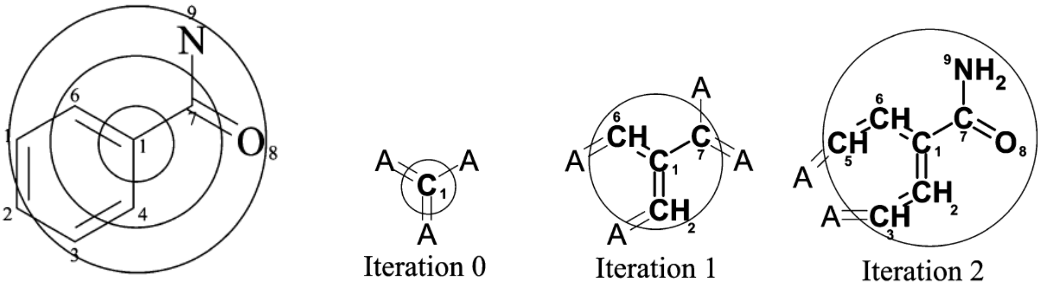
The extended Connectivity Fingerprints were explicitly designed for structure-activity modelling. ECFPs are circular fingerprints based on the Morgan algorithm [3] and they can represent wide range of structural features including stereochemical information.

Extended Connectivity Fingerprints are generated as follows [12]:

1. **Initialization:** Each atom is an assigned integer identifier.
2. **Update, iterative step:** Each atom identifier is updated to reflect the identifiers of each atom's neighbours, any duplicates are marked.
3. **Removal step:** Duplicates are reduced to a single representative in the final feature list (possible a number of duplicates is noted).

At each update step, there is created an array where are put all identifiers of bonds and atoms which should be added in this iteration and than they are hashed into a single integer, which is used for next iterative step.

Number of iterations is usually specified by the number of bonds as a diameter of the circle, for instance ECFP-2, assume circle with diameter of 2 bonds, so the update process ends at iteration number 1 (Fig. 1.4) for each atom.



(a) Overview of iterations

(b) Several iteration steps

Figure 1.4: Illustration of the iterative updating of the identifier for atom 1 in benzoic acid amide [12].

Extended Connectivity Fingerprints (ECFPs) and Functional Connectivity Fingerprints (FCFPs) differs in the assigning method of the initial identifiers. ECFPs are designed to capture precise atom environment substructural features, while FCFPs should capture more abstract role-based substructural features [12]. Initial identifiers are integers that are created by hash function from topological (number of immediate neighbours) and atomic properties (atomic mass, charge...). Detailed description can be found in Rogers and Hahn [12].

BCUT descriptors

BCUT descriptors can encode atomic properties relevant to intermolecular interactions. They are calculated from matrix representation of the molecule‘s connection table. BCUT matrices can be extended to encode also atomic charge, atomic polarisability and atomic hydrogen bonding ability [13]. Each matrix, encoding certain property, is further modified and the highest and lowest eigenvalues of the matrix are evaluated and used as descriptors.

2. Drug-discovery in modern chemistry

Modern organic chemistry and biochemistry is capable of synthesis of almost any compound in the chemical universe. However, chemical universe itself is huge, only the number of small organic molecules is exceeding 10^{60} [14]. Therefore modern drug discovery and drug design needs tools for efficient searching, filtering and evaluating of compounds with desired biological activity.

2.1 Virtual screening

Exact biological activity must be always confirmed experimentally, but which compounds should be likely purchased, synthesised and finally tested, that is where virtual screening comes in handy. Virtual screening is an *in silico* method that scores, ranks and filters a set of structures using one or more computational procedures [2] according to the defined criteria.

There is a wide range of criteria and methods by which the chemical compounds may be judged. Virtual screening can starts by general filters and substructure queries for possible candidates. Additionally, molecules that contain certain undesirable, or even toxic functionality can be eliminated.

After the molecule set is chosen, the biological activity is estimated by more involved methods. Virtual screening methods can be divided into four main classes based on amount of structural and bioactivity data that is available [15].

If 3D structure of the target's binding spot is known a protein–ligand docking can be employed [16]. If one or more active molecules are known, similarity searching can be used [17]. In case of multiple known active molecules, pharmacophore mapping can be carried out to determine common patterns of features that might be responsible for the biological activity [18]. If both active and inactive molecules are available, they can be used to train a machine learning technique such as statistical criteria, decision trees and neural networks [19].

Methods based on examples of known active molecules are usually referred to as ligand-based virtual screening methods. Whereas, methods based on known structure of target protein are referred to as structure-based virtual screening methods.

2.1.1 Protein–ligand docking

The protein–ligand docking is a term used for computational schemes that attempt to find the best matching between two molecules: a receptor and a ligand according to some score function [16].

It is a computationally intensive and complex procedure because it involves many degrees of freedom. The translation and rotation of one molecule relatively to the another involves six degrees of freedom. There are in addition the conformational degrees of freedom of both the ligand and the protein. The solvent may also play a significant role in the determining of the protein–ligand geometry and the free energy of binding. But it is often ignored [2]. Some successful usage of docking are summarized by Lyne [20]. Greatest shortcomings of the docking are connected to the used score functions. However, a promising improvement offers the machine learning scoring functions as was showed by Wójcikowski et al. [21].

Spatially 3-dimensional structures of more than 44,000 x-ray and nuclear magnetic resonance (NMR) structures of proteins and protein–ligand complexes are available in the Protein Data Bank (PDB).

2.1.2 Pharmacophore mapping

The pharmacophore mapping is the process of identification of common 3D pharmacophores in the set of known active molecules and it is usually followed by a 3D database search. Pharmacophores can be defined as the set of features, such as hydrogen bond donors and acceptors, positively and negatively charged groups, hydrophobic regions and aromatic ring, together with their relative spatial orientation. Some functional groups can have similar biological, chemical and physical properties, these are called *bioisosteres*.

There are various methods for pharmacophore mapping, for instance constrained systematic search, maximum likelihood method, maximum clique detection or even genetic algorithms, further reading can be found in Leach and Gillet [2].

2.1.3 Similarity searching

The similarity searching does not require the exact 3D structure of target binding spot and it is also not necessary to precisely identify pharmacophores responsible for the activity.

The major idea of this method is so called similar property principle, that similar molecules could have similar biological activity [22]. The query compound is used to score and rank every molecule in the testing set according to the similarity.

The 2D similarity searching methods usually describe molecules by fragment-based descriptors [17] such as atom pairs or extended connectivity fingerprints (see Section 1.2.7).

The 3D similarity searching methods additionally include atom positions and conformational properties of the molecules. The similarity can be than computed by alignment methods [23] or by superimposed electron density maps [24]. However additional dimension increases significantly the computational complexity.

Description of similarity in 2D

Most widely used method for evaluating similarity between binary descriptors is *Tanimoto similarity*, defined as

$$S_{AB} = \frac{c}{a + b - c}, \tag{2.1}$$

where S_{AB} is Tanimoto similarity coefficient between molecule A and B, a are bits set to one in molecule A, b bits set to one in molecule B and c are bits common to both A and B. Tanimoto coefficient ranges from zero to one. A value of one indicates that the molecules have identical fingerprint representations [2].

However an universal similarity coefficient can be evaluated as *Euclidean distance*, D_{AB}

$$D_{AB} = \left(\sum_{i=1}^{N} (x_{iA} - x_{iB})^2 \right)^{0.5}, \tag{2.2}$$

where x_{iA} is a value of property i in molecule A and x_{iB} in molecule B.

