Internal Report 2013–03 March 2013

Universiteit Leiden
Opleiding Informatica

An evolutionary algorithm for finding diverse sets of molecules with user-defined properties

Benjamin van der Burgh

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

The multidisciplinary field of drug discovery deals with the discovery and synthesis of novel medications. For over half a century, Computer Science has aided chemists in exploring the extremely large *chemistry space* that comprises the set of all drug-like molecules. But only in recent years, the technological advances in computational power and the development of novel algorithms have enabled researchers to start experimenting with *in silico* screening of promising compounds. While many classes of algorithms have been successfully applied in this field, one of the most prevalent is the class of *Evolutionary Algorithms (EA)*. This nature inspired optimization algorithm, which will be discussed in this thesis, allows for the exploration of large search spaces with the goal of finding good solutions to a given problem. Although the classical EA require maintaining diversity of population, they do not necessarily have the goal of finding a diverse set of solutions. This thesis discusses the application of the novel *Evolutionary Level-Set Algorithm (ELSA)* for finding not only a good set of solutions, but also diverse solutions, and application of this algorithm for search of drug-like molecules given certain constraints on these molecules. This diverse set can be explored by the chemist to enhance creativity and provide a starting point for further research. For the comparison of molecules with respect to their quality and to measure similarity/diversity, a proper metric is needed. The ELSA algorithm, along with five different diversity indicators were implemented as an extension module in *METool (Molecular Evolutionary Tool).* Also, experiments were conducted to test the performance of ELSA and the various diversity indicators in the context of finding diverse sets of molecules with user-defined constraints. The results of the experiments show that the simple measures, despite being of low computational complexity, perform surprisingly well when used as a quality indicator in the ELSA algorithm.

Contents

Chapter 1

Introduction

This chapter provides the reader with some basic background information on the problem domain. This thesis is focused on the application of techniques from computer science to the field of chemistry, which fits in a relatively new interdisciplinary branch of science called *chemoinformatics.* The following section starts with an overview of this science and then provides an idea of the drug discovery pipeline.

1.1 A brief history of chemoinformatics

Computers have been used in the field of chemistry for many decades. In recent years there has been a shift in the application of computers as a data processing tool towards a drug discovery tool. In the 1970s, for example, the word *informatics* was used to describe the field that dealt with the collection, storage, processing, and interpretation of data. Hippe [10] described the field of *chemical informatics* as follows.

> *Thus we may say about chemical informatics, i.e., informatics as applied to certain principles in chemical sciences, that the main areas are: (1) numerical calculations connected with mathematical models of processes and/or systems, (2) data processing, and (3) computer-controlled automation of chemical processes or unitary operations in the chemical industry.*

From this definition it can be seen that the emphasis was not in drug discovery, but rather on the descriptive and predictive modeling of large chemical systems in the industry. In the 1990s Frank Brown came up with a much broader definition that put the emphasis on drug discovery, and named it *chemoinformatics* [4]. He defined it as follows:

> *Chemoinformatics is the mixing of those information resources to transform data into information and information into knowledge for the intended purpose of making better decisions faster in the area of drug lead identification and optimization.*

Today the field encompasses applications that come from bioinformatics, data mining, optimization and many more sciences. Though chemoinformatics might be less well-known than bioinformatics, it has a considerable history. The difference is that bioinformatics has a focus on sequence data, where chemoinformatics focuses on structure information of small molecules.

1.2 Drug discovery process

This section will give an introduction to the process of drug discovery. The goal is to inform the reader of the different steps in the drug discovery pipeline, and to give an impression of the role of chemoinformatics. First, a broad overview of the target-to-lead or lead discovery phase is given, followed by the discussion of various methods to find such leads. The last section explains the need for diversity optimization and where it fits in the drug discovery pipeline.

1.2.1 Target identification

The human DNA consists of approximately 20.000 to 25.000 genes, which make up for a total of 3 billion base pairs. Each gene encodes a protein and each protein has its own function in the complex human body. Proteins can, for example, catalyze biochemical reactions, which regulate metabolism. They play a role in cell signaling, immune responses, digestion and basically every process.

Proteins also interact with each other, which can in turn catalyze interactions with other proteins, creating complex cascades of molecular reactions, called *chemical pathways*. Knowing how these pathways work is an essential step in the process of drug discovery, since diseases are often the consequence of an abnormality along it. The process of finding one of the proteins in the chemical pathway that is responsible for the disease is called the *target identification*. Such a protein has to be “drugable”, i.e., it must be able to potentially interact with and be affected by another molecule, called the *ligand*.

1.2.2 Target validation

Next, researchers must confirm that the identified target is a relevant link in the chemical pathway. This process of demonstrating that a molecular target is therapeutically relevant, is called the *target validation*. Experiments are performed in living cells to elucidate the role the protein has in the disease and to gain insight in its workings. This does not necessarily mean that the mechanism is fully understood, but it aids in the creation of a model of the target.

To find active compounds that react with the target, researchers can use a technology called *High-Throughput Screening* (HTS). In this process, a machine is loaded with a solution of biological matter of experimental interest and many different compounds, each in its own well. The reaction is observed, and if there is an interesting response, the experiment is referred to as a *hit*. A difficult challenge herein is to select the compounds based on the reaction caused by the compound and the reaction of the compound with other biochemicals. The selected molecules of interest are called *lead compounds* and are selected for further study. By using HTS, up to 100.000 different compounds can be tested per day, which makes HTS one of the most powerful tools in the drug discovery process.

Another method for finding lead compounds is called *de novo* design. In contrast to HTS, de novo design tries to search the chemical space *in silico* (virtually) rather than *in vitro* (physically). The three-dimensional shape of either known ligands or the target can be used to define shape constraints that a search algorithm can use to score the generated structures. These are called the *primary target constraints*. The *secondary target constraints* include submolecular physical and chemical properties, and constraints on the ligand-receptor interaction in the form of interaction sides, which are typically made up of hydrogen-bond acceptor and donor interactions [21]. All these constraints make for a highly complex multi-objective model of the target and a search space which is subject to combinatorial explosion. The number of chemically feasible, drug-like molecules is estimated to well exceed 10^{60} [3], thus it is impossible to consider all of them. Instead, a *stochastic combinatorial search*, such as an *evolutionary algorithm*, is performed on the chemical space. This approach is of particular interest in this thesis and will be discussed throughout Chapters 2 and 3.

1.2.3 Lead optimization

Once the target has been validated and a lead compound has been found, the next step is to *optimize* this lead. “Optimizing”, in this sense, means that the derived compound is modified in order to, for example, reduce off-target activities (side-effects) and improve the potency and pharmacokinetic properties of the lead, such as solubility. This optimization is accomplished through chemical modification of the lead structure, which is either led by the structural information on the target (target-led) or by using information about known ligands.

1.3 Multi-objective optimization

The optimization of lead compounds has been approached as a *single objective optimization problem* (SOOP) for many years, even though many different objectives were considered, but the optimizations were done sequentially, for instance by means of a weighted-sum-of-objective-functions method [18], which takes the following form:

$$f(x) = w_1 \cdot (Objective_1) + w_2 \cdot (Objective_2) + \ldots + w_n \cdot (Objective_n),$$

where $f(x)$ is a fitness function valuating the quality of the solution x, consisting of n components/objectives.

It is often desired to search for solutions, or molecules, that satisfy certain conditions. These conditions are often properties of the solution in the objective space and are defined by *constraints*. These are described by a set of formulas g_j as *equalities* or *inequalities*. Including such constraints is a very useful tool in guiding the search towards a specific area of interest or to exclude unwanted solutions. Kruisselbrink et al [12]. suggested an approach where not only the objective, in the form of *desirability functions*, but also the constraint functions are aggregated. This yields an optimization problem with only two objectives, one aggregating objectives and another aggregating constraints, which can be reduced even further to a single-objective optimization problem if all functions are aggregated as follows:

$$f = f_1 \cdot f_2 \cdot ... \cdot f_N \cdot g_1 \cdot g_2 \cdot ... \cdot g_M$$

Optimizing the different objectives in this manner results in only one solution, since the weights are fixed *a priori*. This makes it harder to consider alternatives and furthermore, assigning proper weights can be a difficult task. In the case of, for example, the optimization of ADMET[1] properties of a chemical, we would like to find those solutions that are optimal in at least one objective. Finding such solutions is tedious with the aggregation method, but are easier with specialized *multi-objective optimization methods* (MOOP). These methods try to find compromises and trade-offs by optimizing different objectives simultaneously and without deciding on the order of importance or weights of objectives *a priori*.

