Molecular Feature Mining in HIV Data

Stefan Kramer, Luc De Raedt and Christoph Helma
Institute for Computer Science, Machine Learning Lab
Albert-Ludwigs-University Freiburg
Georges-Köhler-Allee Geb. 79
D-79110 Freiburg/Br., Germany
{skramer,deraedt,helma}@informatik.uni-freiburg.de

ABSTRACT

We present the application of Feature Mining techniques to the Developmental Therapeutics Program's AIDS antiviral screen database. The database consists of 43576 compounds, which were measured for their capability to protect human cells from HIV-1 infection. According to these measurements, the compounds were classified as either active, moderately active or inactive. The distribution of classes is extremely skewed: Only 1.3 % of the molecules is known to be active, and 2.7 % is known to be moderately active.

Given this database, we were interested in molecular substructures (i.e., features) that are frequent in the active molecules, and infrequent in the inactives. In data mining terms, we focused on features with a minimum support in active compounds and a maximum support in inactive compounds. We analyzed the database using the levelwise version space algorithm that forms the basis of the inductive query and database system MolFea (Molecular Feature Miner). Within this framework, it is possible to declaratively specify the features of interest, such as the frequency of features on (possibly different) datasets as well as on the generality and syntax of them. Assuming that the detected substructures are causally related to biochemical mechanisms, it should be possible to facilitate the development of new pharmaceuticals with improved activities.

1. INTRODUCTION

Recently, the database research community has devoted a lot of attention to data mining. Vice versa, the data mining community demonstrated an increasing interest in (and a need for) advanced database techniques. The common research interests of these communities has led to a number of proposals for integrating advanced database technology and data mining techniques. One of the most interesting proposals is that for inductive databases [10]. Inductive databases integrate data with patterns, i.e. generalizations or regularities. Furthermore, they provide a query language (and inductive database management system) that supports the querying of both patterns and data. Ideally, the query language should allow the data miner to declaratively specify the patterns of interest using a number of constraints on e.g. frequency, generality, syntax, etc. Today, several proposals for inductive database query languages exist (cf. [15, 8, 10, 5]). Most of them support the mining of association rules (and their variants).

In this paper, we present a domain specific inductive database called MolFea (Molecular Feature Miner) for use in computational chemistry. MolFea mines for fragments in chemicals. Fragments are (linear) substructures of compounds. The compounds are stored in the SMILES format [21] and fragments are formulated in the SMARTS language (see also [11]). SMILES and SMARTS are commonly used in computational chemistry; they are supported by many tools, such as, e.g., the Daylight toolkits [11]. Within MolFea, the user can declaratively specify the fragments of interest using simple but powerful primitives. Primitives that are supported include mining for fragments that have a minimum (resp. maximum) frequency on a set of compounds, that are more general (resp. more specific) than a given fragment, etc. As an example query, consider finding all fragments that are frequent on the active compounds, infrequent on the inactives and that are a subfragment of a specified compound, and extend a specified fragment. MolFea efficiently computes the solutions to inductive queries using the levelwise version space algorithm that we have introduced elsewhere [6, 13]. The levelwise version space algorithm integrates Mitchell's version space algorithm [17] with the well-known levelwise algorithm [14] underlying Apriori [1].

The effectiveness of MolFea is demonstrated here on an important scientific discovery application: MolFea was used to analyze a set of 43576 compounds from the DTP AIDS antiviral screen database (`http://dtp.nci.nih.gov`). In this domain, the problem is to find structural properties that are related to the protection of human cells from HIV-1 infection [23].

The paper is organized as follows: in section 2, an overview of the molecular feature miner MolFea is given; in section 3, we present the DTP AIDS antiviral screen database; in section 4, we report on the experiments and the findings in the data; finally, in section 5, we touch upon related work and conclude.

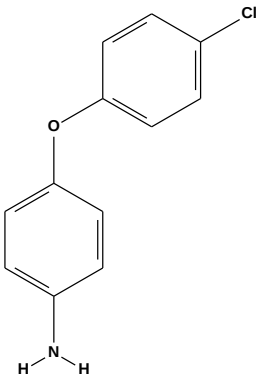

Figure 1: Example compound in a 2-D representation. $'Cl-c:c:c:c-O'$ is an example fragment occurring in the molecule. (Benzene rings consist of six carbon atoms connected by aromatic bonds.)

2. THE MOLECULAR FEATURE MINER

2.1 Representing Molecules Using SMILES

Figure 1 shows an example structure in a 2-D representation. The compound contains two benzene rings (consisting of aromatic carbons) connected through an oxygen atom. The 2-D structure of a compound can be represented in different ways. One common representation format consists of so-called connection tables. In a connection table, all pairs of atoms with bonds between them are listed and specified. Also multi-relational database representations (cf., e.g., [4]) or graph-based representations [18] have been used for mining such data. In MolFea, we use the representation language SMILES (Simplified Molecular Input Line Entry System), which is a wide-spread language for representing molecules. It is supported by all major software tools in computational chemistry such as, e.g., the Daylight toolkit [11].