Or as *Cosine similarity*

$$S_{AB} = \frac{\sum_{i=1}^{N} x_{iA} x_{iB}}{\left(\sum_{i=1}^{N} (x_{iA})^2 \sum_{i=1}^{N} (x_{iB})^2 \right)^{0.5}}. \tag{2.3}$$

2.1.4 Machine learning

We will discuss usage of machine learning in ligand-based virtual screening which is currently a promising and fast developing area. Every method have to be trained with input data of active or also inactive molecules before actual screening.

Neural network

The artificial neural network is derived from biological neural networks, number of computational elements or nodes operate in parallel and while connected via weights that are modified during learning [25]. There are two widely used neural network architectures: feed-forward network and the Kohonen network.

The feed-forward neural network [26] is built of layers of nodes with assigned states between zero and one. There is one input layer, several layers with hidden nodes and one output layer, each layer is connected to adjacent layers by weighted edges. Input nodes can be assigned for example values of molecular descriptors. Parameters as states and weights are adjusted during training, where to each input molecule is known desired output. It is a supervised learning method, because it uses the values of the dependent variables to derive the model [2]. Feed-forward neural network was for instance used to predict the physicochemical properties of molecules [27].

The Kohonen network, or self-organising map [28] is an unsupervised learning method. Nodes are organised to rectangular grid and each node has an associated vector that corresponds to the input data (molecular descriptors). All vectors are assigned small random values which are than updated during learning. Thus, training creates areas of neighbouring nodes that have similar input data [2].

Decision tree

The decision trees are straight-forward and easily interpretable as a set of rules. It is used to divide large dataset into smaller and more homogeneous sets. At each decision step, the method identifies the feature, according to, the set would be divide into two most diverse subsets. Selection is done statistically for example via t-test.

The decision trees are often combined with the other machine learning algorithms. There are two widely used techniques, bagging and boosting. The bagging [29] is a type of the model averaging approach. The bagging model is formed by repeatedly selecting bootstrap samples of the dataset and training the trees on these data [25]. The boosting [30] combines large number of hypotheses, each of which is generated by training the given learning algorithm on a different set of examples. Trees can be built in sequential fashion, that previous trees are combined with new sample concentrating on less well predicted compounds. Or trees can be built in parallel and each tree can vote on the prediction, this method is referred to as Random Forest [31].

Naïve Bayesian

The naïve Bayesian (NB) method is based on Bayes rule for conditional probability, for example class (C) posterior probabilities given a feature vector X is equal to the class-conditional feature probability distribution times the prior class probability divided by the prior feature probability [25]

$$P(C = i|X = x) = \frac{P(X = x|C = i)P(C = i)}{P(X = x)}. \tag{2.4}$$

NB weights each molecular features, descriptors by assigning greater significance to features that appear to distinguish good samples from baseline samples [25].

NB is efficient, robust and easy to use. On the other hand it also suffers from its intrinsic simplicity, because it is unable to analyse combined effect of multiple descriptors [32]. However, the performance can be increased by usage of NB on molecular fragments features [33] instead of entire molecules.

Support vector machine classification

The support vector machine has become one of the most popular machine learning methods in drug design, virtual screening and combinatorial chemistry [25]. It is well describe, for instance, in monograph written by Cristianini et al. [34].

The training of a SVM is based on a the set of learning data belonging to two different classes. For these data SVM constructs the maximal separating hyperplane separating the training objects into two classes. If data are not linearly separable, than a kernel function is used to project the input data into a higher dimensional feature space, where a hyperplane can be constructed [25]. Testing data are then classified according to these hyperplanes.

2.2 Latent semantic indexing

The latent semantic indexing (LSI) is the method for data analysis, that assumes that there is some underlying or latent structure in the data. LSI is usually related to text searching and comparing of relevant documents. As words can have multiple synonyms or even different meanings, it is important to analyse surrounding text, not only queried words. Estimation of latent structure is done via singular value decomposition (SVD) of matrix formed by frequencies of used words in each document. We project queries and documents from high-dimensional space of words to low-dimensional latent semantic space, often 2-dimensional or 3-dimensional space for the purpose of visualization.

This can be also understood as least-squares method, because the projection of matrix A to lower dimensional space $\hat{A}$ is chosen in the way to minimize the Euclidean distance Δ between document's vectors

$$\Delta = ||A - \hat{A}||_2 \tag{2.5}$$

This setup has the consequence that the dimensions of the reduced space correspond to the axes of greatest variation.

2.2.1 Latent Semantic Structure Indexing

The latent semantic structure indexing (LaSSI) is a method firstly published by Hull et. al. [35], and it is used to analyse molecules, instead of documents. According to Hull, molecules are described by the vectors $\mathbf{a_j} = (d_{1j}, d_{2j} \dots d_{mj})^T$ where d_{ij} is the raw non-negative frequency of the descriptor i, which are for example atom pair or topological torsion descriptors. In context of LSI, molecule can be viewed as a document containing terms. And each structural feature is a different term. Thus, LaSSI tries to capture hidden concepts in structural patterns.

The molecular matrix of n molecules: $A \in \mathbb{R}^{m \times n}$ is decomposed by the singular value decomposition defined as

$$A = USV^T, \tag{2.6}$$

where, $U \in \mathbb{R}^{m \times r}$, $V \in \mathbb{R}^{r \times n}$ are the orthogonal matrices (r is the rank of A) and $S \in \mathbb{R}^{r \times r}$ is the diagonal matrix with elements $\sigma_{11}, \sigma_{22}, \dots, \sigma_{rr}$ which are called the singular numbers of matrix A and they are square root of the eigenvalues of the matrix $A^T A$. SVD can be written as follows

$$\begin{pmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mn} \end{pmatrix} = \begin{pmatrix} u_{11} & \cdots & u_{1r} \\ u_{21} & \cdots & u_{2r} \\ \vdots & \ddots & \vdots \\ u_{m1} & \cdots & u_{mr} \end{pmatrix} \begin{pmatrix} \sigma_{11} & & \\ & \ddots & \\ & & \sigma_{rr} \end{pmatrix} \begin{pmatrix} v_{11} & v_{21} & \cdots & v_{n1} \\ \vdots & \vdots & \ddots & \vdots \\ v_{1r} & v_{2r} & \cdots & v_{nr} \end{pmatrix}. \tag{2.7}$$

If we choose first k biggest singular numbers and we set other to zero $\sigma_{(k+1)(k+1)} \dots \sigma_{rr} = 0$, we get the k-th rank approximation of A, denoted by A_k, for $k < r$,

$$A_k = US_kV^T. \tag{2.8}$$

Each row of A belongs to one molecular descriptor, each column of A is description of one molecule. Row i of matrix V, $\mathbf{v_i}$, is a projection of molecule i to latent semantic space.

Similarity between 2 arbitrary molecules i_1 and i_2, computed in latent semantic space, is a cosine similarity between rows i_1 and i_2 of matrix V

$$\text{Similarity between molecules } i_1 \text{ and } i_2 = \frac{\mathbf{v_{i_1}} \cdot \mathbf{v_{i_2}}}{|\mathbf{v_{i_1}}||\mathbf{v_{i_2}}|} \tag{2.9}$$

Molecule that is originally not included in matrix A, has to be transformed to latent semantic space in order to determine similarity to other molecules. Let $\mathbf{z}$

be the vector describing the molecule by same molecular descriptors as molecules in matrix A, then vector $\mathbf{y}$ is the projection to latent semantic space defined as

$$\mathbf{y} = S_k^{-1} U^T \mathbf{z}. \tag{2.10}$$

Vector $\mathbf{y}$ can be treated as a row of matrix V.