1.4 Drug discovery and diversity

Pharmaceutical companies keep a large collection of compound libraries, called *combinatorial libraries*, that are used in the HTS process and can sometimes contain over a million of different compounds. As the purchase and synthesis of these compounds can be very expensive, it is desirable to maintain a library that is very diverse, so that it can be used in random HTS campaigns. Also, a subset of this library has to be selected from the available compounds to create a screening library that is targeted towards the biochemical target in the screening process. In this case, diversification helps to explore the biochemical space on a large scale rather than a space that was biased towards a certain compound. Merely optimizing diversity on the molecular structure has shown however that this can lead to molecules that possess undesirable physicochemical properties that make them less suitable as a drug lead [16]. It is therefore required to take the physicochemical properties of the molecules into account and include them as constraints in the selection process. The substituents that satisfy these conditions are known as *feasible*.

Another application of diversification in drug research can be found in de novo computer applications such as the *Molecule Evoluator* [13], which implements an interactive way of searching. In this software tool, the user is first presented with a set of random molecules. After that, the most interesting molecules are manually selected by applying rules from chemistry and implicit knowledge. The chemist thus effectively operates as a “fitness function” for the program. In this case, maintaining diversity can be helpful for generating diverse sets of molecules in each step, therefore covering a larger area of interest.

In this study, METool, which is developed by researchers of the Leiden / Amsterdam Center for Drug Research (Prof. Ad IJzerman), Leiden University (Maartje van der Sar) and the Leiden Institute for Advanced Computer Science (Johannes Kruisselbrink), is used. METool has an evolutionary approach that is very similar to the Molecule Evoluator, although the focus here is on diversity optimization, which is not part of the Molecule Evoluator.

[1] ADMET is an abbreviation in pharmacology for absorption, distribution, metabolism, excretion and toxicity, and describes the tendency of a pharmaceutical compound to interact with an organism.

Chapter 2

Molecular diversity

The aim of this thesis is to find an answer to the following overarching question: how can we apply diversity-based search in drug discovery? In the previous chapter a broad description of the drug discovery process was given, and it was explained why the notion of diversity is useful here. This chapter elaborates on a more specific definition of diversity and other facets that come into play in the context of drug discovery. The following sections split the question into three sub-questions, namely:

- What is diversity and how is it measured?
- What are the usual constraints and how are they specified?
- What are speed-dynamics in this context?

2.1 Comparison of diversity in biological species and chemical compound libraries

One of the diversity measures that will later be discussed is called the Solow-Polasky measure [23], which comes from biology. In their article, Solow and Polasky explain the need for such a measure in the context of species conservation programs. They state that the extinction of any species comes with a cost, and since there is only a limited amount of resources, it should be decided which species contribute the most to biological diversity. They assume that if two species are very similar in genetic code, then they are likely to possess the same characteristics and one of them does not contribute much to the diversity of the population, i.e., the species are partially redundant. In contrast, if species are very dissimilar then each one of them contributes a lot to the diversity of the population. Since it is unlikely that all of the species useful properties are known *a priori*, it is better to keep highly dissimilar population of species, as the number of preserved properties is more likely to grow this way.

This idea is analyzed in more formal framework in Section 4.3, but what should be noted at this point is that the contribution of a species to the diversity of a population can be described in terms of its added value, expressed by its dissimilarity to other members of the population. This same idea transfers over to combinatorial libraries of chemical compounds. As stated in the previous chapter, it is desirable to cover as much of the chemical space as possible, but this can be expensive and time-consuming. The focus, therefore, is on picking those compounds that have the least overlap in their properties. This allows the creation of diverse libraries consisting of only limited compounds.

Another approach, which is utilized in this study, is to do the inverse. Instead, the focus is on compounds that are partially redundant, i.e., contribute less to the diversity of the population based on some dissimilarity function. These compounds can then safely be removed from population, while preserving the diversity of the population as much as possible. A population is first initialized with randomly generated molecules. Next, a new molecule is added, followed by the removal of the molecule that contributes the least to the diversity of the population. This approach is used in the ELSA algorithm that is the subject of Chapter 4.

2.2 Constraints on the properties of the molecule

Searching for a diverse set of molecules would not in itself be of any practical use, as there are practically infinitely many molecules. Such a subset would include molecules of many different types and posses many unrelated properties and potential applications. To guide the search process it is desirable to define constraints that describe the "family" of molecules that is searched for. These constraints can be stated on the *physicochemical properties* of the molecule, such as solubility or the number of rings. It is also possible to include constraints on particular external information about the molecule, such as the *expected costs* involved in its synthesis, or whether the molecule is *proprietary*. Only molecules that match these constraints are considered, which are called feasible molecules.

The inclusion of such constraints can be made part of the quality indicators in such a way that a replacement of an infeasible by a feasible molecule is always beneficial. The indicators that include such penalties are called *augmented quality indicators* [6] and are further discussed in Chapter 4.

2.3 Speed dynamics of the indicators

The third key goal of this thesis is to study the various quality indicators for measuring the performance of a set of alternatives that have been proposed throughout the years. The different indicators, ranging from very simple to complex, are compared in their ability to efficiently create diverse set of molecules. These indicators are described in detail in Chapter 4 with the experiments included in Chapter 6.

Chapter 3

Methodology

This chapter discussed the tools and techniques that were used to evolve populations of molecules in METool. In order to work with molecular structures within computer software, it is required to have a suitable virtual representation, which is discussed in Section 3.1. Furthermore, the evolutionary algorithm, that is used, searches the chemistry space by changing intermediate solutions by means of *mutation*, which is discussed in Section 3.2. After that, the notion of *(dis)similarity* based on molecular fingerprints, is introduced in Section 3.4. This allows us to measure the distance between pairs of molecules and utilize it to compute the diversity of the population.

3.1 Molecular representations

Chemoinformatics has spawned many different approaches to the virtual representation of molecules within software. Depending on the domain of interest, one has to decide which representation is best suited for the problem at hand. The interested reader can take a look at Brown's "An Introduction to Chemoinformatics" [5] for an overview of these molecular representations. Many commercial packages use either the line notation of SMILES [2] or the connectivity table for molecular representation. These methods are well suited for applications that contain static models, e.g. for graphical views of molecules and chemical descriptor computation purposes. In METool, the molecular structures will dynamically change as they are constantly evolved by the algorithm.

Many of the algorithms for modifying molecules and computing chemical properties come down to solving graph problems. It is therefore most natural to represent molecular structures as a set of nodes (the atoms) and a set of edges (the bonds), as is most commonly used in chemistry. This is also the representation used in METool, in which we the experiments of Chapter 5 were implemented.

3.2 Mutation of molecules

In order to evolve the molecules over time, one needs a method that allows to modify the molecular structures in a systematic way. This modifying of molecules is called *mutation*, in the context of evolutionary algorithms. METool

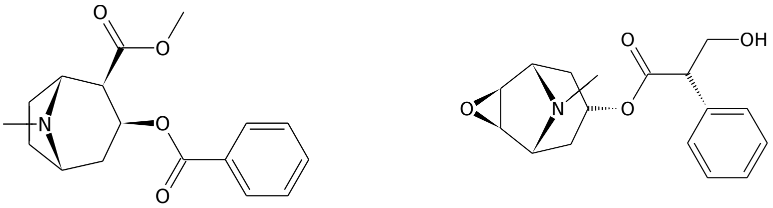


Figure 3.1: Cocaine (left) and scopolamine (right) share the same chemical formulas ($C_{17}H_{21}NO_4$), but have different structural formulas and properties.

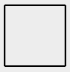
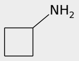
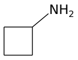
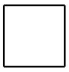
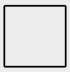
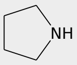
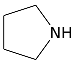
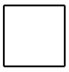
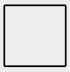
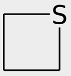
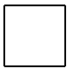
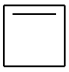
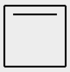
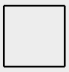
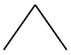
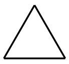
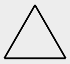
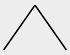
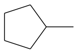
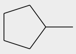
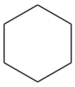
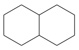
has twelve different ways to perform a mutation on a molecule. These operators will be described in the next few paragraphs and a graphical overview will be provided in the table below. The mutation operators are very similar to those originally used in the *Molecule Evoluator* [13], with the exception of a few subtle changes.

3.3 Extended-connectivity fingerprints

In the 1960s the American Chemical Society created a database with information on chemical compounds, called the Chemical Abstracts Service (CAS). As in every database, each entry needs a unique identifier. This, however, is not as trivial for the chemical compounds. Firstly, one has to decide on the representation of the compounds, which can be one of many, e.g.: as a linear (SMILES, molecular formula) or as 2-D or 3-D geometric representations. Secondly, some representations have a problem in the sense that they are ambiguous, rendering them unsuitable for the use of unique identifiers. An example of such ambiguity is given in Figure 3.1, where two molecules share a molecular formula, but are very dissimilar in their structure and properties. This ambiguity is a common problem in graph theory, where it is referred to as the *graph isomorphism problem*:

Definition 1 (Graph isomorphism)**.**
Two graphs G_1 and G_2 are isomorphic if there exists a one-to-one mapping f of the labels of vertices of G_1 onto the labels of vertices of G_2 such that adjacency is preserved, i.e., two vertices in G_1, are adjacent if and only if the corresponding vertices of G_2 are adjacent [19].