The SMILES representation of the compound in Figure 1 is 'Nc1ccc(Oc2ccc(Cl)cc2)cc1', or 'N-c1:c:c:c(-O-c2:c:c:c(-Cl):c:c2):c:c1', when the bonds are explicitly written. This notation can be understood as follows:

- A SMILES representation is essentially a sequence of atoms and bonds; for instance, '$O-C-C$' in SMILES means: "an oxygen atom connected with a single bond to a carbon atom connected with a single bond to another carbon". In such expressions 'C', 'N', 'Cl', etc. denote elements, and '$-$' denotes a single bond, '=' a double bond, '#' a triple bond, and ':' an aromatic bond.

- The elements of aromatic atoms are written in lowercase, otherwise the elements start with a capital.

- The hydrogen atoms and the single and aromatic bonds need – under certain conditions – not be written. They can be computed from the other information.

- Cyclic structures are represented by breaking one bond in each ring. The atoms adjacent to the broken bond obtain the same number. E.g. 'cccccc' denotes a sequence of 6 aromatic carbons and 'c1ccccc1' denotes a *ring* of 6 aromatic carbons, a benzene ring. In transcribing Figure 1, the lower ring is identified by number 1, and the upper one by number 2. The first two carbons subsequent to the nitrogen on the right-hand side of the lower benzene ring can be broken and labeled with '1'. Analogously, the upper ring can be encoded.

- Side-structures are written between brackets. E.g. in the example compound, the upper substructure starting from the oxygen atom is written as 'Oc2ccc(Cl)cc2'. So, we have an oxygen followed by a benzene ring. Furthermore, there is a chlorine bonded to the fourth carbon of the benzene ring.

The final SMILES code for the overall compound in Figure 1 can then be regarded as a nitrogen (hydrogens are implicitly added), followed by a benzene ring which is connected by an oxygen to a chlorinated benzene ring: 'Nc1ccc(Oc2ccc(Cl)cc2)cc1'.

While the SMILES notation is not unique, it is possible to use the so-called canonical SMILES notation [22], which guarantees that the resulting string encoding the compound is unique. Also note that, e.g., for matching a pattern against an example compound, an internal data structure is constructed and used.

There are three key advantages of using SMILES for data mining applications in computational chemistry. Firstly, SMILES is a language designed, understood and "spoken" by (computational) chemists. Secondly, SMILES code is extremely compact for storage and manipulation as compared to other representations. Indeed, on the website of Daylight Chemical Information Systems one finds the statement that "a typical SMILES will take 50% to 70% less space than an equivalent connection table." A database of 23,137 structures is said to require only an average of 1.6 bytes per atom. Using Ziv-Lempel compression, this can further be reduced to 0.42 bytes per atom.

As another illustration, consider the Prolog formulation for use in multi-relational data mining or inductive logic programming of the compound in Figure 1. It occupies 2100 bytes. Contrast this with the 26 bytes needed for SMILES. Thirdly, efficient and optimized tools exist for testing whether a fragment matches a compound. Using SMILES (and SMARTS, cf. below) one can rely on the existing state-of-the-art computational chemistry technology. As we will see, MolFea employs the SMILES and SMARTS toolkits from Daylight Chemical Information Systems as the underlying computational chemistry tool.

2.2 Representing Features and Fragments Using SMARTS

The aim of MolFea is to mine for molecular fragments (sometimes also called features) of interest in chemical data. To represent such fragments we use a subset of the language SMARTS, which is derived from SMILES in order to specify substructures. Almost all SMILES specifications are valid SMARTS targets. However, SMARTS also allows one to make abstraction of specific elements in the fragment. E.g. the notation $'\sim'$ is used to denote *any* type of bond, the character $'*'$ denotes *any* type of atom. Further abstractions are built into SMARTS [11]. Also, the semantics of a SMARTS construct is different than that of a SMILES one. SMARTS encode fragments, SMILES encode molecules, and these can be quite different, cf. [11]. In the present implementation

of MolFea, we focus on a simple subset of SMARTS called $\mathcal{M}$ which corresponds to fully specified linear sequences of atoms and bonds. Extensions including side-chains and abstractions are planned.

A molecular fragment f *covers* an example compound e if and only if f considered as a graph is a subgraph of example e. For instance, fragment $'Cl-c:c:c:c-O'$ covers the example compound in Figure 1.

There are a number of interesting properties of the language of molecular fragments $\mathcal{M}$:

- fragments in $\mathcal{M}$ are partially ordered by the *is more general than* relation; when fragment g is more general than fragment s we will write $g \leq s$;
- within this partial order, two syntactically different fragments are equivalent only when they are a reversal of one another; e.g. $'C-O-S'$ and $'S-O-C'$ denote the same substructure;
- $g \leq s$ if and only if g is a subsequence of s or g is a subsequence of the reversal of s; e.g. $'C-O' \leq 'C-O-S'$.