2.2.2 QSAR model and projection to latent structures

The quantitative structure–activity relationship models try to find how are connected molecular properties and observed biological activity to the molecular structure. QSAR model describes estimated property as a function of descriptors [2].

If we select m descriptors, known for n training molecules. We have a descriptor matrix $A = (\mathbf{a_1}, \ldots \mathbf{a_n})$, where $\mathbf{a_j} = (d_{1j}, \ldots, d_{mj})$. And for each molecule j we know value of estimated property y_j, giving property vector $\mathbf{y} = (y_1, \ldots y_n)$. Finally, predicted functionality as function f of the descriptors can be obtained as

$$f : \mathbb{R}^m \to \mathbb{R}, y' = f(\mathbf{a}), \tag{2.11}$$

where y' is a value of estimated property for molecule described by descriptor vector $\mathbf{a}$.

An example how to determine function f is a linear regression or a partial least squares method [36].

Partial least squares

Partial least squares method does not use directly computed descriptors, but tries to find new latent variables that explains as much of the variance of the observed property and the descriptors as possible with minimal total number of variables. This method is also referred to as the projection to latent structures and it is a generalization of the problem solved by LaSSI [2].

Latent variable k, $\mathbf{t}_k = (t_{1k}, \ldots, t_{nk})$ is extracted as a linear combination of descriptors from the descriptor matrix $A = (d_{ij}$. Component t_{jk} of the latent variable $\mathbf{t}_k$ is defined as

$$t_{jk} = \sum_{i=1}^{m} p_{ik} d_{ij}, \tag{2.12}$$

where p_{ik} describes the influence of a descriptor i on $\mathbf{t}_k$. If we set the number of latent variable to $k_{\max} < m$, then the m-dimensional space is reduced to $k_{\max}$ dimensions. The resulting *score* matrix $T = (t_{jk})$ can be used similarly to original descriptor matrix for linear regression.

If the new variables takes into count only variance in the descriptor space and not the variance in the observed property y, we call them principal components [2] and the method is principal component analysis (PCA). Principal components can be obtained by singular value decomposition of the descriptor matrix. Thus, the resulting projection is the same as in the case of LaSSI. So if the SVD is defined as in eq. (2.6), than the principal components or latent variables can be computed as

$$A = USV^T = PT^T, \tag{2.13}$$

$$P = U, \quad T = VS, \tag{2.14}$$

where $P = (p_{ik})$ is the *loading matrix* describing the influence of descriptors on latent variables.

2.3 Testing of performance

We have discussed several methods used in Ligand-Based Virtual screening, but which method is the best or which method can be improved? If we want to analyse the overall performance and compare methods between each other, we need proper rating technique and standardized data sets, same for all compared methods.

2.3.1 Performance rating

There are several evaluation methods, the widely used are Enrichment Factor (EF) and Receiver Operating Characteristic (ROC) curve. Input is a set of ranked molecules with binary information if molecule is active or decoy.

Enrichment Factor

The enrichment factor (EF) simply analyse the top rated fraction of molecules in data set. It compares number of active molecules found in the fraction and the average number of active molecules that would be found in case of uniform distribution. EF is defined as

$$EF(\phi) = \frac{\sum_{i=1}^{n} \delta(r_i)}{\phi n}, \text{ with } \delta(r_i) = \begin{cases} 1, & r_i \leq \phi N \\ 0, & r_i \leq \phi N \end{cases}, \tag{2.15}$$

where ϕ is specified fraction, usually 0.01–0.05, n is total number of active molecules, N is total number of all molecules, r_i is a rank of i-th active molecule.

Problem of EF is a large variation when a small number of actives is used. Methods that try to overcome this problem are for example Robust Initial Enhancement (RIE) or Boltzmann-Enhanced Discrimination of ROC (BEDROC) [37].

Receiver Operating Characteristic

The receiver operating characteristic was firstly used in signal detection analysis. ROC curve express dependency of sensitivity and specificity of the method [38].

ROC curve analyse a set of molecule rated and ordered by a VS method. Essentially we define a threshold and all molecules above this value are selected and the remaining molecules discarded. The active molecules in the selected set are *True Positives*, while the inactive *False Positives.* The molecules in the discarded set are similarly called *False Negatives* if they are active and *True Negatives* if they are inactive [38].

The sensitivity (Se) is a ratio of the correctly selected *True Positives* to all active molecules

$$\mathrm{Se} = \frac{n_{\mathrm{TP}}}{n_{\mathrm{TP}} + n_{\mathrm{FN}}}, \tag{2.16}$$

where n_{TP} is number of selected actives and n_{FN} number of incorrectly discarded actives.

The specificity (Sp) is the ratio of the discarded inactive compounds to all inactive molecules

$$\mathrm{Sp} = \frac{n_{\mathrm{TN}}}{n_{\mathrm{TN}} + n_{\mathrm{FP}}}, \tag{2.17}$$

where n_{TN} is number of discarded inactives and n_{FP} number of wrongly selected inactives.

Finally ROC curve plot Selectivity, Se as a function of (1-Sp). See Fig. 2.1, the closer is the curve to ideal curve, the better VS method is. The ROC curve can be integrated in order to obtain area under the curve (AUC), which is a single value performance indicator.

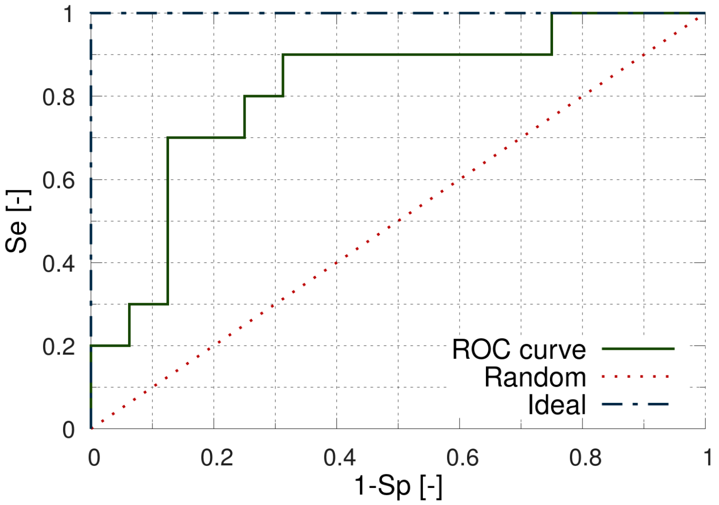


Figure 2.1: Illustration of ROC curve, with AUC=0.8.

Usage of AUC as performance indicator for VS is also criticised [37]. For instance,it struggles to identify the *early recognition* phenomenon. Early recognition means that VS method ranks actives very early in the large data set of compounds. It can be better if VS method rank half of the active molecules at the very beginning and half at the end of the list, than rank all active molecules in the middle of the list. In both cases the AUC is 0.5, but first example offers fast results in experimental testing [37].

However, AUC still does possess desirable statistical behaviors for large data sets, is easy to use, and there are results of other screening methods rated by AUC. So we have chosen to use AUC as a performance indicator in this thesis.

3. Model - general information

Goal of this thesis is the implementation of ligand-based virtual screening method utilizing the latent semantic analysis and analyse possible improvement of this method.

Virtual screening methods were implemented in Python with usage of cheminformatics libraries. Python is a very popular programming language to solve cheminformatics and bioinformatics problem. The main reason of using python is that there is a number of available scientific libraries that provide evaluating and computational tools as well as analysing, statistic and visualisation tools. Thus, new methods can be easily tested. However the main drawback is a limited performance as python is an interpreted language that does not compile the code to machine-language. This disadvantage is moderated by usage of pre-compiled c++ libraries wrapped by Python API.