Morgan proposed an algorithm [17] for finding unique identifying labels for molecules in 1965. This algorithm exploits the fact that the molecular graph is a particular class of graphs, and as such, each vertex of the graph is numbered with an atomic number. Each non-hydrogen atom is initially assigned a numerical *identifier* equal to the degree, which is the number of non-hydrogen neighbors (see Figure 3.2a). Next, for each atom, the identifiers of the neighboring atoms are summed iteratively until each atom in the structure has a number that is as unique as possible (see Figure 3.2b). Finally, a graph walk is done, starting at the atom with the highest identifier, to which the number 1 is assigned. The neighbor with the second highest identifier is assigned 2, and so on (see Figure 3.2c). There is a tie when two neighbors have the same identifiers. In

Operator	Description	Pre	Post
Replace atom	An atom is selected randomly from the molecular structure and replaced with another random atom.[2]		
Remove atom	An atom that has only one non-hydrogen atom as its neighbor is removed from the molecular structure.		
Insert atom	The program first searches for a bond of order one and then adds to it an atom with valence 2 or higher.		
Uninsert atom	The program first creates a list of atoms in the molecular structure that have exactly two non-hydrogen neighbors and then randomly removes one from this list. A new bond is created between the neighbors of the removed atom to make the graph connected.		
Mutate atom	The program selects a non-hydrogen atom from the molecular structure and tries to find an atom with at least the same valence. If one is found, the element is mutated into the found element. If no element can be found, the graph is not changed.		
Increase bond order	The program searches for a bond of order 1 or 2 and increases it to a bond of order 2 or 3 respectively.		
Decrease bond order	The program searches for a bond of order 2 or 3 and decreases it to a bond of order 1 or 2 respectively.		
Create ring	Closes a ring by inserting a bond between two atoms that did not have a bond before.		
Break ring	Breaks a ring by removing a bond between two atoms that are in a ring.		
Add group	The program adds a known functional group to a free atom.[3]	CH_4	
Remove group	The program removes a functional group from the molecular structure.		CH_4
Ring Bond Fusion	An existing ring is fused with a compound of the static set of ring fragments, creating a ring fusion bond.		

[2] The program chooses the element from a table consisting of 11 atoms: hydrogen, carbon, oxygen, nitrogen, sulfur, fluorine, chlorine, bromine, iodine, phosphorus and boron.

[3] The program uses two SDF data files containing 291 ring and 296 branch fragments.

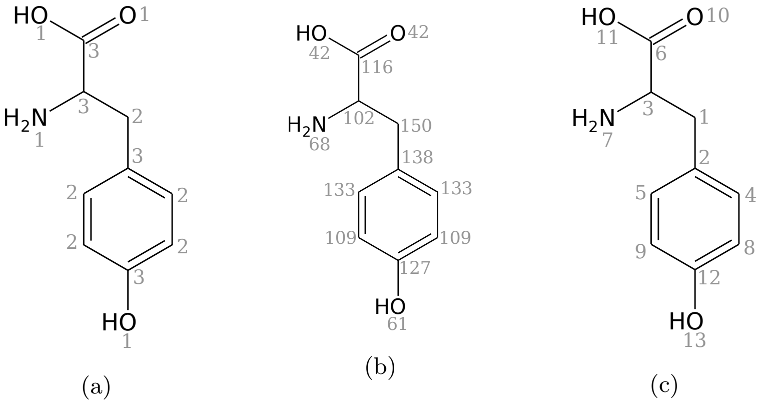


Figure 3.2: The figures above show the three stages of the Morgan algorithm for the tyrosine molecule. (a) shows the atom identifiers after initialization. (b) shows the atom identifiers after a few update cycles. (c) shows the end-results, with each atom identifier assigned a unique value.

that case, the atom with the highest degree of atomic bond takes precedence over the other.

An alternative approach was introduced in the Pipeline Pilot [1] software in 2000, called *Extended-Connectivity Fingerprints* (ECFP) [20]. This method is related to the Morgan algorithm, but it is adapted in two ways. First, the EFCP algorithm terminates after a predetermined number of iterations rather than after identifier uniqueness is achieved. Secondly, the Morgan algorithm tries to avoid collisions, i.e., where two different atom environments are given the same identifier, which might result in two atom identifiers given the same identifier. The ECFP algorithm replaces the collision-avoiding step by a fast hashing scheme. An overview of the algorithm is given below.

Algorithm 1 (ECFP Algorithm)**.**

1. An initial assignment stage, in which each atom has an integer identifier assigned to it.

2. An iterative updating stage, in which each atom identifier is updated to reflect the identifiers of each atom's neighbors, including identification of whether it is a structural duplicate of another feature.

3. A duplicate identifier removal stage, in which multiple occurrences of the same feature are reduced to a single representative in the final feature list. (The occurrence count may be retained if one requires a set of counts rather than a standard binary fingerprint).

3.3.1 Initial assignment stage

To compute an integer identifier for each atom, a set of rules similar to the *Daylight atomic invariants rules* [25] can be used. These encompass seven rules in total that do not depend on the atomic numbering.

- The number of immediate non-hydrogen neighbors;
- The valence minus the number of hydrogen atoms;
- The atomic number;
- The atomic mass;
- The atomic charge;
- The number of attached hydrogens;
- Whether the atom is contained in at least one ring.

The computed values are hashed into a single 32-bit integer value and are used as the initial atom identifier.

3.3.2 Iterative updating stage

Each iteration produces a set of features that represents each atom within a larger substructure of the atom. In the previous stage, the information of an atom with neighborhood of size 0 was captured. For a generalized ECFP_n fingerprint, process is iterated $n/2$ more times, where n is the effective diameter of the largest feature, i.e., the largest possible atom environment after n iterations. For example, if iterated three times, the largest possible fragment has a width of six bonds and the result is an ECFP_6 fingerprint.

In each consecutive step, the atom identifier from the previous iteration is used in the updating of the atom identifiers, which are added to a fingerprint set. This is achieved by using the information from the previous step, effectively looking at a fragment with diameter k, and adding to it the non-hydrogen attachments, thus obtaining the features of a fragment with diameter $k + 1$. The information of the previous step is captured in the current atom identifier, so it does not need to be re-evaluated.

The way this is implemented is as follows. After initialization we have a 32-bit integer atom identifier for each atom in the molecule. Next, we create an array of integers that captures the features of a fragment with radius i, the iteration count. This array is initialized with i as the first number and the current atom identifier as the second. To this list is appended a list of ordered pairs (o, I) with o the bond order and I the atom identifier of the neighbor, which is sorted according to bond order. The new atom identifier is then computed by again hashing the resulting array to a 32-bit integer. This process is repeated for every atom in the molecule.

3.3.3 Duplicate structure removal

An ECFP consists of an array of features as represented by the hashed 32-bit integer values. Two molecules can then be compared by looking at the features that they have. A problem with this updating scheme is that similar substructures can result in different integer values. For example, if two atoms have a different atomic number, they are initialized with different values (see Section 3.3.1). After a few generations, the feature region of one atom might be exactly the same as that of another atom, but might be represented by a different hashed value.

To resolve this possible redundancy, a record of a set of bonds for each fingerprint feature is kept. A *bond set* is a bit string with length equal to the number of bonds in the molecule. This is because a feature region cannot be larger than the total number of bonds in the molecule. Each bit in the bit string is a flag that is set whenever the bond that is represented by the bit is part of the current neighborhood of the atom. When two features describe the same neighborhood, i.e., they are from equivalent bond sets, the features collide and one of them must be removed. Two simple rules are used to achieve this deterministically [20]:

1. If the features were generated from a *different number* of iterations, the feature from the larger number of iterations is rejected.

2. If the features were generated from the *same* number of iterations, then the larger hashed identifier value (interpreted as an integer) is rejected.

3.3.4 Duplicate identifier removal (optional)

The last step is an optional one. As one molecule can contain substructures that may appear several times in the same molecule, we can remove the duplicate fingerprint features that were generated. The terminology *fingerprint with counts* was proposed in the original paper on extended-connectivity fingerprints [20]. The additional information on the duplicate substructures might be useful in some cases.

3.3.5 Remarks on the ECFP algorithm

Let us conclude this overview of the ECFP generation algorithm with three remarks.

Firstly, 64-bit integer values were investigated by Rogers et al. but led to no significant improvements [20]. We therefore use the 32-bit variant, as they are more efficient to work with on 32-bit computer architectures.

Secondly, the choice of a hash function is free. The hash function has only the following two requirements: it should be able to map arrays of integers both *randomly* and *uniformly* distributed into the 2^{32}-size space of all possible integers.