Note that the representation of molecular fragments is relatively restricted compared to some other representations employed in data mining, such as first-order queries [4] or subgraphs [18]. Although fragments are a relatively restricted representation of chemical structure, it is easy for trained chemists to recognize the functional group(s) that a given fragment occurs in. Thus, the interpretation of a fragment reveals more than meets the eye.

2.3 Constraints on Fragments

The features of interest can be declaratively specified using a conjunction of primitive constraints $c_1 \wedge ... \wedge c_n$. The primitive constraints c_i that can be imposed on the unknown target fragments f are:

- $f \leq p$, $p \leq f$, $\neg(f \leq p)$ and $\neg(p \leq f)$: where f is the unknown target fragment and p is a specific pattern; this type of primitive constraint denotes that f should (not) be more specific (general) than the specified fragment p; e.g. the constraint $'C-O' \leq f$ specifies that f should be more specific than $'C-O'$, i.e. that f should contain $'C-O'$ as a subsequence;
- $freq(f, D)$ denotes the relative frequency of a fragment f on a set of molecules D; the relative frequency of a fragment f w.r.t. a dataset D is defined as the percentage of molecules in D that f covers;
- $freq(f, D_1) \leq t$, $freq(f, D_2) \geq t$ where t is a positive real number and D_1 and D_2 are sets of molecules; this constraint denotes that the relative frequency of f on the dataset D_i should be larger than (resp. smaller than) or equal to t; e.g. the constraint $freq(f, Pos) \geq 0.95$ denotes that the target fragments f should have a minimum relative frequency of 95 % on the set of molecules Pos.

These primitive constraints can now conjunctively be combined in order to declaratively specify the target fragments of interest. Note that the conjunction may specify constraints w.r.t. any number of datasets, e.g. imposing a minimum frequency on a set of active molecules, and a maximum one on a set of inactive ones. E.g. the following constraint:

$$('C-O' \leq f) \wedge \neg(f \leq 'C-O-S-C-O-S') \wedge$$

$$freq(f, Act) \geq 0.95 \wedge freq(f, InAct) \leq 0.05)$$

queries for all fragments that include the sequence $'C-O'$, are not a subsequence of $'C-O-S-C-O-S'$, have a frequency on Act that is larger than 95 percent and a frequency on $InAct$ that is smaller than 5 percent.

2.4 Solving Constraints

In this subsection, we show that the solution space $sol(c_1 \wedge ... \wedge c_n)$ in $\mathcal{M}$ for a conjunctive constraint $c_1 \wedge ... \wedge c_n$ is a version space and can therefore be represented by its borders.

Due to the fact that the primitive constraints c_i are independent of one another, it follows that

$$sol(c_1 \wedge ... \wedge c_n) = sol(c_1) \cap ... \cap sol(c_n)$$

So, we can find the overall solutions by taking the intersection of the primitive ones.

Secondly, each of the primitive constraints c is monotonic or anti-monotonic w.r.t. generality (cf. [14]). A constraint c is *monotonic* (resp. anti-monotonic) w.r.t. generality whenever

$$\forall s, g \in \mathcal{M} : (g \leq s) \wedge (g \in sol(c)) \rightarrow (s \in sol(c))$$

(resp. replace $\leq$ by $\geq$). The basic anti-monotonic constraints in our framework are: $(f \leq p), freq(f, D) \geq m$, the basic monotonic ones are $(p \leq f), freq(f, D) \leq m$. Furthermore the negation of a monotonic constraint is anti-monotonic and vice versa.

Monotonic and anti-monotonic constraints are important because their solution space is bounded by a border. This fact is well-known in both the data mining literature (cf. [14]), where the borders are often denoted by BD^+, as well as the machine learning literature (cf. [17]), where the symbols S and G are typically used.

To define borders, we need the notions of minimal and maximal elements of a set w.r.t. generality. Let F be a set of fragments, then define

$$min(F) = \{f \in F \mid \neg\exists q \in F : f \leq q\}$$

$$max(F) = \{f \in F \mid \neg\exists q \in F : q \leq f\}.$$[1]

We can now define the borders $S(c)$ and $G(c)$[2] of a primitive constraint c as

$$S(c) = min(sol(c)) \text{ and } G(c) = max(sol(c))$$

Anti-monotonic constraints c will have $G(c) = \{\top\}$ and for proper constraints $S(c) \neq \{\bot\}$; proper monotonic constraints have $S(c) = \{\bot\}$ and $G(c) \neq \{\top\}$. Furthermore, as in Mitchell's version space framework we have that

$$sol(c) = \{f \in \mathcal{M} \mid \exists s \in S(c), \exists g \in G(c) : g \leq f \leq s\}$$

[1] Note that *min* gives the *minimally general* elements, and *max* the *maximally general* ones. In contrast, $g \leq s$ denotes that g is more general than s.

[2] At this point, we will follow Mitchell's terminology, because he works with two dual borders (a set of maximally general solutions G and a set of maximally specific ones S). In data mining, one typically only works with the S-set.

