THE MOLECULAR FEATURE MINER MOLFEA

CHRISTOPH HELMA, STEFAN KRAMER AND LUC DE RAEDT

Albert-Ludwigs-Universität Freiburg, Institut für Informatik, Georges-Köhler-Allee,
Geb. 079, D-79110 Freiburg im Breisgau, Germany
E-Mail: {helma, skramer, deraedt}@informatik.uni-freiburg.de

Received: 26th June 2002 / Published: 15th May 2003

ABSTRACT

Inductive databases are a new generation of databases, that are capable of dealing with data but also with patterns or regularities within the data. A user can generate, manipulate and search for patterns of interest using an inductive query language. Data mining then becomes an interactive querying process.

The inductive database framework is especially interesting for bio- and chemoinformatics, because of the large and complex databases that exist in these domains, and the lack of methods to gain scientific knowledge from them. In this article we present an example for inductive databases: Molfea is the Molecular Feature Miner that mines for linear fragments in the 2D-structure of chemical compounds. In the methodological part we will explain the inner working of the Molfea algorithm, using a simple example. In the second part we will present applications to the NCI DTP AIDS Antiviral Screen database and several benchmark Structure-Activity Relationship problems in toxicology.

INTRODUCTION

The automation of experimental techniques in biology and chemistry has led to an enormous growth of biochemical databases. But the generation of data is only the first step towards a better understanding of the underlying biochemical mechanisms and processes. The second step – where computer science plays a central role – is the analysis of the data in order to find patterns and regularities of scientific interest. In a third step, these patterns have to be interpreted and related to current knowledge, in order to obtain new hypothesis and scientific insights. The whole process of identifying valid, novel, potentially useful and ultimately understandable patterns and models is called *Knowledge Discovery* (1).

Data Mining is a step in the Knowledge Discovery process, that consists of the application of statistics, machine learning and database techniques to the data. During the last few years, it became obvious, that a tight integration of advanced database technologies and data mining techniques would be very desirable. One of the most interesting proposals in this respect are *Inductive Databases* (2, 3, 4, 5). Inductive databases tightly integrate data and patterns, i.e. generalizations or regularities within the data, in a database. They provide also a query language and an inductive database management system that supports the querying of both patterns and data.

The query language allows the user to specify the patterns that are of interest (using a number of constraints e.g. on frequency, generality, syntax, etc., cf. below), the inductive database management system searches efficiently for the patterns that satisfy these constraints.

The inductive database framework is extremely attractive for bio- and cheminformatics because it provides a tool to support scientists in each of the three steps sketched above. Our favourite view of an inductive database user is a scientist who queries interactively an inductive database (possibly with the help of a graphical interface), who inspects the resulting patterns, obtains new ideas, and reformulates the original query until a new scientific insight is obtained.

In this paper, we present a domain specific inductive database called MOLFEA (Molecular Feature Miner). MOLFEA is an instance of the general *Inductive Database Framework*. Another instance is e.g. PROFEA, the Protein Feature Miner, an inductive database for the analysis of the secondary structure of proteins (6). In the next section we will explain the basic concepts of the MOLFEA algorithm using examples and analogies, readers who are interested in a formal presentation are referred to the original literature (7, 8). In the third section we will present some applications of MOLFEA to biomedical and toxicological databases. Finally we will discuss related work, limitations and future extensions in the last section.

METHODS

MOLFEA mines databases with chemical structures for fragments, i.e. linear sequences of atoms and bonds, that fulfil user defined criteria. Within MOLFEA, the user can specify the fragments of interest using simple but powerful *primitives*. Primitives may require e.g. that fragments have a minimum (resp. maximum) frequency on a set of compounds, or that they contain a given subfragment.

A query might request, for example, all fragments that are present in at least 90% of the active molecules but in less than 5% of the inactives. MOLFEA efficiently computes the solutions to these inductive queries using the level wise version space algorithm (8).

MOLECULAR FRAGMENTS

Fragments are linear sequences of non-hydrogen atoms and bonds that are present in molecules. `N-C-C-O`, for instance, is a fragment meaning: “a nitrogen connected with a single bond to a carbon connected to a carbon connected to an oxygen”. A molecular fragment *f matches* an example compound *e* if and only if *f* is a substructure of *e*. For instance, fragment `N-C-C-C` matches the first example compound in the dataset *A* of Fig. 1.

In computer science terms, fragments are strings over an alphabet consisting of elements and bond types. The language of molecular fragments M has some interesting properties, which can be used to develop efficient algorithms:

- **Generality:** One fragment *g is more general* than a fragment *s* (Notation: g ≤ s) if *g* is a substructure of *s* (e.g. `C-O` is more general than `N-C-C-O`). This has the consequence that *g* matches whenever *s* does.
- **Symmetry:** Two syntactically different fragments are equivalent, when they are a reversal of one another (e.g. `C-C-O` and `O-C-C` denote the same substructure).
- **Summary:** $g \leq s$ if and only if *g* is a subsequence of *s* or *g* is a subsequence of the reversal of *s* (e.g. `C-O` ≤ `C-C-O` and `O-C` ≤ `C-C-O`).