Thirdly, multiple features might be represented by the same bit code, therefore, the absence of a code is determinative, but the presence of a code is only suggestive.

3.4 Jaccard distance coefficient

Now that we can describe molecular structures using Extended-Connectivity Fingerprints, it is possible to compare two molecules based on the substructures

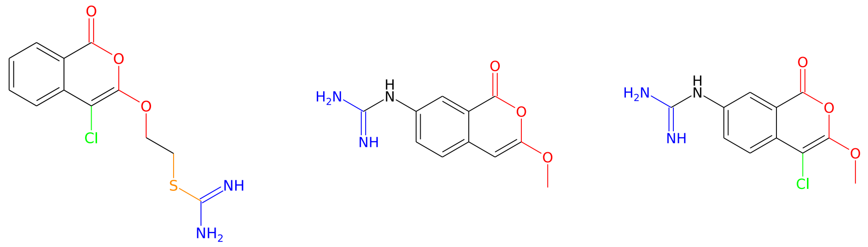


Figure 3.3: The ECFP algorithm has collected several features from which four have been highlighted. It can be seen that the molecules are similar to some degree, as they share some features, e.g. the red and blue substructures.

that each fingerprint is composed of. A widely used similarity index is the *Jaccard* coefficient. In the literature the name *Tanimoto coefficient* is often used, but this is historically defined on bit strings rather than sets of features [8], as in the case of ECFP. Therefore, the remaining part of this text will refer to the Jaccard index.

There are many other coefficients that have been used in chemoinformatics and could have been used here for (dis)similarity measures. This is however not the focus of the thesis and the interested reader could look into [11] for a thorough overview of such coefficients.

Informally, the Jaccard coefficient is defined on two sets of features, A and B, as the number of common features divided by the number of all presented features. Or more formally:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{3.1}$$

This is the ratio of common or shared features of the two sets as compared to the total number of known features. The *dissimilarity coefficient* or *distance measure*, called the Jaccard distance, is easily derived by subtracting the Jaccard similarity from 1.

$$J_\delta(A,B) = 1 - J(A,B) = \frac{|A \cup B| - |A \cap B|}{A \cup B} \tag{3.2}$$

The distance is a real value with interval $[0,1]$, with value of 1 for maximum similarity and 0 for minimum similarity.

An example of similarity search using molecular substructures is given in Figure 3.3. Four features have been given a different color and it can be seen that some of these are shared among the molecules. The ECFP algorithm collects for each molecule all these features and the Jaccard distance is computed according to the number of similarities.

3.4.1 Properties of the Jaccard distance coefficient

A distance coefficient can be seen analogously to distances in multidimensional geometric space, although they are not necessarily precisely equivalent to such distances [27]. A distance coefficient is only a *metric* if the following axioms hold.

1. $d(x, y) \geq 0$ *(non-negativity)*
2. $d(x, y) = 0 \Leftrightarrow x = y$ *(identity of indiscernible)*
3. $d(x, y) = d(y, x)$ *(symmetry)*
4. $d(x, z) \leq d(x, y) + d(y, z)$ *(triangle inequality)*

The first three properties hold trivially for the Jaccard distance, and the fourth property has been proven by Alan Lipkus [15], meaning that it is a proper distance metric.

It is also possible to use a distance metric that is a member of the more general class of Minkowski distances, such as Euclidean or Manhattan distances, which are defined as follows:

$$d_{\mathbb{M}}(x, y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p \right)^{1/p} \qquad |x| = |y| = n \tag{3.3}$$

The fundamental difference between the class of Minkowski distances and the Jaccard distance is that the former considers the common absence of attributes (or low values in continuous space) as evidence for similarity, where the Jaccard distance does not. Since molecular fingerprints are usually bit string and contain mostly zeroes, the Jaccard distance is more natural to use.

Chapter 4

Evolutionary Level-set Algorithm

One of the recently developed evolutionary methods for finding optimal diverse sets of solutions is the indicator based Evolutionary Level-set Algorithm (ELSA), which was proposed by Emmerich, Deutz and Kruisselbrink [6]. This chapter will first give an outline of the Black-Box levelset problem in Section 4.1. After that, a formal description of five indicators for diversity that were used in the conducted experiments is given in Sections 4.2–4.4. Sections 4.5 and 4.6 deal with the feasibility and druglikeness of molecules. This chapter ends with an adaption of the utilities that were discussed to the domain of molecules.

4.1 Black-box Level Set Problem

In classical optimization problems, the objective is often to look for a solution that minimizes (or the inverse: maximizes) a certain objective function. However, if one is not necessarily interested in finding the *optimal* solution, but rather a set consisting of solutions that match certain constraints, the problem is referred to as a *Black-box Level Set Problem.*

Definition 2 (Black-Box Level Set Problem)**.**
Given a black box function $f : X \to \mathbb{R}$ and a target space $T \subseteq \mathbb{R}$, find all solutions in $L = \{x \in X \mid f(x) \in T\}$. Solutions in L will be termed *feasible solutions.*

Here, the target space T could, for example, be a point, an interval box or any space, for which the condition $f(x)$ can be evaluated. A few examples of the problems that belong to this family of problems are given in [6].

In the context of diversity optimization in populations of molecules, this means that there is a function that maps molecules to an N-dimensional space, with N the number of constraints. Furthermore, the goals of the search can be split twofold:

1. Diversity: spread the molecules as much as possible within the target space.

2. Feasibility: find molecules that match given constraints.

4.2 Simple Spread Indicators

To compute the distance between two solutions in a population, the Jaccard distance, based on Extended-Connectivity Fingerprints, is used, as described in the previous chapter. As stated before, the maximization of diversity of the population can be seen as spreading the molecules throughout the decision space, or maximizing the gap between the molecules in this same space. Maximizing diversity can then be restated as maximizing a *spread indicator*. For this purpose, let us first define the distance from an element x to a set A:

$$d(x, A) = min_{a \in A} d(x, a) \tag{4.1}$$

Emmerich et al. [6] have proposed three *simple spread indicators* with "simple" referring to the efficient computability of the indicators.

- $IS_N(A) := \min_{x \in A} d(x, A - \{x\})$ (Minimal Gap)
- $IS_\Sigma(A) := \dfrac{1}{|A|} \sum_{x \in A} d(x, A - \{x\})$ (Arithmetic Mean Gap)
- $IS_\Pi(A) := (\prod_{x \in A} d(x, A - \{x\}))^{\frac{1}{|A|}}$ (Geometric Mean Gap)

Maximizing these indicators results in maximizing diversity of a population of *fixed size*. It does not, however, necessarily reward adding more molecules to the population, i.e., the *monotonicity in species property* does not hold.

Definition 3 (Monotonicity in Species Property)**.**
If $x \notin A$ is added to a set A, then

$$D(A \cup \{x\}) > D(A) + d(x, A)$$

It should be noted that since the ELSA algorithm uses a population of fixed size, this does not cause a problem. Two populations of the same size are easily comparable, and the population with a larger value on the spread indicator can be considered more diverse.

4.3 Weitzman Diversity

A diversity measure that does fulfill the monotonicity and a few other desirable properties, is the Weitzman distance [26], which will be discussed shortly. Weitzman proposed a recursive definition for measuring the diversity in biological species, which is defined as follows.

$$D(A) = D(A - x) + d(x, A - x) \qquad \forall x \in A \tag{4.2}$$

Once the initial condition $D(i) \equiv d_0$ has been set − usually $d_0 = 0$ − the algorithm computes the diversity for each subset of increasing size. This is a very rigorous approach for measuring diversity with the downside that a straightforward approach has a time complexity of $\mathcal{O}(|A|!)$. Weitzman proposed an algorithm that reduces the complexity to $\mathcal{O}(2^{|A|})$. This approach uses a dynamic programming technique and makes use of the notion of *link species*.

Definition 4 (Link property)**.**
For all A, with $|A| \geq 2$, there exists at least one species $x \in A$, called the link species, that satisfies $D(A) = D(A - \{x\}) + d(x, A - \{x\})$.

Informally, this means that there is some species that will reduce the diversity of the population exactly by the distance of that species to its closest relative. Weitzman has shown in his paper that the link species is one of two species that are closest to each other. This means that instead of computing all diversity values for each species in the population on each iteration (which gives rise to the factorial complexity), it is instead possible to compute the diversity value as follows:

$$D_W(A) = \max_{x \in A}\{D_W(A - \{x\}) + d(x, A - \{x\})\}$$

An implementation of this approach would maximize the cumulative sum of distances between the link species from populations of size N to 1. This effectively reduces the time complexity to $\mathcal{O}(2^{|A|})$. A proof can be found in Weitman's paper on diversity.