This last property implies that $S(c)$ (resp. $G(c)$) are proper borders for anti-monotone (resp. monotone) constraints.

So, we have that the set of solutions $sol(c_i)$ to each primitive constraint is a simple version space completely characterized by $S(c_i)$ and $G(c_i)$. Therefore, the set of solutions $sol(c_1 \wedge ... \wedge c_n)$ to a conjunctive constraint $c_1 \wedge ... \wedge c_n$ will also be completely characterized by the corresponding $S(c_1 \wedge ... \wedge c_n)$ and $G(c_1 \wedge ... \wedge c_n)$.

Elsewhere [6, 5], we have presented the levelwise version space algorithm for computing the S and G sets corresponding to the constraints. This algorithm basically integrates the levelwise algorithm by [14] with Mellish's description identification algorithm. In principle, one might also employ Hirsh's version space merging algorithm [9]. The most important property of the levelwise version space algorithm is that it essentially has the levelwise algorithm as a special case. Furthermore, the key optimizations made in Apriori like algorithms carry over. This includes e.g. the determination of the frequency of all candidates at a certain level using one pass through the database.

2.5 The Levelwise Version Space Algorithm

In this section, we outline the part of the levelwise version space algorithm [6] that is relevant to the task at hand. In particular, we sketch those parts of the algorithm that handle minimum and maximum frequency constraints. Other constraints are handled by Mellish's description identification algorithm [16].

The algorithms outlined below employ refinement operators:

- A refinement operator $\rho_s(f) = max\{f' \in \mathcal{M} \mid f < f'\}$, i.e. extending a fragment by one atom.
- A generalization operator $\rho_g(f) = min\{f' \in \mathcal{M} \mid f' < f\}$, i.e. removing one atom from one side of a fragment.

To deal with the frequency constraints c, we employ the following generalization of the levelwise algorithm.

Let c be a constraint of type $freq(f, D) \geq m$
$L_0 := G$; $i := 0$
while $L_i \neq \emptyset$ **do**
  $F_i := \{p \mid p \in L_i$ *and p satisfies constraint c*$\}$
  $I_i := L_i - F_i$ the set of infrequent fragments considered
  $L_{i+1} := \{p \mid \exists q \in F_i : p \in \rho_s(q)$
    and $\exists s \in S : p \leq s$ *and* $\rho_g(p) \cap (\cup_j I_j) = \emptyset$ $\}$
  $i := i + 1$
endwhile
$G := F_0$
$S := min(\cup_j F_j)$

To explain the algorithm, let us first consider the case where $S = \{\bot\}$ and $G = \{\top\}$. In this case, the above algorithm will behave roughly as the levelwise algorithm. The L_i will then contain only fragments of size i and the algorithm will keep track of the set of frequent fragments F_i as well as the infrequent ones. The algorithm will then repeatedly compute a set of candidate refinements L_{i+1}, delete those fragments that cannot be frequent by looking at the frequency of its generalizations, and evaluate the resulting possibly frequent fragments on the database. This process continues until L_i becomes empty.

The basic modifications to the levelwise algorithm that we made are concerned with the fact that we need not consider any fragment that is not in the already computed version space (i.e. any element not between an element of the G and the S set). Secondly, we have to compute the updated S set, which should contain all frequent fragments whose refinements are all infrequent.

It is also possible to modify the above algorithm (exploiting the dualities) in order to handle *monotonic* frequency constraint of the form $freq(f, D) \leq m$. In this case, we employ the following algorithm:

Let c be a constraint of type $freq(f, D) \leq m$
$L_0 := G$; $i := 0$
$S := \{s \in S \mid s$ *satisfies* $c\}$
while $L_i \neq \emptyset$ **do**
  $I_i := \{p \mid p \in L_i$ *and p satisfies constraint c*$\}$
  $F_i := L_i - I_i$ the set of frequent fragments considered
  $L_{i+1} := \{p \mid \exists q \in F_i : p \in \rho_s(q)$
    and $\exists s \in S : p \leq s$ *and* $\rho_g(p) \cap (\cup_j I_j) = \emptyset$ $\}$
  $i := i + 1$
endwhile
$G := max(\cup_j I_j)$

In [6], we also introduced dual variants of these algorithms (searching "upwards").

2.6 Optimizations

Various optimizations to the algorithms are possible.

First, though we have adopted the standard levelwise algorithm to search for the borders when handling frequency constraints, it would also be possible to adopt some more recent and more efficient algorithms, such as those presented by [2, 7]. These directly focus on the most specific (the longest) patterns, i.e. the S-set.

Secondly, Apriori-style algorithms can be made more efficient, if elements of one level in levelwise search are combined to give the candidates for the subsequent one. This can also be done for fragments. For instance, if '$O-S-C$' and '$S-C-O$' are known to be frequent at level 3, '$O-S-C-O$' is a candidate for a frequent fragment at level 4. However, several variants (with respect to order) have to be considered; e.g. '$O-S-C$' and '$S-O-C$' can be combined into '$C-S-O-C$' as well as into '$C-O-S-C$'.