3.1 Used cheminformatics libraries

3.1.1 RDKit

RDKit is an open-source cheminformatics software [39]. Core data structures and algorithms are written in C++, but there is also available a very popular Python wrapper that significantly facilitate usage of the library. Alongside Python wrapper, there are also less known Java and C# wrappers. RDKit is distributed under business-friendly BSD license, that impose minimal restrictions on the use and redistribution.

Usage of RDKit in python is transparent and straight-forward. RDKit offers a wide-ranging functionality, namely reading, writing and drawing of molecules, substructure searching, generation of structural fingerprints (AP, TT, ECFP ...), atomic descriptors, pharmacophore fingerprints and it also provide supports for working with chemical reactions.

3.1.2 Biochem-tools

Biochem-tools is a python library developed by Petr Škoda [40] that is capable of extracting fragments from molecules. It utilize module *RDKit.Chem* and it can extract fragments derived from extended connectivity fingerprints and topological torsion fingerprints. In case of ECFP, first a Morgan fingerprint is generated and subsequently it translates info of each non-zero bit to corresponding SMILES fragment. In case of TT, the process is slightly different, it uses SMARTS molecular patterns for substructure search to find all possible fragments of specified length.

Apart from fragment functionality it also simplifies generation of atomic molecular descriptors. It contain an explicit list of all descriptors and their corresponding functions that are available in RDKit.

3.2 Data sets

Before usage of the screening model, it is necessary to prepare input data sets from molecular collections. We have used 2 different collections of data sets in this thesis. First is Maximum Unbiased Validation (MUV) data set, second is a compilation of data sets with known average performance rating.

3.2.1 Maximum Unbiased Validation data set

MUV data sets are considered to be most difficult data sets for ligand-based Virtual screening because of the way how they are constructed. Thus, they are ideal for testing of the robustness of the methods.

Design of MUV Data sets comprises of three major steps. Firstly, a collection of bioassays is analysed and molecules that are found certainly active or inactive are extracted to form the initial set. Subsequently, chemical universe around each active molecule is statistically examined if it is well surrounded by decoys. All inconvenient molecules are discarded. Finally, an experimental design algorithms are applied to select subsets of 30 active and 15 000 inactive molecules with spatially random distribution regarding simple molecular properties [41].

3.2.2 Compilation of data sets

Hoksza and Škoda [42] have collected several data sets and separate them into classes according to performance of ligand-based similarity methods using structural fingerprints. Data sets are no further statistically analysed and modified unlike MUV sets. But they are arranged according to the average AUC value, there are 4 main classes 0.8–0.85, 0.85–0.9, 0.9–0.95, 0.98–1.0. These data sets are much easier for similarity methods in comparison to MUV sets and they are ideal for tweaking and initial testing of the VS method. Every data set consist of 4900 inactive molecules, total number of active molecules ranges from 100–300.

3.2.3 Data set preparation for screening

Our VS method needs a train set of active molecules. as well a target test set. In order to obtain a statistically significant results we have prepared 10 random sets from every single original data set. Always two groups of actives were randomly chosen. One served as train set and the other was mixed with inactive molecules to form a test set.

In case of MUV sets, we have chosen a train set of 20 actives uniformly at random and remaining 10 actives form together with 12000 inactives a test set.

Compilation of data sets contains larger number of active molecules, so only 30 actives were chosen for train set and 20 actives together with 4900 inactives for test set. This selection was not done in random but we have rather used a previously prepared random selections, because there are available results of VS methods that analysed exactly the same data sets and selections [33].

4. Model - implementation and results

4.1 General model description

Our model is based on Latent Semantic Structure Indexing described in Section 2.2.1 and further modified and improved.

However comparison of results previously published by Hull [35, 43] was not possible because they used data sets extracted from MDDR library [44], which is commercial and currently not accessible at our institution. Additionally Hull used an in-house designed Atom-Pairs and Topological torsion molecular descriptors that are also not accessible.

4.2 Molecule based LSI

The molecule based LSI is a reimplementation of the method described by Hull. We have also tested Atom-Pairs and Topological torsion molecular descriptors implemented in RDKit. Additionally these results were compared with ECFP and atomic descriptors that have not been tested before.

Screening process with molecule based LaSSI is as follows, assuming that we have prepared train and test set

1. Vector of descriptors is generated for each molecule in train set
2. All vectors are put into one descriptor matrix and chosen weight function is applied.
3. Latent variables are generated by singular value decomposition of descriptor matrix ($A = USV^T$).
4. Each molecule in test set is rated as
   (a) Vector $\mathbf{z}$ of descriptors is generated for tested molecule.
   (b) Chosen weight function is applied.
   (c) Vector $\mathbf{z}$ is projected to latent space, via $\mathbf{y} = S^{-1}U^T\mathbf{z}$.
   (d) Cosine similarities between projected tested molecule (vector y) and each projected training molecule (rows of matrix V) are computed.
   (e) Maximum cosine value is chosen and the value is saved to file.
5. After rating of all molecules, they are ordered by the descending similarity and the resulting performance is measured by AUC.

4.2.1 Implementation notes

Each column of the descriptor matrix belongs to one train molecule and each row to one molecular descriptor. The matrix consist only from non-zero descriptors for at least one train molecule, so rows of zeros only are not allowed.

Structural fingerprints are integer vectors describing the frequency of occurrence of a certain molecular feature.

Atomic descriptors are collection of simple structural, formula based counts and descriptors describing the physicochemical properties. From simple counts we use number of carbocycles, hetero-cycles, aliphatic rings or specific functional groups. Advanced descriptors are namely estate indices, kappa, chi indices, partition coefficients *SlogP*, molar Refractivity *SMR*, partial charge *PEOE*. All available descriptors in RDKit are listed on the web page of official documentation [45].

4.2.2 Weight functions

We have used different weight functions. In case of **structural fingerprints** *(AP, TT, ECFP)*, if no weight function is applied and frequencies are directly used, we use notation *freq*. If **integer vector** is modified and represents only binary information about occurrence of certain structural feature, we use notation *bin*. And if each frequency is divided by maximal frequency found in descriptor matrix, notation is *max*.

Atomic descriptors *(desc)* are used unmodified or normalised in two different ways:

- Division by maximal value in absolute for each descriptor separately, maximal value is determined from the train descriptor matrix, notation is *abs*.

- Standard normalization,value is shift to zero and division by the difference of maximal and minimal value for each descriptor separately, notation is *norm*.

4.2.3 Results

All results for different fingerprints and different weight functions are summarized in Fig. 4.1 and in Fig. 4.2.

In case of compilation of data sets (Fig. 4.1), binarized descriptors perform better than direct usage of frequencies. ECFP fingerprints are more effective than atom-pair or topological-torsion descriptors. Descriptors that use atomic properties of entire molecule showed the worst results regardless the weight function. The best results were obtained by utilisation of combination of ECFP2 and ECFP4. Similar trends can be observed for MUV data sets (Fig. 4.2), where the best performance shows combination of ECFP2 and ECFP6.