Definition 5 (Twin Property)**.**
Suppose that some species $y \in A$ is identical to some species $x \notin A$, meaning that for all $z \in A$, $d(x, y) = 0$ and $d(x, z) = d(y, z)$. Then, if y is added to A, there is no change in diversity, i.e., $D(A \cup y) = D(A)$.

This is a reasonable requirement as adding copies of a species does not add any additional value to the population. Replacing a redundant species s_1 by another species s_2 with $d(s_2, A) > 0$ is always preferred and the twin property makes sure this is always the case.

Definition 6 (Continuity in Distances Property)**.**
Let there be two sets A and A', with $|A| = |A'|$. Let $\phi(.)$ be a one-to-one function mapping A onto A'. Then, for all $\epsilon > 0$, there exists a $\delta > 0$, such that $\sum_{i \in A, j \in A} |d(i, j) - d(\phi(i), \phi(j))| < \delta$ implies that $|D(A) - D(A')| < \epsilon$.

This property tells us that a slight modification of pairwise distances in S results in a slight modification of the population's diversity. The property is very useful since there is some uncertainty in the real values of the distances, induced by the discrete representation of real values in a computer.

Definition 7 (Monotonicity in Distances)**.**
Let $|A| = |A'| \geq 2$. Let $\phi(.)$ be a one-to-one function mapping A onto A'. Suppose that $d(\phi(i), \phi(j)) \geq d(i, j)$, for all $i, j \in A$, $i \neq j$. Then $D(A') \geq D(A)$.

This means that if a structurally identical set has a greater or equal distance between pairs, the resulting diversity value should be greater as well. This is a necessary property for any reasonable diversity function.

Definition 8 (Maximum diversity that can be added by a species)**.**
If a species y is added to a set A, then $D(A \cup y) \leq D(A) + d_{max}(y, A)$, for all $y \notin S$, where $d_{max}(y, A) = \max_{i \in A} d(y, i)$.

Intuitively, this can be deduced from the algorithm that uses the link property: the addition to the diversity value per iteration is at most the maximum distance from the added species to the set.

The distance function, based on Extended-Connectivity Fingerprints, has values on the interval $[0,1]$. As the maximum value that can be added is equal to 1, the interval of the Weitzman diversity for the intended purpose is therefore $[0,n]$, with $n = |A|$ the size of the set. The Weitzman diversity indicator therefore gives some intuition on the diversity of a population, since its boundaries are easily determined and normalization can easily be applied.

4.4 Solow-Polasky Diversity

A problem with the Weitzman measure is the time complexity of the algorithm and the interpretability of the indicator value, as it lacks a clear scale on which different values can be compared on. Clearly, if there are two sets A and A' of equal size and $D_w(A') > D_w(A)$ then A' is strictly more diverse than A. However, by just looking at the indicator value of a population with unbounded distances, it is hard to say whether the population is very diverse at all. It would be more convenient to talk about a population A with k species with $0 \leq k \leq |A|$. The Solow-Polasky measure [22] manages to do exactly this.

Definition 9 (Solow-Polasky measure)**.**
Suppose we have a set $A = \{a_1, ..., a_n\}$ where $|A| = n$. Then we can define a matrix M with entries $m_{i,j} = \exp(-\theta \cdot d(a_i, a_j))$, for all $i, j \in \{1, \ldots, n\}$, $\theta > 0$. Then, if M is non-singular, the Solow-Polasky diversity, denoted by D_{SP}, is defined as follows:

$$D_{SP}(A) = \sum_{i=1}^{n} \sum_{j=1}^{n} M_{i,j}^{-1}$$

The computational complexity of the Solow-Polasky indicator is $\mathcal{O}(n^3)$, which makes it more attractive for large sets than the Weitzman diversity indicator [24].

There is, however, a problem with this indicator. If M is not invertible, then the Solow-Polasky indicator value is not defined. We can overcome this problem by using the *Moore-Penrose inverse* (or *pseudo-inverse*) of a matrix, which is defined for any square matrix. A commonly used method is to compute the pseudo-inverse by means of *Singular-Value Decomposition*, but this method is still very expensive to compute. However, since the considered sets are relatively small, this does not pose a severe problem.

4.5 The search space and druglikeness

The chemistry space consists of all possible combinations of atoms and their topology that form molecules. A *drug-like molecule* is a substance that has physicochemical properties that make it more likely that it will interact with proteins in the human body, thus making it a target for further drug research. Such a property can be computed using predictive modeling techniques and provides the chemist with values that can be used to base decisions on.

A traditional method to predict *druglikeness* is Lipinski's rule of 5 [14]. The rule is based on the fact that a drug has to satisfy several molecular properties in order for it to be absorbed and distributed through the body. These properties are the so-called *ADMET* properties referring to absorption, distribution, metabolism, excretion and toxicity. While the rule does not predict whether the drug is pharmacologically active, it is useful as a rule of thumb, since molecules that do not satisfy the rules are unlikely to be drug-like. Lipinski's rules are defined as follows:

- There are no more than 5 hydrogenbond donors (expressed as the sum of OH- and NH-groups);
- The molecular weight is less than 500 Daltons;
- The Log P is over 5 (or MLogP is over 4.15);
- There are more than 10 hydrogenbond acceptors (expressed as the sum of N and O atoms);

This rule will be used as a guideline for setting the constraints in the experiments of Section 5. Many extensions have been proposed that narrow down the drug-like chemical space. The interested reader can look into [9] for more details.

4.6 Feasibility of molecules

The complete set of possible molecules is enormous, but not all molecules have to be screened, since many do not satisy the imposed constraints. These requirements can be, for example, constraints on the ADME properties. Therefore, we can apply filters on these properties to exclude substances of low interest. The molecules that fulfill the requirements are called *feasible*. The next section provides details on how the notion of feasibility is incorporated in the evolutionary level-set algorithm.

4.7 Finding diverse sets of molecules

At the beginning of this chapter, we defined the Black-Box Levelset problem. In this section, this problem will be formalized in the context of finding diverse sets of molecules. As stated before, the goal is to find a set of feasible molecules, i.e., molecules that satisfy user-specified constraints, that are as diverse as possible. The diversity measure can be any diversity measure, such as those described in Sections 4.2–4.4. The search space under consideration is the *chemical space* S of all possible drug-like molecules, where we can use Lipinski's rule of 5 (see Section 4.5) to specify the constraints. The remaining part of this chapter will use a slightly different notation to emphasize the shift from abstracts sets to sets of molecules. S is the chemical space (the search space) and P will be a population of molecules.

The levelset problem for finding drug-like molecules is defined as follows:

Definition 10 (Levelset problem for molecules)**.**
We consider molecules $x \in S$. Furthermore, there is a function $f : S \to \mathbb{R}^k$, that maps molecules to a vector $\boldsymbol{p} = (p_1, ..., p_k)^T$, containing for each property considered its computed value. The target space $T \in \mathbb{R}^k$ is defined by constraints on the values in such a vector $\boldsymbol{p}$.
Find all solutions in the levelset $L = \{x \in S \mid f(x) \in T\}$.

As the set of feasible solutions can be very large, it is essentially impossible and impractical to search for the entire set L. Therefore, an *approximation set* L^* that is maximally diverse and of bounded size, is searched for instead. Using the definitions of distances between molecules and the notion of diversity, it is now possible to further refine the problem.

Definition 11 (Optimize diversity on sets of molecules)**.**
Given a distance measure $d(x, y)$, compute a set $L^* \subseteq L$ of size n that maximizes $D(L^*)$.

In Section 2.1 it was stated that some species are partially redundant, i.e., they share some characteristics. The less features two molecules share, the higher the probability is that the molecules will have unique properties. To maximize the diversity of the population, the *diversity quality indicator contribution* [6] is used. Such indicators compute for each molecule in the population its contribution to the overall diversity.

Definition 12 (Diversity Contribution)**.**
Given a set of solutions P, we define the diversity contribution of a solution $p \in P$ as

$$\Delta_{QI}(p, P) \leftarrow QI(P) - QI(P - \{p\}).$$

Since the approximated levelset L^* contains only feasible molecules, a mechanism for excluding infeasible molecules is required. This is done by penalizing infeasible molecules, using an adapted version of the quality indicator contribution, called *augmented indicators*, as suggested by Emmerich et al. [6].

Definition 13 (Augmented Diversity Contribution)**.**
Given a set of solutions P, the augmented diversity contribution of a solution $p \in P$ is defined as:

$$\Delta_{QI^+}(p, P) \leftarrow \Delta_{QI(p,P)} + d_{\mathbb{M}}(p, T),$$

where $d_{\mathbb{M}}(p, T)$ is a Minkowski distance (see Section 3.4.1) from a molecule $x \in P$ to the target space T. In the experiments of Chapter 5, the Manhattan distance d_1 was used, since it is highly efficient to compute.