Thirdly, we keep track of the fragments in canonical form. As indicated earlier, each fragment is equivalent to its reversal. In the implementation, we use the canonical form of a fragment which is defined as the maximum (w.r.t. a lexicographic ordering) of the fragment and its reversal. The implementation of the operators takes care of this.

Fourthly, one problem with the implementation of the framework for fragments stems from the fact that the bottom $\bot$ is not a "valid" fragment that can be manipulated. In particular, $\rho_g(\bot)$ is not defined. Thus, we cannot search upwards from the set S if $S = \{\bot\}$. Instead, we have to search downwards from G. If however, S is not equal to $\{\bot\}$, then we can process maximum frequency constraints "upwards" starting with S. For the same reason ($\rho_g(\bot)$ is undefined), we cannot start a query with a constraint $\neg(f \leq p)$, where p is a concrete fragment.

2.7 Implementation

For the implementation, we used functions from the SMILES and SMARTS toolkits available from Daylight Chemical Information Systems. The levelwise version space algorithm is implemented in Prolog, but the coverage tests

Table 1: Results by level: # f.'s denotes the number of solution fragments per level, T_{min_f} the time spent for this level solving the minimum frequency constraint, and T_{max_f} the time spent for the maximum frequency constraint. Runtimes are measured in CPU seconds on a Linux PC with a Pentium III 600 MHz processor.

	min(CA) = 13, max(CI) = 516			min(CA) = 13, max(CM) = 8		
Level	# f.'s	T_{min_f}	T_{max_f}	# f.'s	T_{min_f}	T_{max_f}
1	-	4.68	19.83	-	4.69	2.48
2	1	7.19	39.78	-	7.45	3.18
3	5	13.83	107.79	2	13.67	5.04
4	7	14.54	203.51	1	14.42	7.91
5	35	35.79	577.76	7	35.41	16.28
6	58	60.54	986.66	16	59.84	27.96
7	130	97.52	1475.32	5	96.67	42.25
8	171	136.97	1823.14	70	135.37	54.36
9	206	162.56	1787.57	90	162.12	59.78
10	241	183.88	1880.89	105	181.18	49.76
11	251	188.21	1112.46	108	188.02	38.09
12	191	144.22	448.90	104	141.42	19.60
13	132	88.63	258.86	73	88.00	9.18
14	75	54.97	217.68	42	54.05	10.13
15	43	31.19	225.74	26	30.79	7.15
16	24	19.09	269.22	12	19.65	6.26
17	20	17.93	99.37	8	18.10	6.01
18	12	17.88	59.86	5	17.83	4.92
19	8	5.15	-	4	5.10	2.85
20	5	3.59	-	3	3.56	0.37
21	4	3.28	-	2	3.26	-
22	2	3.18	-	1	3.18	-
23	1	0.87	-	-	0.88	-
24	1	0.91	-	-	0.91	-
25	-	1.24	-	-	1.23	-

at each level are entirely delegated to functions in the SMILES and SMARTS toolkits. For each level, we construct the internal data structures of each fragment, then test instance by instance and increment the counters for the fragments accordingly. At any point in time, only one molecule is fully built up in some internal data structure for matching the fragments.

3. THE DEVELOPMENTAL THERAPEUTICS PROGRAM'S AIDS ANTIVIRAL SCREEN DATABASE

3.1 The Database

The DTP AIDS Antiviral Screen program (`http://dtp.nci.nih.gov`) has checked tens of thousands of compounds for evidence of anti-HIV activity. Available are screening results and chemical structural data on compounds that are not covered by a confidentiality agreement. The available database (October 1999 Release) contains the screening results for 43,382 compounds. We are not aware of previous data mining attempts in this domain.

The screen utilizes a soluble formazan assay to measure protection of human CEM cells from HIV-1 infection [23]. Compounds able to provide at least 50 % protection to the CEM cells were retested. Compounds that provided at least 50 % protection on retest were listed as moderately active (CM, confirmed moderately active). Compounds that reproducibly provided 100 % protection were listed as confirmed active (CA). Compounds neither active nor moderately active were listed as confirmed inactive (CI).

We used the conversion tool `babel` (`http://smog.com/chem/babel`) to generate SMILES strings from the connection tables provided by the DTP's Drug Information System, which generated 41768 syntactically correct compounds: 40282 of class CI, 1069 of class CM and 417 of class CA.

3.2 Experimental Set-Up

In this subsection, we describe the queries posed to MolFea for feature mining in the HIV domain. We will assume that there are two sets of examples, D_1 and D_2.[3] Furthermore, $E = D_1 \cup D_2$. During feature mining we will take into account the class information. This is akin to e.g. [19], but contrasts with [3]. Class information is taken into account using the following type of query $(freq(f, D_1) \geq \epsilon) \wedge (freq(f, D_2) \leq \delta)$. One then finds those fragments that are frequent in example set D_1 and infrequent in D_2. Features that are a solution to these types of queries are likely to be predictive with regard to the class information. The choice of ϵ and δ is essentially free. E.g., one can choose these parameters in such a way that the resulting fragments are significant w.r.t. the class distribution.