4.3 Fragment based LSI

Fragment based LSI is a modification where latent variables are not generated from descriptors computed from entire molecules. Instead we decompose all train molecules to fragments of certain length. The reason is, that we have noticed of poor performance of atomic descriptors on entire molecules. So we decided to test the idea if molecular fragments can describe the relationship between structure and final activity better.

		freq ecfp2	bin ecfp2	max ecfp2	bin ecfp2+4	bin ecfp2+6	bin ap	freq ap	bin tt	freq tt	abs desc	norm desc	desc
0.80-0.85	5HT2B	0.762	0.776	0.734	0.779	0.768	0.717	0.688	0.717	0.66	0.638	0.635	0.624
	5HT2C	0.749	0.784	0.73	0.792	0.787	0.73	0.706	0.773	0.708	0.668	0.667	0.634
	ADA2A	0.778	0.799	0.767	0.807	0.812	0.761	0.715	0.744	0.721	0.653	0.658	0.587
	CDK2	0.681	0.732	0.713	0.736	0.731	0.7	0.666	0.702	0.642	0.62	0.609	0.525
	HDAC01	0.686	0.699	0.693	0.727	0.724	0.699	0.656	0.689	0.612	0.53	0.538	0.593
0.85-0.90	PXR_Agonist	0.851	0.866	0.843	0.883	0.896	0.861	0.818	0.827	0.847	0.632	0.659	0.607
	ACM1_Agonist	0.756	0.769	0.713	0.769	0.768	0.73	0.7	0.701	0.688	0.664	0.656	0.593
	ADA2B_Antagonist	0.785	0.828	0.794	0.824	0.815	0.77	0.738	0.761	0.731	0.705	0.703	0.6
	ADA2C_Antagonist	0.793	0.799	0.78	0.801	0.793	0.738	0.682	0.717	0.735	0.649	0.636	0.608
0.90-0.95	CHK1	0.841	0.87	0.859	0.877	0.864	0.76	0.787	0.783	0.761	0.677	0.671	0.526
	5HT1F_Agonist	0.774	0.762	0.709	0.795	0.786	0.759	0.709	0.762	0.638	0.643	0.655	0.599
	DRD1_Antagonist	0.8	0.837	0.778	0.838	0.833	0.836	0.77	0.793	0.759	0.699	0.699	0.643
	DRD2_Agonist	0.863	0.88	0.855	0.885	0.882	0.859	0.827	0.819	0.837	0.638	0.655	0.637
	LSHR_Antagonist	0.809	0.864	0.827	0.882	0.883	0.84	0.82	0.818	0.782	0.648	0.656	0.631
0.98-1	OPRM_Agonist	0.842	0.86	0.832	0.861	0.857	0.879	0.827	0.866	0.839	0.753	0.765	0.609
	DHFR	0.931	0.914	0.884	0.947	0.956	0.804	0.764	0.73	0.673	0.469	0.551	0.604
	MTR1A_Agonist	0.747	0.77	0.712	0.799	0.797	0.846	0.81	0.731	0.706	0.669	0.673	0.551
	MTR1B_Agonist	0.793	0.822	0.76	0.835	0.826	0.872	0.82	0.747	0.709	0.624	0.637	0.604
	V2R_Antagonist	0.837	0.865	0.816	0.901	0.906	0.86	0.827	0.83	0.746	0.644	0.626	0.593
	AVG	0.794	0.816	0.779	0.828	0.825	0.791	0.754	0.764	0.726	0.643	0.65	0.598

Figure 4.1: Results of molecule based LSI - Compilation of data sets (see Section 4.3.2 for legend)

	freq ecfp2	bin ecfp2	max ecfp2	bin ecfp2+4	bin ecfp2+6	bin ap	freq ap	bin tt	freq tt	abs desc	norm desc	desc
466	0.624	0.629	0.598	0.633	0.641	0.701	0.655	0.644	0.659	0.62	0.623	0.596
548	0.65	0.687	0.662	0.684	0.687	0.652	0.676	0.629	0.604	0.628	0.581	0.621
600	0.643	0.668	0.641	0.657	0.659	0.612	0.601	0.624	0.588	0.539	0.556	0.545
644	0.755	0.711	0.736	0.723	0.722	0.727	0.737	0.663	0.693	0.659	0.642	0.626
652	0.561	0.672	0.65	0.657	0.66	0.726	0.575	0.655	0.578	0.562	0.585	0.572
689	0.554	0.681	0.64	0.687	0.681	0.607	0.518	0.549	0.636	0.539	0.476	0.573
692	0.531	0.481	0.488	0.495	0.501	0.46	0.484	0.527	0.514	0.564	0.543	0.517
712	0.621	0.546	0.578	0.535	0.546	0.63	0.621	0.603	0.54	0.631	0.619	0.528
713	0.678	0.681	0.617	0.681	0.677	0.598	0.612	0.589	0.603	0.587	0.587	0.582
733	0.582	0.656	0.64	0.651	0.651	0.626	0.632	0.625	0.557	0.549	0.541	0.532
737	0.513	0.535	0.497	0.528	0.543	0.465	0.57	0.485	0.507	0.525	0.537	0.577
810	0.646	0.688	0.703	0.726	0.735	0.62	0.624	0.588	0.562	0.625	0.642	0.53
832	0.648	0.732	0.696	0.763	0.779	0.753	0.736	0.724	0.645	0.566	0.576	0.536
846	0.652	0.67	0.632	0.695	0.707	0.677	0.741	0.727	0.679	0.622	0.636	0.567
852	0.702	0.758	0.749	0.737	0.747	0.763	0.698	0.724	0.7	0.664	0.662	0.622
858	0.652	0.625	0.585	0.649	0.652	0.655	0.62	0.563	0.603	0.656	0.667	0.609
859	0.457	0.513	0.461	0.51	0.506	0.562	0.487	0.5	0.501	0.503	0.498	0.406
AVG	0.616	0.643	0.622	0.648	0.652	0.637	0.623	0.613	0.598	0.59	0.587	0.561

Figure 4.2: Results of molecule based LSI - MUV data sets (see Section 4.3.2 for legend)

We use fragments derived from circular ECFP and TT fragments, the process of generation is described in Section 3.1.2. And fragments are compared to fragments of tested molecule via LSI.

Screening process with fragment based LSI is as follows, assuming that we have prepared train and test set

1. Chosen molecular fragments are extracted from set of train molecules.
2. Vector of molecular descriptors is generated for each unique molecular fragment.
3. All vectors are put into one training descriptor matrix and chosen weight function is applied.
4. Latent variables are generated by Singular value decomposition of descriptor matrix ($A = USV^T$).
5. Each molecule in test set is rated as
   (a) Chosen molecular fragments are extracted.
   (b) Vector $\mathbf{z}$ of molecular descriptors is generated for each unique molecular fragment.

(c) Chosen weight function is applied.

(d) Vector $\mathbf{z}$ is projected to latent space, via $\mathbf{y} = S^{-1}U^T\mathbf{z}$.

(e) Cosine similarities between projected molecular fragment (vector y) and each projected fragment from training matrix (rows of matrix V) are computed.

(f) Maximum cosine value is chosen

(g) Cosine similarity values are collected for all fragments of tested molecule

(h) Finally an average value is computed and saved to file.

6. After rating of all molecules, they are ordered by the descending similarity and the resulting performance is measured by AUC.

4.3.1 Implementation notes

Atomic descriptors are discussed in Section 4.2.1. We store a cosine similarity value in hash table for each fragment and the value is reused, if the same fragment should be processed again for different molecule.

4.3.2 Weight functions

We use weight functions for atomic descriptors described in Section 4.2.2. But the weight functions are applied on atomic descriptors generated from each molecular fragments.

4.3.3 Results

Fragment based LSI perform better than previous method (see Fig. 4.3 and Fig. 4.4). Fragments extracted via ECFP seems to be more effective than linear fragments from TT fingerprints. Increasing size of fragments have positive influence on resulting performance, however, it significantly increases computational time.