Because the contributions are highly correlated, i.e., the contributions of the solutions are dependent on each other, a stable replacement strategie is a steady-state scheme such as that in NOAH [24] and SMS-EMOA [7]. Evolutionary algorithms utilizing such replacement strategies insert a new solution in the population at the beginning of each iteration and then delete one as specified by the objective and constraint functions. In this case, the molecule with the

Algorithm 2

```
procedure ELSA
    $P_0 \leftarrow \text{init}()$
    $t \leftarrow 0$
    while not terminate do
        $q \leftarrow \text{generate}(P_t, p_m)$
        $P_t^{'} \leftarrow P_t \cup \{q\}$
        $r \leftarrow \text{argmax}_{p \in P_t^{'}} \{\Delta_{QI+}(p, P_t^{'})\}$
        $P_{t+1} \leftarrow P_t^{'} \setminus \{r\}$
        $t \leftarrow t + 1$
    end while
    return $P_t$
end procedure
```

lowest augmented indicator value is selected for deletion. By doing so, penalized (infeasible) molecules are likely to be deleted first, after which the diversity is optimized by replacing partially redundant molecules.

The evolutionary level set approximation algorithm given in Algorithm 2 can be used to generate a diverse set of molecules. A population of random molecules is first initialized. After that, an evolutionary loop starts where in each iteration a new molecule is added by either mutating from an existing molecule or by randomly generating a new molecule with a probability p_m. For each molecule in the set, its contribution to the diversity of the population is computed. The molecule with the lowest contribution is then removed from the population and the next iteration starts.

Chapter 5

Experiments

Earlier work on the comparison of the quality indicators described in Sections 4.2–4.4, have shown that even the simple indicators can do a good job in the ELSA algorithm to generate a diverse set of solutions on artificial problems, such as the sphere function [6]. This chapter provides a description of the experiments that were conducted to measure the performance of the indicators in their ability to find diverse sets of molecules. We also explored various values for p_m that controls the probability that a molecule is mutated from an existing parent or randomly generated.

5.1 Comparison of different quality indicators

For this experiment the Weitzman diversity indicator was used, as it has all the minimal properties (see Section 4.3) that are commonly associated with diversity [26]. The indicators that were compared to the Weitzman diversity indicator are the *Minimal Gap* (IS_N), *Arthitmetic Mean Gap* (IS_Σ), *Geometric Mean Gap* (IS_Π) and *Solow-Polasky* (SP). The population size was set to a small value of 10 to reduce the required calculation time, since the Weitzman indicator is computationally very expensive, and was computed at each iteration and for all indicators. The population was initialized by random arrangements of two static sets of ring and branch fragments, which are not required to be feasible. In each iteration of the algorithm, one new molecule was either generated from scratch using this same method with probability $1 - p_m$, or mutated from a parent molecule using one or more randomly picked operators that were mentioned in Section 3.2 with probability p_m.

In this experiment we also compared the performance of the algorithm in two different scenarios. The first one is the *unguided search* using non-augmented indicators which does not take into account any set constraints, i.e., any molecule is considered feasible. The focus herein is solely on the capabilities of the algorithm to find a diverse set and is therefore less likely to be of use in practice. It was included for theoretical intents only. The second scenario is the *guided search* which uses the augmented indicator as defined in Definition 13 (see page 24). This is more relevant since it resembles a real-life use case in which a diverse set of molecules meeting certain constraints is required. We refer to Algorithm 2 (see page 25) for a detailed description of the procedure used.

We also note that there is no explicit order in which the objectives of feasibility and diversity are considered in the case of a guided search, since the augmented diversity indicators aggregate both objectives in one value, the *augmented diversity contribution*. This means that it is possible that infeasible solutions are kept if their contribution to the diversity of the population is significant. It is however very likely that infeasible solutions are replaced by feasible ones.

The mutation probability p_m was set to 0.5. The diversity quality contribution was then computed for each molecule in the population and the molecule with the lowest contribution was deleted from the population. This process was repeated for a total of 100 iterations per indicator. The results of 30 separate runs were collected and the mean over these values for each iteration was taken.

5.2 Mutation probability

No experiments were previously conducted on the mutation probability p_m that is hidden in the *generate* function of line 6 in Algorithm 2. For select this probability value, the effects of varying this parameter from 0.0 (never mutate) to 1.0 (always mutate) were studied.

The population size was again set to 10 and randomly generated from ring and branch fragments. As the quality indicator for diversity, the Weitzman indicator was used because of its “nice” properties, as stated in the previous section. The experiment was run 10 times for 100 iterations.

Chapter 6

Results

In this chapter, the results for the two experimental setups – the non-augmented (unguided) and augmented (guided) as explained in the previous chapter – are described and analyzed.

With Experiment 1 we denote the comparison of the various quality indicators for diversity with respect to the Weitzman indicator, as described in Section 5.1. Experiment 2 refers to the setup that compares various settings for the mutation probability in ELSA as described in Section 5.2.

6.1 Experiment 1: Indicator comparison

The first experiment was run 100 times for each quality indicator, both for non-augmented and augmented versions. The mean of each indicator was computed after each iteration and plotted in Figures A.2 and A.4, which show the evolution of the indicator value over time. The end-results after 100 iterations are shown in Tables 6.1–6.2 and shown as box plots in Figures A.3 and A.5.

The end-results for both non-augmented and augmented indicators show negligible difference, which is particularly clear in the box-plots. Despite the simple nature and low computational complexity of the *simple quality indicators* (Minimal Gap, Arithmetic Mean and Geometric Mean Gap), they perform surprisingly well in the experiments. In the non-augmented experiment, the Geometric Mean Gap was a clear winner with both the best intermediary and end-results, followed by the Solow-Polasky indicator with $\theta = 20$.

From these results it can also be seen that the Solow-Polasky indicator's performance is highly dependent on the value of the θ parameter. Where the averaged results for most indicators show an almost monotonic increase in quality value, the Solow-Polasky indicator with $\theta = 1$ shows many fluctuations, indicating irregular performance over the various runs. This shows another interesting advantage of the simple quality indicators over the Solow-Polasky indicator, namely that these indicators have no parameters that influence the performance.

	MG	AMG	GMG	SP_1	SP_5	SP_{10}	SP_{20}
Mean	7.9658	7.9849	8.0193	7.8482	7.9219	7.9695	8.0070
Median	7.9732	7.9920	8.0248	7.8815	7.9331	7.9966	8.0024
σ	0.1018	0.0887	0.0887	0.2028	0.1337	0.1304	0.0946

Table 6.1: End results after 100 runs with various regular, non-augmented indicators.

	MG	AMG	GMG	SP_1	SP_5	SP_{10}	SP_{20}
Mean	7.7610	7.8561	7.9039	7.8887	7.6587	7.7686	7.5982
Median	7.8373	7.8645	7.9040	7.8846	7.8409	7.8527	7.8424
σ	0.7887	0.1023	0.0897	0.0980	1.1080	0.7957	1.3488

Table 6.2: End results after 100 runs with various augmented indicators.

6.2 Experiment 2: Mutation probabilities

The second experiment investigates the influence of various settings of the mutation probability on the ability of the ELSA algorithm to find diverse sets of molecules. Only the Weitzman indicator is used as the measure of diversity in this experiment.

The mean value over 10 runs was computed after each iteration and for every mutation probability and plotted in Figure A.1. The end-results for each setting of the mutation probability can be found in Table 6.3.

The results show that the obtained population diversities after 100 iterations are very similar for all settings. Setting the mutation probability p_m to 1.0 results in poor performance compared to the other settings.

Mut. prob.	Mean	Median	σ
0.0	8.2667	8.2780	0.0575
0.1	8.2995	8.2750	0.1316
0.2	8.2958	8.2984	0.0544
0.3	8.2916	8.3058	0.0813
0.4	8.2896	8.2674	0.0882
0.5	8.2704	8.2923	0.0735
0.6	8.2520	8.2559	0.0673
0.7	8.2791	8.2758	0.0665
0.8	8.4243	8.2725	0.3081
0.9	8.3746	8.3167	0.2505
1.0	8.2176	8.2420	0.1366

Table 6.3: End results after 10 runs with mutation probabilities.

Chapter 7

Conclusions and Future Work

In this thesis we described methods for generating diverse sets of molecules with user-specified properties. We gave details about the representation, about distance measures using extended-connectivity fingerprints and Jaccard distance, and we described how new molecules are generated from parent molecules using various mutation operators. The ELSA algorithm was able to successfully evolve an initially random set of molecules using a steady-state replacement scheme utilizing the contribution of diversity to the molecule population.

We started out with the following question: How can we apply diversity-based search in drug discovery? We have shown that many of the required tools are available, but that a suitable algorithm for the particular optimization goal of finding diverse sets was not yet experimented with. The ELSA algorithm has shown to be a suitable candidate for the task. The results of the experiments show that all diversity measures, and their respective indicators of diversity contribution, are able to rapidly select infeasible molecules and replace them. In the case of unconstrained diversity optimization, i.e., with use of non-augmented indicators, the algorithm was able to generate a set that was more diverse than the random initial sets.