CONSTRAINTS ON FRAGMENTS

The fragments of interest can be specified by declaring constraints. Using the example datasets from Figure 1, it is e.g. possible to ask for fragments, that are present in at least two molecules from *A* and in not more than one compound from *B*. In more formal terms, one would formulate a query:

$$freq\ (f, A) \geq 2 \wedge freq\ (f, B) \leq 1$$

The whole query consists of a conjunction of primitive constraints $c_1 \wedge \ldots \wedge c_n$. Presently the following primitive constraints c_i are implemented in MOLFEA:

- $f \geq p$, $p \geq f$, $\neg\ (f \geq p)$ and $\neg\ (p \geq f)$: where *f* is the unknown target fragment and *p* is a predefined fragment; this type of primitive constraint denotes that *f* should (not) be more specific (general) than the specified fragment *p*; e.g. the constraint *f* ≥ `C-O` specifies that *f* should be more specific than `C-O`, i.e. that *f* should contain `C-O` as a subsequence;

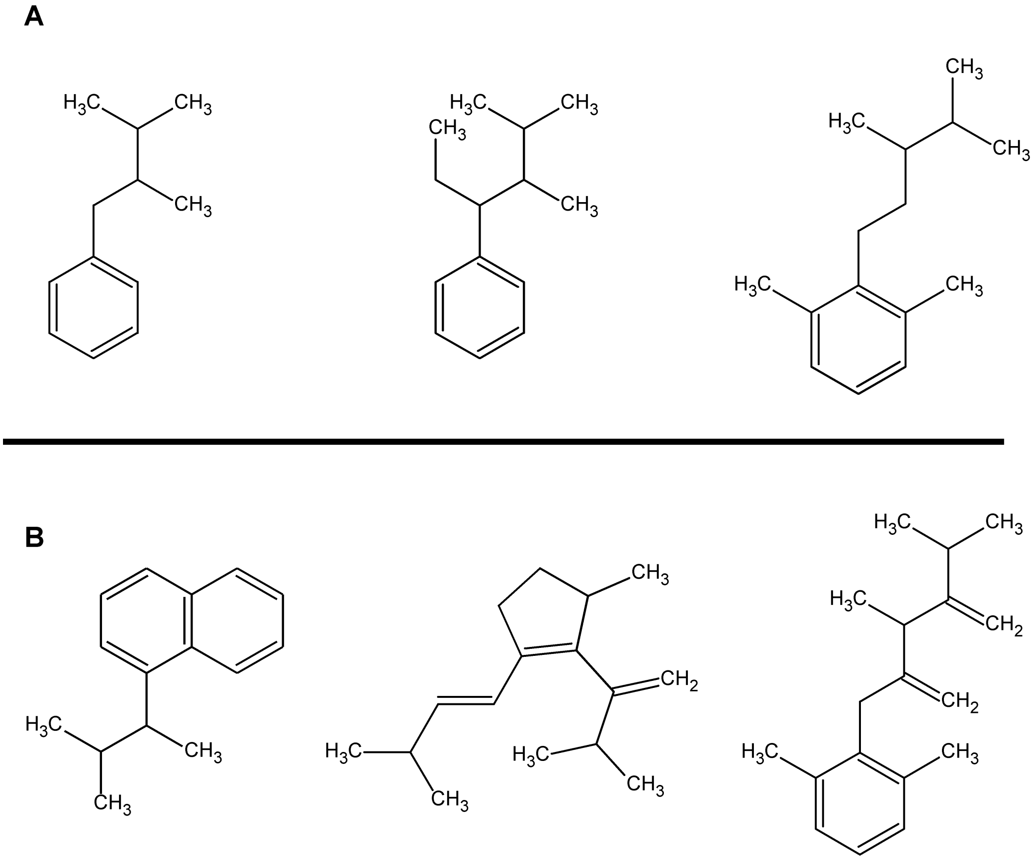

Figure 1. Example datasets A and B for the query $freq(f, A) \geq 2 \wedge freq(f, B) \leq 1$.

- $freq(f, D)$ denotes the frequency of a fragment f on a set of molecules D; the frequency of a fragment f in a database D is defined as the number of molecules in D that f matches;

- $freq(f, A) \geq t$, $freq(f, B) \leq t$ where t is a positive number and A and B are sets of molecules; this constraint denotes that the frequency of f on the dataset A or B should be larger than (resp. smaller than) or equal to t; e.g. the constraint $freq(f, A) \geq 2$ denotes that the target fragments f should match at least 2 molecules in the A set of active molecules. The first type of primitive is called a maximum frequency constraint, the second one a minimum frequency constraint.

These primitive constraints can be combined conjunctively in order to specify the fragments of interest. Note that the conjunction may specify constraints with respect to 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:

$$(\texttt{C-O} \leq f) \wedge \neg (f \leq \texttt{N-C-C-C-O}) \wedge freq\ (f, X) \geq 200 \wedge freq(f, Y) \leq 10$$

queries for all fragments that include the sequence `C-O`, are not a subsequence of `N-C-C-C-O`, match more than 200 molecules in X and less than 10 molecules in Y .