Interesting observation is that standard normalisation of descriptors have negative influence on performance. On the other hand screening using division by maximum in absolute value as weight function performs almost the same as screening with unmodified descriptors. Both Compilation of data sets and MUV data sets have the same trend.

4.4 Comparison with other methods

We compare our LSI model with results of similarity methods directly using fingerprints, namely AP, ECFP2, FCFP2, MACCS, TT (see Fig. 4.5 and Fig. 4.6). Molecule based LSI performs worse than all other fingerprint methods.

On the other hand, fragment based LSI have comparable results with most fingerprint methods on compilation of data sets. It surpasses fingerprints for some more difficult data sets but lacks of performance for the less difficult sets.

		f abs ecfp2	f ecfp2	f norm ecfp2	f abs ecfp3	f abs tt2	f abs tt4	f abs tt6
0.80-0.85	5HT2B	0.862	0.862	0.562	0.845	0.692	0.794	0.847
	5HT2C	0.889	0.889	0.664	0.863	0.71	0.837	0.888
	ADA2A	0.902	0.903	0.584	0.892	0.785	0.876	0.899
	CDK2	0.836	0.836	0.551	0.815	0.584	0.78	0.817
	HDAC01	0.821	0.821	0.568	0.849	0.604	0.767	0.816
0.85-0.90	PXR_Agonist	0.878	0.879	0.695	0.879	0.764	0.845	0.896
	ACM1_Agonist	0.845	0.848	0.803	0.819	0.765	0.818	0.833
	ADA2B_Antagonist	0.912	0.912	0.605	0.902	0.797	0.873	0.895
	ADA2C_Antagonist	0.907	0.909	0.698	0.901	0.733	0.855	0.886
0.90-0.95	CHK1	0.895	0.896	0.561	0.891	0.643	0.804	0.889
	5HT1F_Agonist	0.947	0.946	0.491	0.957	0.772	0.887	0.921
	DRD1_Antagonist	0.934	0.934	0.853	0.921	0.8	0.872	0.897
	DRD2_Agonist	0.93	0.929	0.566	0.916	0.7	0.862	0.884
	LSHR_Antagonist	0.903	0.903	0.485	0.918	0.659	0.833	0.89
0.98-1	OPRM_Agonist	0.95	0.95	0.649	0.964	0.787	0.921	0.942
	DHFR	0.993	0.993	0.968	0.988	0.919	0.987	0.987
	MTR1A_Agonist	0.971	0.973	0.967	0.977	0.731	0.904	0.942
	MTR1B_Agonist	0.976	0.976	0.974	0.961	0.753	0.931	0.949
	V2R_Antagonist	0.987	0.986	0.674	0.989	0.766	0.916	0.967
	AVG	0.913	0.913	0.68	0.908	0.735	0.861	0.897

Figure 4.3: Results of fragment based LSI - Compilation of data sets (see Section 4.3.2 for legend)

	f abs ecfp2	f ecfp2	f norm ecfp2	f abs ecfp3	f abs tt2	f abs tt4	f abs tt6
466	0.646	0.646	0.575	0.695	0.516	0.547	0.599
548	0.744	0.743	0.508	0.773	0.488	0.62	0.677
600	0.717	0.717	0.666	0.72	0.52	0.613	0.635
644	0.791	0.791	0.547	0.805	0.467	0.675	0.681
652	0.548	0.549	0.534	0.558	0.457	0.468	0.504
689	0.551	0.549	0.54	0.631	0.463	0.546	0.531
692	0.576	0.575	0.582	0.566	0.576	0.502	0.501
712	0.645	0.643	0.47	0.569	0.456	0.495	0.594
713	0.571	0.573	0.493	0.562	0.534	0.51	0.527
733	0.604	0.604	0.574	0.571	0.435	0.611	0.587
737	0.648	0.648	0.59	0.614	0.535	0.53	0.594
810	0.742	0.74	0.45	0.729	0.489	0.607	0.686
832	0.882	0.882	0.546	0.862	0.605	0.804	0.86
846	0.91	0.91	0.768	0.912	0.651	0.805	0.846
852	0.767	0.767	0.542	0.77	0.497	0.645	0.735
858	0.601	0.601	0.504	0.58	0.51	0.538	0.599
859	0.517	0.517	0.471	0.498	0.47	0.443	0.501
AVG	0.674	0.674	0.551	0.671	0.51	0.586	0.627

Figure 4.4: Results of fragment based LSI - MUV data sets see Section 4.3.2 for legend)

In the case of MUV data sets the performance is worse than almost all fingerprints except the MACCS.

4.5 Application structure

The application structure consists of several python modules, see Fig. 4.7. Application is controlled via command line user interface located in *green* main module. All internal *red* modules form computational core of application. External *orange* library is used for generation of molecular fragments. Source code is available on

		f abs ecfp2	bin ecfp2+4	ecfp1-2	AP	ECFP2	FCFP2	MACCS	TT
0.80-0.85	5HT2B	0.86	0.78	0.83	0.78	0.8	0.8	0.82	0.82
	5HT2C	0.89	0.79	0.85	0.85	0.85	0.85	0.85	0.86
	ADA2A	0.9	0.81	0.92	0.84	0.83	0.84	0.86	0.83
	CDK2	0.84	0.74	0.87	0.83	0.79	0.75	0.8	0.86
	HDAC01	0.82	0.73	0.94	0.81	0.83	0.78	0.76	0.83
0.85-0.90	PXR_Agonist	0.88	0.88	0.95	0.83	0.85	0.83	0.87	0.86
	ACM1_Agonist	0.84	0.77	0.91	0.87	0.89	0.9	0.9	0.85
	ADA2B_Antagonist	0.91	0.82	0.92	0.88	0.88	0.88	0.89	0.86
	ADA2C_Antagonist	0.91	0.8	0.94	0.87	0.88	0.86	0.89	0.87
0.90-0.95	CHK1	0.9	0.88	0.96	0.91	0.9	0.85	0.86	0.96
	5HT1F_Agonist	0.95	0.8	0.95	0.94	0.94	0.93	0.95	0.94
	DRD1_Antagonist	0.93	0.84	0.96	0.94	0.94	0.94	0.95	0.94
	DRD2_Agonist	0.93	0.88	0.97	0.95	0.95	0.9	0.96	0.96
	LSHR_Antagonist	0.9	0.88	0.89	0.93	0.93	0.91	0.92	0.96
0.98-1	OPRM_Agonist	0.95	0.86	0.96	0.95	0.96	0.93	0.93	0.97
	DHFR	0.99	0.95	1	0.98	0.99	0.99	1	1
	MTR1A_Agonist	0.97	0.8	0.99	0.99	0.99	0.96	0.98	0.99
	MTR1B_Agonist	0.98	0.83	0.99	0.99	0.99	0.96	0.99	0.99
	V2R_Antagonist	0.99	0.9	0.98	0.98	0.98	0.97	0.98	0.99
	AVG	0.91	0.83	0.94	0.9	0.9	0.89	0.9	0.91

Figure 4.5: Comparison of the LSI model to standard fingerprints on compilation of data sets

	f abs ecfp2	bin ecfp2+6	AP	ECFP2	FCFP2	MACCS	TT
466	0.65	0.64	0.66	0.59	0.59	0.57	0.66
548	0.74	0.69	0.87	0.83	0.8	0.63	0.8
600	0.72	0.66	0.75	0.76	0.72	0.65	0.77
644	0.79	0.72	0.84	0.8	0.77	0.7	0.84
652	0.55	0.66	0.71	0.68	0.58	0.49	0.7
689	0.55	0.68	0.68	0.67	0.64	0.56	0.81
692	0.58	0.5	0.6	0.54	0.53	0.53	0.6
712	0.65	0.55	0.74	0.7	0.6	0.62	0.77
713	0.57	0.68	0.69	0.69	0.67	0.5	0.67
733	0.6	0.65	0.71	0.64	0.71	0.53	0.68
737	0.65	0.54	0.77	0.74	0.75	0.7	0.7
810	0.74	0.73	0.73	0.67	0.64	0.65	0.79
832	0.88	0.78	0.86	0.84	0.82	0.85	0.88
846	0.91	0.71	0.9	0.89	0.83	0.81	0.9
852	0.77	0.75	0.86	0.8	0.8	0.81	0.81
858	0.6	0.65	0.67	0.64	0.62	0.6	0.68
859	0.52	0.51	0.55	0.58	0.67	0.61	0.56
AVG	0.67	0.65	0.74	0.71	0.69	0.64	0.74

Figure 4.6: Comparison of the LSI model to standard fingerprints on MUV data sets

github [46] or in attachments.