Our main interest in the experiments was to mine for frag-

[3]We wish to stress that it is easy to generalize the techniques for n classes. Some of the techniques can also be generalized for other learning tasks such as e.g. regression.

Table 2: Summary of the most significant fragments responsible for anti-HIV activity. # CA, #CI and # CM denote the absolute frequencies of the fragment on the respective classes. G/S denotes the origin of the fragment: either from the G set or from the S set of solution fragments. *crit.* means the criterion based on which it was chosen: either statistical significance (the χ^2 statistic) or accuracy (e.g., simply #CA/(#CA + #CI)).

Shorth.	Fragm.	#CA	#CI	G/S	crit.
a	N-C-c:c:c:o	21	25	G	acc.
b	N=N=N-C-C-C-n:c:c:c=O	51	11	S	χ^2
c	N=N=N-C-C-C-n:c:n:c=O	51	11	S	χ^2
d	C-C-C-C-C-C-C-C-C-C-C-C-O-C-C-O-C-C-C-N=N=N	15	0	S	acc.
e	C-C-C-C-C-C-C-C-C-C-C-C-O-C-C-C-N=N=N	15	0	S	acc.
f	O=C-C-C-C-C-O-C-C-O-C-C-C-O-C=O	14	1	S	acc.
g	N=N=N-C-C-C-O-C-C-O-P=O	22	2	S	acc.
h	N=N=N-C-C-C-O-P=O	22	2	S	acc.
Shorth.	Fragm.	#CA	#CM	G/S	crit.
i	N=N-C-C-C-O-C=O	27	3	G	χ^2
j	C-C-C-O-C-C-C-N=N	27	3	G	χ^2
k	N-C-C-C-N-C-O	26	2	G	χ^2
l	C-c:c:c:o:c-C	21	0	G	acc.
m	C-O-C-c:c:c-N	22	0	G	acc.
n	C-O-C-c:c:c:c:c-N	22	0	G	acc.
o	N-C-c:c:c:o	21	0	G	acc.
p	N-C-c:c:o	21	0	G	acc.
q	N=N=N-C-C-C-O-C-C-O-C=O	27	3	S	χ^2

ments with a minimum support in the active compounds (class CA), and a maximum support in inactive compounds (class CI). A second question of interest was: Which fragments discriminate moderately active (CM) and active compounds (CA)? In other words, what distinguishes the positive examples from near misses?

Although the application of MolFea is quite straightforward, we still had to choose reasonable settings for the minimum and maximum support parameters. We arbitrarily set the minimum relative frequency in the active examples to 3%. This corresponds to an absolute minimum frequency of 13. We were then interested in fragments that are, statistically significant, over-represented in the active compounds and under-represented in the inactive (resp. moderately active) compounds. So, we used the χ^2-Test applied to a 2×2 contingency table with the class as one variable and the occurrence of the fragment as the other one to determine the maximum allowable frequency in the non-active examples.

In this way, we obtained a minimum frequency of 13 in active compounds and a maximum frequency of 516 in inactive compounds for the first task at hand (discriminating actives and inactives). Given these frequencies, the occurrence of a fragment in the active class is not due to chance at a significance level of 0.999. Likewise, we obtained a minimum frequency of 13 in the actives (class CA) and a maximum frequency of 8 in moderately actives (class CM) for the second task at hand (discriminating actives and moderately actives).

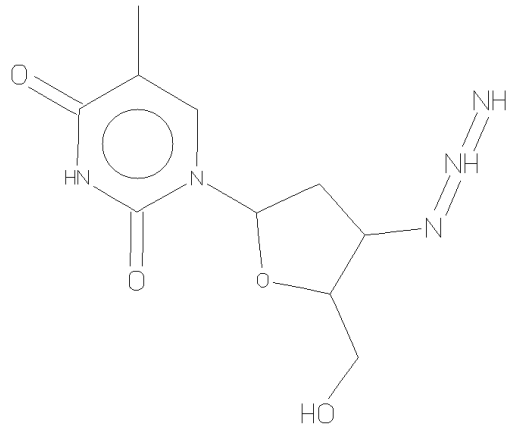

Figure 2: Chemical Structure of Azidothymidine

4. RESULTS

In this section, we summarize our results and findings from applying MolFea to the AIDS Antiviral Screen Database. After presenting a few quantitative results, we continue with an interpretation of the discovered fragments.

In Table 1, the quantitative results of our experiments are shown. The left part of the table contains the results for the first task (distinction between active and inactive compounds) and the right part the results for the second task (distinction between active and moderately active compounds). For each level of the search,[4] we list the number of fragments in the solution space, and the time (in CPU seconds) spent for this level solving the minimum frequency constraint (T_{min_f}) and solving the maximum frequency constraint (T_{max_f}). It can be seen that the solution space for the first task is larger than for the second task.