Another aspect of this thesis was the comparison of various indicators of diversity. Two well-known indicators, Solow-Polasky and Weitzman, were tested with three less complex measures, called simple spread indicators. Our results show that the latter are as suitable as the former in finding diverse sets. This result is rather surprising, since their nature is much more simple and they are less expensive to compute. The Minimal Gap indicator, for example, has complexity $\mathcal{O}(|A|^2)$ for the initial computation of the population diversity, but updates to the indicator can be done in linear time.

The experiments with the mutation probability give no definite answer to the best setting of this parameter. Further research could be conducted to investigate whether this parameter can be dynamically adapted to improve the performance of the algorithm.

Although the METool was able to find diverse populations of molecules, it is unclear whether the molecules it finds can be synthesized. The application should be extended to implement such synthesizability checks, as this would greatly enhance its practical value. Further research should therefore be done on finding a suitable way to integrate the synthesizability, for example as an additional constraint on the objective space.

Bibliography

[1] Accelerys. *Pipeline Pilot*. 2012. URL: `http://accelrys.com/products/pipeline-pilot/`.

[2] Eric Anderson, Gilman D. Veith and David Weininger. *SMILES, a line notation and computerized interpreter for chemical structures*. Technical report EPA/600/M-87/021. US Environmental Protection Agency, Environmental Research Laboratory, 1987.

[3] Regine S. Bohacek, Colin McMartin and Wayne C. Guida. The art and practice of structure-based drug design: a molecular modeling perspective. In: *Medicinal Research Reviews* 16.1 (1996), pages 3–50.

[4] Frank K. Brown. Chapter 35. Chemoinformatics: What is it and how does it impact drug discovery. In: *Annual Reports in Medicinal Chemistry*. Edited by James A. Bristol. Volume 33. Academic Press, 1998, pages 375–384.

[5] Nathan Brown. Chemoinformatics—an introduction for computer scientists. In: *ACM Computing Surveys* 41.2 (Feb. 2009), 8:1–8:38.

[6] Michael T.M. Emmerich, André H. Deutz and Johannes W. Kruisselbrink. On quality indicators for black-box level set approximation. In: *EVOLVE—A Bridge Between Probability, Set Oriented Numerics and Evolutionary Computation*. Edited by Emilia Tantar, Alexandru-Adrian Tantar, Pascal Bouvry, Pierre Del Moral, Pierrick Legrand, Carlos A. Coello Coello and Oliver Schütze. Volume 447. Studies in Computational Intelligence. Springer Berlin Heidelberg, 2013, pages 157–185.

[7] Michael Emmerich, Nicola Beume and Boris Naujoks. An EMO algorithm using the hypervolume measure as selection criterion. In: *Evolutionary Multi-Criterion Optimization*. Edited by Carlos A. Coello Coello, Arturo Hernández Aguirre and Eckart Zitzler. Volume 3410. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, pages 62–76.

[8] Michael A. Fligner, Joseph S. Verducci and Paul E. Blower. A modification of the Jaccard–Tanimoto similarity index for diverse selection of chemical compounds using binary strings. In: *Technometrics* 44.2 (2002), pages 110–119.

[9] Arup K. Ghose, Vellarkad N. Viswanadhan and John J. Wendoloski. A knowledge-based approach in designing combinatorial or medicinal chemistry libraries for drug discovery. 1. A qualitative and quantitative characterization of known drug databases. In: *Journal of Combinatorial Chemistry* 1.1 (Jan. 1999), pages 55–68.

[10] Z. Hippe. Chemical informatics in the organic coating industry. In: *Progress in Organic Coatings* 5.3 (1977), pages 219–227.

[11] Zdenek Hubálek. Coefficients of association and similarity, based on binary (presence-absence) data: an evaluation. In: *Biological Reviews* 57.4 (1982), pages 669–689.

[12] Johannes W. Kruisselbrink, Michael T.M. Emmerich, Thomas Bäck, Andreas Bender, Ad P. IJzerman and Eelke Horst. Combining aggregation with pareto optimization: a case study in evolutionary molecular design. In: *Evolutionary Multi-Criterion Optimization.* Edited by Matthias Ehrgott, Carlos M. Fonseca, Xavier Gandibleux, Jin-Kao Hao and Marc Sevaux. Volume 5467. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, pages 453–467.

[13] Eric-Wubbo Lameijer, Joost N. Kok, Thomas Bäck and Ad P. IJzerman. The Molecule Evoluator. An interactive evolutionary algorithm for the design of drug-like molecules. In: *Journal of Chemical Information and Modeling* 46.2 (2006), pages 545–552.

[14] Christopher A. Lipinski, Franco Lombardo, Beryl W Dominy and Paul J Feeney. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. In: *Advanced Drug Delivery Reviews* 46.1-3 (Sept. 2001), pages 3–26.

[15] Alan H. Lipkus. A proof of the triangle inequality for the Tanimoto distance. English. In: *Journal of Mathematical Chemistry* 26 (1-3 1999), pages 263–265.

[16] Eric J. Martin and Roger E. Critchlow. Beyond mere diversity: tailoring combinatorial libraries for drug Discovery. In: *Journal of Combinatorial Chemistry* 1.1 (Jan. 1999). PMID: 10746013, pages 32–45.

[17] H. L. Morgan. The generation of a unique machine description for chemical structures—A technique developed at Chemical Abstracts Service. In: *Journal of Chemical Documentation* 5.2 (1965), pages 107–113.

[18] Christos A. Nicolaou, Nathan Brown and Constantinos S. Pattichis. Molecular optimization using computational multi-objective methods. In: *Current Opinion in Drug Discovery and Development* 10.3 (May 2007), pages 316–24.

[19] Ronald C. Read and Derek G. Corneil. The graph isomorphism disease. In: *Journal of Graph Theory* 1.4 (1977), pages 339–363.

[20] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. In: *Journal of Chemical Information and Modeling* 50.5 (May 2010). PMID: 20426451, pages 742–754.

[21] Gisbert Schneider and Uli Fechner. Computer-based de novo design of drug-like molecules. In: *Nature Reviews Drug discovery* 4.8 (Aug. 2005), pages 649–63.

[22] Andrew R. Solow and Stephen Polasky. Measuring biological diversity. English. In: *Environmental and Ecological Statistics* 1 (2 1994), pages 95–103.

[23] Andrew Solow, Stephen Polasky and James Broadus. On the measurement of biological diversity. In: *Journal of Environmental Economics and Management* 24.1 (1993), pages 60–68.

[24] Tamara Ulrich and Lothar Thiele. “Maximizing population diversity in single-objective optimization”. In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*. GECCO ’11. Dublin, Ireland: ACM, 2011, pages 641–648.

[25] David Weininger, Arthur Weininger and Joseph L. Weininger. SMILES. 2. Algorithm for generation of unique SMILES notation. In: *Journal of Chemical Information and Computer Sciences* 29.2 (1989), pages 97–101.

[26] Martin L. Weitzman. On diversity. In: *The Quarterly Journal of Economics* 107.2 (May 1992), pages 363–405.

[27] Peter Willett, John M. Barnard and Geoffrey M. Downs. Chemical similarity searching. In: *Journal of Chemical Information and Computer Sciences* 38.6 (1998), pages 983–996.

Appendix A

Figures

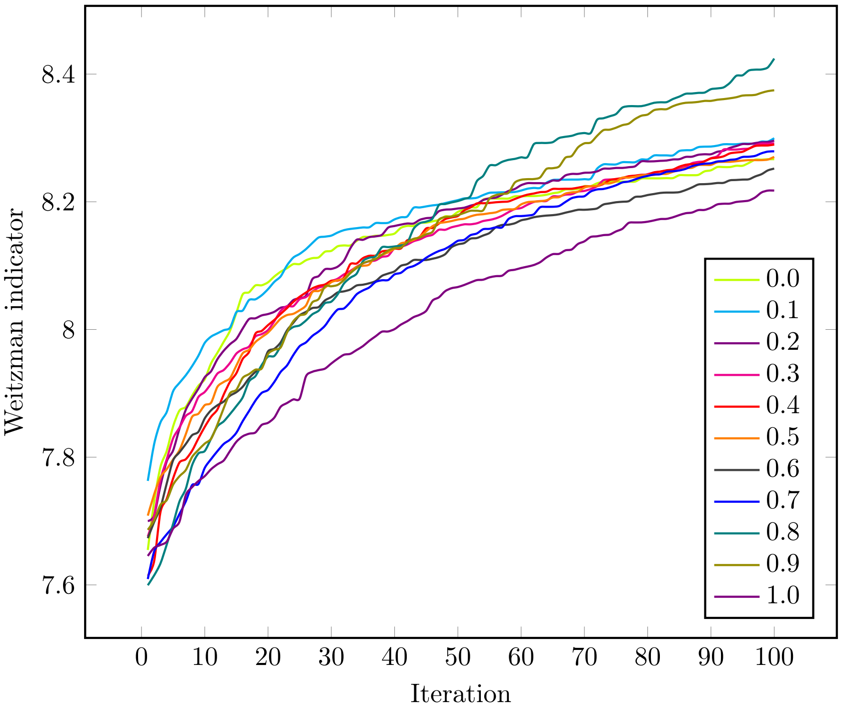


Figure A.1: The evolution of averaged Weitzman indicator value over 10 runs using various mutation probabilities.