General screening begins with generation of train and test set. Selections can be prepared in advance, supported format is the same as generated by benchmarking platform created by Škoda and Hoksza [42]. It consists of molecule database in sdf format and library of json files that describes each selection. But it is possible to also create new random selections, where supported is currently only MUV database in the format described by Rohrer [41].

Each selection consists of three files in SMILES format, where test set is saved in *data.smi*, train set in *known-ligands.smi* and set of all active molecules for later evaluation in *ligands.smi*.

After evaluation, results are saved into *result.csv* in format: ”cosine similarity, SMILES”. Sorted in *result_sorted.csv*, active and inactive molecules are marked in *result_stat_data.csv* and results are saved to *result_stat.csv*.

Final results of screening of all random selections for a particular data set are collected to a separate file in root directory.

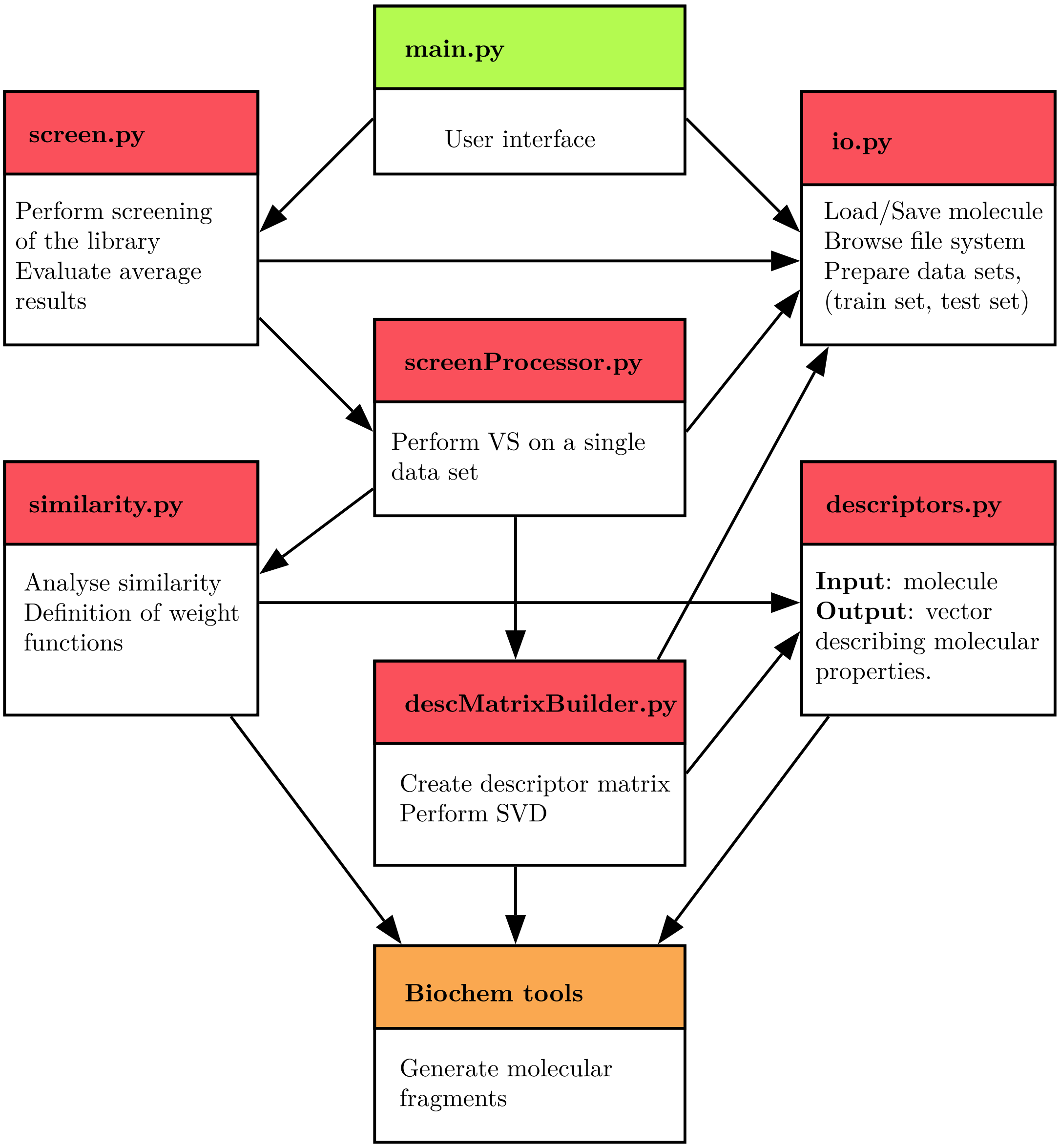


Figure 4.7: Architecture of the application for VS based on LSI

Conclusion

We have examined existing VS method called latent semantic structural indexing (LaSSI) [35], based on description of entire molecule by structural fingerprints AP or TT and we have compared it with ECFP fingerprints. Usage of ECFP fingerprints leads to better performance. We have also tested different weight functions that modify original frequencies of structural features. And usage of only binary information about presence of certain structural feature is the most effective.

We have also tested usage of formula based counts and descriptors describing the physicochemical properties, however, descriptors generated on entire molecules exhibit poor performance.

Finally, we have developed a new model for virtual screening (VS) based on latent semantic indexing combining both structural fingerprints and atomic descriptors. We call this method a Fragment based LSI. Each molecule is represented by molecular fragments of certain length and each fragment is described by atomic descriptors. The extracted fragments from tested molecule are than compared to all fragments that are present in train set of active molecules.

The extraction of fragments based on ECFP seems to be better than TT fingerprints and ECFP with radius of 2 bonds have the best performance. Atomic descriptors were the most effective unmodified or scaled by maximal absolute value for each descriptor.

Fragment based LSI shows far better performance than molecule based LaSSI and it is comparable with other fingerprint methods. However, the computational difficulty is increased because it is necessary to evaluate each fragment separately for each molecule.

Further improvement of the fragment based LSI could be probably possible by identification of structural features, responsible for observed activity. Identified fragments would be favoured during molecule rating.

Bibliography

[1] R. C. Read and D. G. Corneil, “The graph isomorphism disease,” *Journal of Graph Theory*, vol. 1, no. 4, pp. 339–363, 1977.

[2] A. R. Leach and V. J. Gillet, *An introduction to chemoinformatics.* Springer Science & Business Media, 2007.

[3] H. 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.