[4]Note that the level is corresponding to the length of the fragments.

It also becomes apparent that the majority of fragments is found in the medium levels. Also, the majority of the computation time is spent in the medium levels.

For the first task, the total computation time was 19212.31 CPU seconds, where the first part (the minimum frequency query) took only 1544.09 CPU seconds, and the second part (the maximum frequency query) took 17668.22 CPU seconds. The boundary set G contained 222 elements, and S contained 314 elements.

For the second task, the total computation time was 2054.07 seconds, where the first part took 1532.50 and the second part took 521.57 seconds. The boundary set G contained 110 elements, and S contained 127 elements.

For further inspection, we ranked fragments from the boundary sets G and S according to statistical significance (based on the χ^2 statistic) and according to accuracy. Finally, a subset of 17 extremely significant and accurate fragments was chosen for interpretation (see Table 2).[5]

An inspection of these fragments shows, that the majority (fragments b,c,d,e,f,g,h,i,j,q) indicate compounds that are derivatives of Azidothymidine (AZT, Retrovir, Zidovudine, 3'-Azido-3'-deoxythymidine, CAS 30516-87-1, see Figure 2), a potent inhibitor of HIV-1 replication, which is widely used in the treatment of HIV infection. Azidothymidine is structurally similar to Thymidine, one of the four bases which form the alphabet of the DNA. After infection the virus has to replicate its RNA into DNA, which will be incorporated into the host DNA. Azidothymidine interferes with this process, because it is incorporated into DNA like Thymidine, but it cannot form phosphodiester linkages which are necessary for the further elongation of the DNA. So it acts as a chain-terminator of viral DNA during reverse transcription (these compounds are known as reverse transciptase inhibitors).

The fragments of Table 2 reflect the structural requirements for nucleoside analogs to act as reverse transciptase inhibitors. First of all, an azido group ('N=N=N') has to be attached to the 3'-hydroxy group of the sugar moiety (implicitly covered by 'C-C-C-O' or 'C-C-O', depending on the search path in the 5-ring). This group makes the difference between Thymidine and Azidothymidine, it prevents the formation of phosphodiester linkages and thus the replication of virus DNA. The presence of the pyrimidine ring is indicated by sequences like 'n:c:n:c', the relative position in relation to the azido group may be deferred from the number carbons in between. Another requirement is the oxygen in 5' position at the five-ring (indicated by sequences like 'N-C-C-C-O-C-C-O' or 'N-C-C-C-O'). At this position variations are allowed. While AZT has only a '-OH' group at this position other active compounds may be phosphorylated (fragments g,h) or contain longer aliphatic chains or nonaromatic ring systems (fragments d,e,f,q).

The other fragments (a,l,m,n,o,p) indicate another class of reverse transciptase inhibitors, mainly thiocarboxanilide derivatives. They do not mimic nucleosids, but interact directly with DNA polymerase. According to our knowledge these compounds are still in an experimental phase. Also in this case, the structural requirements for anti-HIV activity can be derived from the fragments, using an argumentation as in the examples above.

Similarly other families of compounds with an activity against HIV may be detected from an inspection of the remaining fragments. The automated rediscovery of the most important classes of anti-HIV drugs indicates the utility of the presented approach.

[5]The association between the selected fragments and class membership was extremely strong, so it appears unlikely that this was due to chance. However, another procedure based on randomization tests [12] would be conceivable.

5. RELATED WORK AND CONCLUSIONS

The presented work is related to a variety of different techniques in data mining. This includes inductive logic programming or multi-relational databases, and inductive databases.

MolFea actually has its roots in inductive logic programming and multi-relational data mining (cf. [5]) and its applications (cf. e.g. [4, 20]). However, the main difference between MolFea and other approaches to mining structural or relational data is that MolFea is a special purpose data mining system (as one described in [18]). This has several advantages: understandability and usability by domain experts, as their language is used, efficiency, as one can rely on (almost) optimal procedures for testing frequency and coverage, compactness of the representation. The benefit of this approach becomes clear when one contrasts the present experiments with those performed with traditional inductive logic programming methods. The presented experiments took a couple of hours and involved over 40000 compounds. Past experiments in computational chemistry with inductive logic programming have been limited to a few hundred compounds and have taken at least as much time. In the PTE carcinogenicity domain (337 compounds), where the problem setting is quite similar to the one described in this paper, the general-purpose system Warmr [3] can only search for conjunctions of up to 6 literals. In contrast, MolFea can search for fragments with up to 25 atoms in 40000 compounds.

MolFea can and should also be considered an inductive database. It differs from other proposals for inductive databases such as e.g. [15, 8, 5] in that it is domain-specific and also that it supports a larger variety of constraints. This includes a variety of syntactic constraints based on generality as well as imposing minimum or maximum frequency thresholds on a variety of data sets. Furthermore, as shown here, the constraints can efficiently be solved using the levelwise version space algorithm that we have introduced elsewhere [6]. Early experiments with feature mining in common inductive logic programming benchmarks have been presented in [13]. The present work differs in that MolFea is now speaking SMILES and SMARTS and coupled with the Daylight toolkits. Furthermore, the application to HIV data is new.