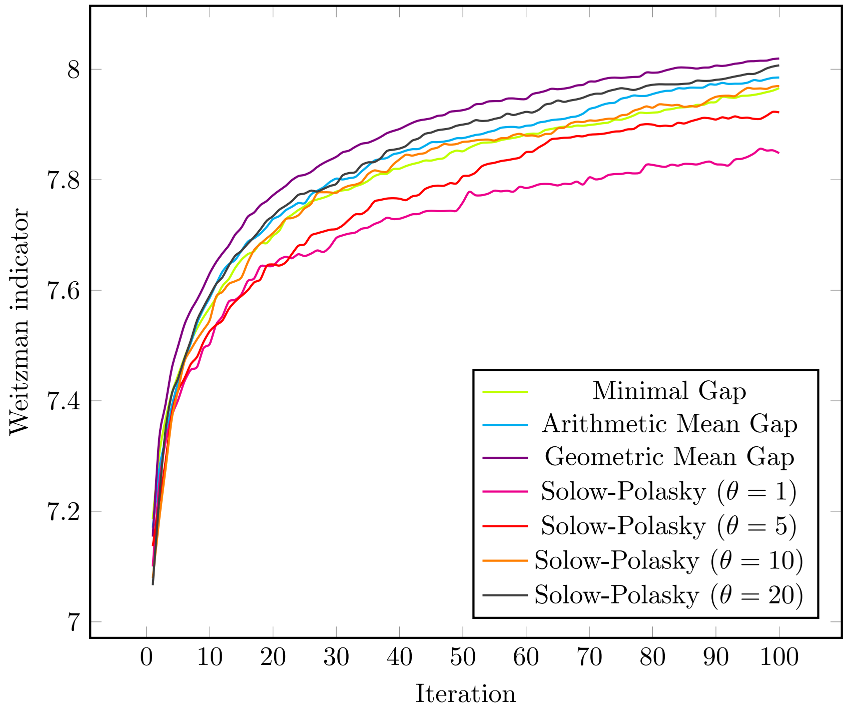


Figure A.2: The evolution of averaged diversity indicator value of 100 runs using various regular, non-augmented indicators.

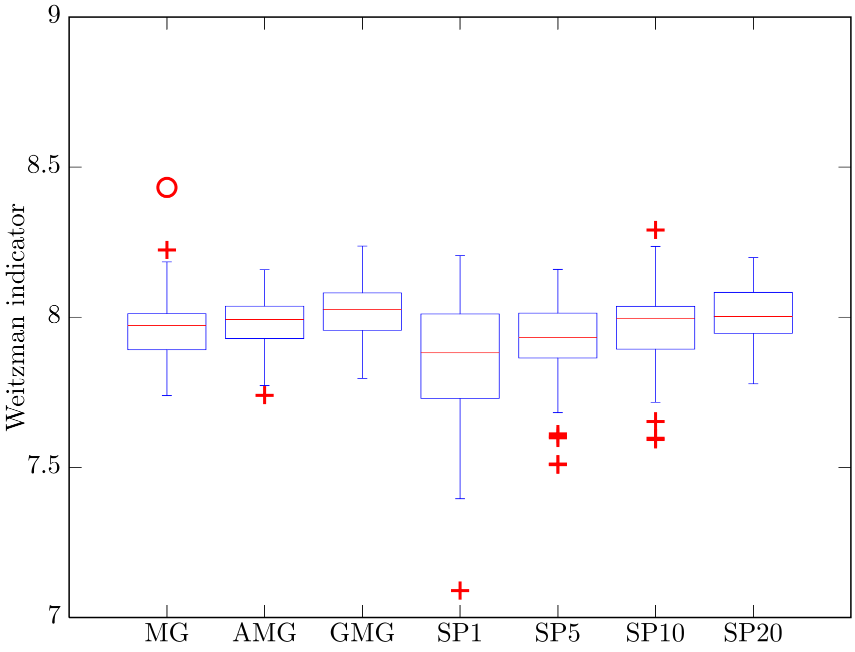


Figure A.3: Box plot of the results of Experiment 1 after 100 runs with various non-augmented indicators.

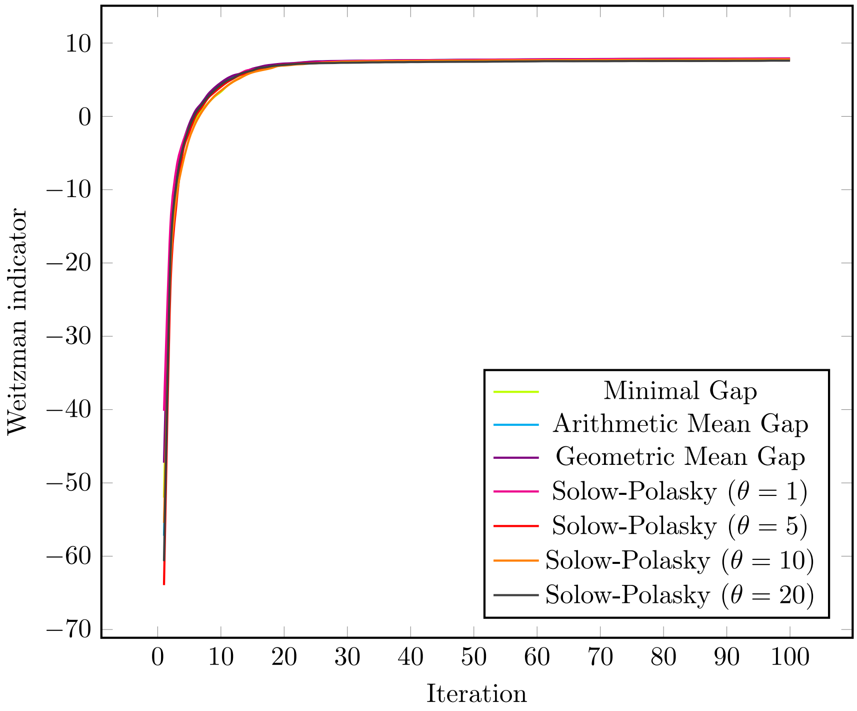


Figure A.4: The evolution of averaged diversity indicator value of 100 runs using various augmented indicators.

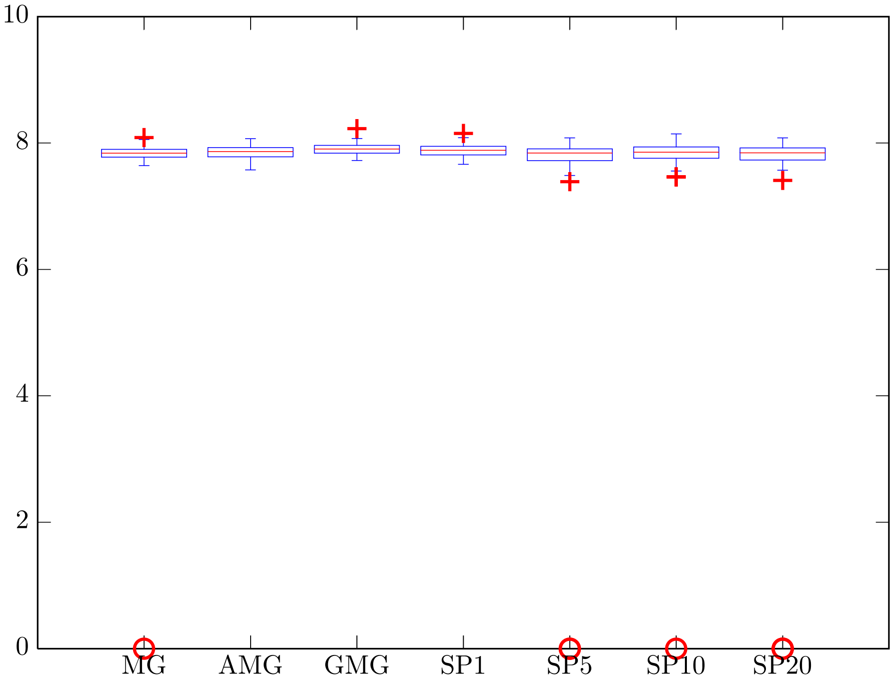


Figure A.5: Box plot of the results of Experiment 1 after 100 runs with various augmented indicators.