[4] D. Weininger, A. Weininger, and J. L. Weininger, “Smiles. 2. algorithm for generation of unique smiles notation,” *Journal of Chemical Information and Computer Sciences*, vol. 29, no. 2, pp. 97–101, 1989.

[5] Y. C. Martin and R. S. DeWitte, *Hydrophobicity and solvation in drug design*, vol. 19. Kluwer/Escom, 2000.

[6] L. B. Kier, L. H. Hall, *et al.*, *Molecular connectivity in structure-activity analysis.* Research Studies, 1986.

[7] L. H. Hall and L. B. Kier, “Reviews in computational chemistry,” *The Molecular Connectivity Chi Indexes and Kappa Shape Indexes in Structure-Property Modeling*, pp. 367–422, 1991.

[8] L. H. Hall and L. B. Kier, “Electrotopological state indices for atom types: a novel combination of electronic, topological, and valence state information,” *Journal of Chemical Information and Computer Sciences*, vol. 35, no. 6, pp. 1039–1045, 1995.

[9] 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.

[10] 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.

[11] 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.

[12] D. Rogers and M. Hahn, “Extended-connectivity fingerprints,” *Journal of chemical information and modeling*, vol. 50, no. 5, pp. 742–754, 2010.

[13] R. S. Pearlman and K. Smith, “Novel software tools for chemical diversity,” *Perspectives in Drug Discovery and Design*, vol. 9, no. 11, pp. 339–353, 1998.

[14] P. Kirkpatrick and C. Ellis, “Chemical space,” *Nature*, vol. 432, no. 7019, p. 823, 2004.

[15] D. Wilton, P. Willett, K. Lawson, and G. Mullier, “Comparison of ranking methods for virtual screening in lead-discovery programs,” *Journal of chemical information and computer sciences*, vol. 43, no. 2, pp. 469–474, 2003.

[16] I. Halperin, B. Ma, H. Wolfson, and R. Nussinov, “Principles of docking: An overview of search algorithms and a guide to scoring functions,” *Proteins: Structure, Function, and Bioinformatics*, vol. 47, no. 4, pp. 409–443, 2002.

[17] P. Willett, J. M. Barnard, and G. M. Downs, “Chemical similarity searching,” *Journal of chemical information and computer sciences*, vol. 38, no. 6, pp. 983–996, 1998.

[18] O. F. Güner, *Pharmacophore perception, development, and use in drug design*, vol. 2. Internat’l University Line, 2000.

[19] P. Gedeck and P. Willett, “Visual and computational analysis of structure–activity relationships in high-throughput screening data,” *Current opinion in chemical biology*, vol. 5, no. 4, pp. 389–395, 2001.

[20] P. D. Lyne, “Structure-based virtual screening: an overview,” *Drug discovery today*, vol. 7, no. 20, pp. 1047–1055, 2002.

[21] M. Wójcikowski, P. J. Ballester, and P. Siedlecki, “Performance of machine-learning scoring functions in structure-based virtual screening,” *Scientific Reports*, vol. 7, 2017.

[22] M. A. Johnson and G. M. Maggiora, *Concepts and applications of molecular similarity.* Wiley, 1990.

[23] C. Lemmen and T. Lengauer, “Computational methods for the structural alignment of molecules,” *Journal of Computer-Aided Molecular Design*, vol. 14, no. 3, pp. 215–232, 2000.

[24] L. Leyda, R. Carbo, and M. Arnau, “An electron density measure of the similarity between two compounds,” *Intl. J. Quantum Chem*, vol. 17, pp. 1185–1189, 1980.

[25] D. Plewczynski, S. A. Spieser, and U. Koch, “Performance of machine learning methods for ligand-based virtual screening,” *Combinatorial chemistry & high throughput screening*, vol. 12, no. 4, pp. 358–368, 2009.

[26] G. Schneider and P. Wrede, “Artificial neural networks for computer-based molecular design,” *Progress in biophysics and molecular biology*, vol. 70, no. 3, pp. 175–222, 1998.

[27] J. Taskinen and J. Yliruusi, “Prediction of physicochemical properties based on neural network modelling,” *Advanced drug delivery reviews*, vol. 55, no. 9, pp. 1163–1183, 2003.

[28] T. Kohonen, “Self organising and associative memory,” *Springer Series on Information Sciences (Springer-Verlag)*, 1989.

[29] L. Breiman, “Bagging predictors,” *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[30] Y. Freund, “Boosting a weak learning algorithm by majority,” *Information and computation*, vol. 121, no. 2, pp. 256–285, 1995.

[31] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[32] E. O. Cannon, A. Amini, A. Bender, M. J. Sternberg, S. H. Muggleton, R. C. Glen, and J. B. Mitchell, “Support vector inductive logic programming outperforms the naive bayes classifier and inductive logic programming for the classification of bioactive chemical compounds,” *Journal of computer-aided molecular design*, vol. 21, no. 5, pp. 269–280, 2007.

[33] D. Hoksza and P. Škoda, “Using bayesian modeling on molecular fragments features for virtual screening,” in *Computational Intelligence in Bioinformatics and Computational Biology (CIBCB), 2016 IEEE Conference on*, pp. 1–6, IEEE, 2016.

[34] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods.* Cambridge university press, 2000.

[35] R. D. Hull, S. B. Singh, R. B. Nachbar, R. P. Sheridan, S. K. Kearsley, and E. M. Fluder, “Latent semantic structure indexing (lassi) for defining chemical similarity,” *Journal of medicinal chemistry*, vol. 44, no. 8, pp. 1177–1184, 2001.

[36] S. Wold, A. Ruhe, H. Wold, and W. Dunn, III, “The collinearity problem in linear regression. the partial least squares (pls) approach to generalized inverses,” *SIAM Journal on Scientific and Statistical Computing*, vol. 5, no. 3, pp. 735–743, 1984.

[37] 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.

[38] N. Triballeau, F. Acher, I. Brabet, J.-P. Pin, and H.-O. Bertrand, “Virtual screening workflow development guided by the “receiver operating characteristic” curve approach. application to high-throughput docking on metabotropic glutamate receptor subtype 4,” *Journal of medicinal chemistry*, vol. 48, no. 7, pp. 2534–2547, 2005.

[39] G. Landrum, “Rdkit.” `https://github.com/rdkit/rdkit`. Accessed: 2017-1-6.

[40] P. Škoda, “Biochem-tools.” `https://github.com/skodapetr/biochem-tools`. Accessed: 2017-1-6.

[41] 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.

[42] P. Škoda and D. Hoksza, “Benchmarking platform for ligand-based virtual screening,” in *Bioinformatics and Biomedicine (BIBM), 2016 IEEE International Conference on*, pp. 1220–1227, IEEE, 2016.

[43] R. D. Hull, E. M. Fluder, S. B. Singh, R. B. Nachbar, S. K. Kearsley, and R. P. Sheridan, “Chemical similarity searches using latent semantic structural indexing (lassi) and comparison to toposim,” *Journal of medicinal chemistry*, vol. 44, no. 8, pp. 1185–1191, 2001.

[44] BIOVIA and T. Reuters, “Mddr library.” `http://accelrys.com/products/collaborative-science/databases/bioactivity-databases/mddr.html`. Accessed: 2017-1-6.

[45] G. Landrum, “Rdkit - module descriptors.” `http://www.rdkit.org/Python_Docs/rdkit.Chem.Descriptors-module.html`. Accessed: 2017-1-6.

[46] J. Kolář, “Source code of screening application.” `https://github.com/kolarji2/virtsc_lassi`. Accessed: 2017-1-6.