Finally, despite the domain specificity of MolFea, we would like to stress that the underlying principles, primitives and algorithms go beyond computational chemistry. Indeed, at an abstract level MolFea mines for sequential fragments in data. This is an important and general task when mining scientific (and other) data, such as e.g. protein or DNA databases. The authors believe that it should be feasible to build MolFea-like systems for use in these domains.

Acknowledgements

This work was partially supported by the European Union IST programme under contract number IST 2000-26469

(consortium on discovering knowledge with inductive queries – cInQ).

6. REFERENCES

[1] R. Agrawal, T. Imielinski, A. Swami. Mining association rules between sets of items in large databases. in: *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, 207–216, 1993.

[2] R. Bayardo. Efficiently mining long patterns from databases. in: *SIGMOD 1998: Proceedings of ACM SIGMOD International Conference on Management of Data*, 85–93, 1998.

[3] L. Dehaspe, H. Toivonen, R.D. King. Finding frequent substructures in chemical compounds. in: *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, 30–36, AAAI press, 1998.

[4] L. Dehaspe, H. Toivonen. Discovery of frequent datalog patterns. in *Data Mining and Knowledge Discovery* 3(1):7–36, 1999.

[5] L. De Raedt. A logical database mining query language. in: *Proceedings of the 10th Inductive Logic Programming Conference*, 78–92, Lecture Notes in Artificial Intelligence, Vol. 1866, Springer Verlag, 2000.

[6] L. De Raedt, S. Kramer. The levelwise version space algorithm and its application to molecular fragment finding. in: *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI-01)*, 2001.

[7] D. Gunopulos, H. Mannila, S. Saluja. Discovering all most specific sentences by randomized algorithms. In F.N. Afrati, P. Kolaitis (eds.): *Database Theory - ICDT '97, 6th International Conference*, 215–229, Lecture Notes in Computer Science 1186, Springer, 1997.

[8] J. Han, L. V. S. Lakshmanan, R. T. Ng. Constraint-based, multidimensional data mining. *Computer*, Vol. 32(8):46–50, 1999.

[9] H. Hirsh. Generalizing version spaces. *Machine Learning*, Vol. 17(1):5–46, 1994.

[10] T. Imielinski, H. Mannila. A database perspective on knowledge discovery. *Communications of the ACM*, 39(11):58–64, 1996.

[11] C.A. James, D. Weininger, J. Delany. *Daylight theory manual – Daylight 4.71*, Daylight Chemical Information Systems, 2000. `http://www.daylight.com/`

[12] D.D. Jensen, P.R. Cohen. Multiple comparisons in induction algorithms. *Machine Learning* 38(3):309–338, 2000.

[13] S. Kramer, L. De Raedt. Feature construction with version spaces for biochemical applications. in: *Proceedings of the Eighteenth International Conference on Machine Learning (ICML-01)*, 2001.

[14] H. Mannila, H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining and Knowledge Discovery*, 1(3):241–258, 1997.

[15] R. Meo, G. Psaila, S. Ceri. An extension to SQL for mining association rules. *Data Mining and Knowledge Discovery*, 2(2):195–224, 1998.

[16] C. Mellish. The description identification problem. *Artificial Intelligence*, 52(2):151–167, 1991.

[17] T.M. Mitchell. Generalization as search, *Artificial Intelligence*, 18(2), 1982.

[18] A. Inokuchi, T. Washio, H. Motoda. An Apriori-based algorithm for mining frequent substructures from graph data. in: D. Zighed, J. Komorowski, J. Zyktow (eds.), *Proceedings of the Fourth European Conference on Principles of Data Mining and Knowledge Discovery (PKDD-2000)*, 13–23, Lecture Notes in Artificial Intelligence, Vol. 1910, Springer-Verlag, 2000.

[19] A. Srinivasan, R. King. Feature construction with inductive logic programming: a study of quantitative predictions of biological activity aided by structural attributes. *Data Mining and Knowledge Discovery*, 3(1):37–57, 1999.

[20] A. Srinivasan, R.D. King, D.W. Bristol. An assessment of submissions made to the predictive toxicology evaluation challenge. in: *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, 270–275, 1999.

[21] D. Weininger. SMILES 1. Introduction and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28, 31, 1988.

[22] D. Weininger, A. Weininger, J.L Weininger. SMILES II, algorithm for generation of unique SMILES notation. *Journal of Chemical Information and Computer Sciences*, 29, 97, 1989.

[23] Weislow, O.S., R. Kiser, D.L. Fine, J.P. Bader, R.H. Shoemaker, M.R. Boyd. New soluble formazan assay for HIV-1 cytopathic effects: application to high flux screening of synthetic and natural products for AIDS antiviral activity. *Journal of the National Cancer Institute*, 81:577–586, 1989.