Solving Constraints: The Molecular Feature Miner MOLFEA

In this section we will demonstrate, how MOLFEA solves queries for fragments efficiently. A naive approach would possibly generate all fragments, that are present in the dataset and check, which of the fragments fulfil the given criteria. This method is, of course, computationally very expensive and infeasible for large datasets. But there are more efficient ways to accomplish the same goal.

We will start with a simple minimum frequency constraint $freq(f, D) \geq t$. This constraint has the important property of *anti monotonicity*. To illustrate anti-monotonicity let us assume, we have two fragments *g* and *s* and we know that:

- *g* is more general than *s* (i.e. $g \leq s$; e.g. *g*: `C-O`, *s*: `C-O-S`), and that
- *s* is a solution to our constraint (i.e. *s* matches at least *t* times in *D*)

Then the anti-monotonicity allows us to conclude, that *g* is also a solution. According the definition of generality, general fragments match whenever the specific ones do. The general fragment *g* must be therefore at least as frequent as the specific fragment *s*. Anti-monotonicity is also the reason, why we do not have to determine all solutions for a query. For anti-monotonic constraints, it is sufficient to know the set of the most specific fragments *S*, all fragments that are more general than an element in *S*, will also fulfil the constraint.

Maximum frequency constraints $freq(f, D) \leq t$, in contrast are monotonic. If *g* is more general than *s* and *g* is a solution, we know, that *s* must be a solution, because whenever *g* does not match, *s* will also not match. In this case we have to determine the most general set of fragments *G*, to determine all solutions.

The concept of determining borders, that completely characterize the set of solutions, is a well known idea in Machine Learning and Data Mining (9, 10, 11). It is especially useful, when we want to solve conjunctive queries, consisting of several primitive constraints c_i. In this case we can take advantage of their independency.

$$sol(c_1 \wedge ... \wedge c_n) = sol(c_1) \,\hat{}\,...\,\hat{}\, sol(c_n)$$

So, we can find the overall solutions by taking the intersection of the primitive ones. In practice, we determine S_1 and G_1 for the first primitive constraint, and update S_i and G_i sequentially for each constraint, depending on the monotonicity of the primitive.

The basic anti-monotonic constraints in the MOLFEA framework are presently: $(f \leq p), freq(f, D) \geq m$, 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.

So far we have not yet explained, how to find S or G for the primitive constraints. We will use the datasets A and B from Figure 1 as an example. If we are interested in finding all fragments, that occur at least twice in compounds A but not more than once in B, we can formulate the query

$$freq(f, A) \geq 2 \wedge freq(f, B) \leq 1$$

We will use the MOLFEA output in Figures 2 and 3 to illustrate the following procedure. Let's start with the first primitive constraint $freq(f, A) \geq 2$. The MOLFEA algorithm uses the set of the simplest possible fragments, the elements, as a starting point. These are the candidates for the first level. The next step is to eliminate those, that are too infrequent (i.e. $freq(f, A) < 2$). The remaining ones (C,N,O and aromatic carbon) fulfil our constraint, but they are not the most specific solution (i.e. there are longer fragments, containing the same elements, that are also frequent). So we have to generate more specific (i.e. longer) fragments for the next level.

Considering the generality relationship, we can take an important shortcut: It is not necessary to elongate the frequent fragments with all possible elements, but we have to consider only the frequent fragments {C,N,O,C}. Chemistry is not considered in this step, because "wrong" fragments are removed in the next elimination step. Again we check for frequencies, eliminate infrequent fragments and generate candidates for the next level by combining frequent fragments of the present level, under consideration of the symmetry relationship.

This process is repeated until no more specific fragments can be generated. The union of the most specific fragments is the new S_1 set {C-C-C-O, N-C-C-O, C-C-C-C-C-C-C-C-C, N-C-C-C-C-C-C-C-C}, the smallest (i.e. most general) fragments, that fulfil the constraint are the new G_1 set {C, N, O, C}.

The solution for the first primitive constraint G_1 and S_1 sets are the input for the algorithm that solves the second constraint $freq(f, B) \leq 1$. In this case, we have to remove those fragments that are too frequent in the dataset B. In other words, we have to update G to remove fragments that match more than one compound in the dataset B. Figure 3 shows the MOLFEA output for this toy example. We end up with a final G set {C-C, C-O, C-C-C-C} and S set {C-C-C-O, N-C-C-O, C-C-C-C-C-C-C-C-C, N-C-C-C-C-C-C-C-C}.

```
LEVEL 1:
Candidates:        [Li], [Be], B, C, N, O, F, [Na], [Mg], [Al], [Si], P, S, Cl, [K],
                   [Ca], [Sc], [Ti], [V], [Cr], [Mn], [Fe], [Co], [Ni], [Cu], [Zn],
                   [Ga], [Ge], [As], [Se], Br, [Rb], [Sr], [Y], [Zr], [Nb], [Mo],
                   [Tc], [Ru], [Rh], [Pd], [Ag], [Cd], [In], [Sn], [Sb], [Te], I.
                   [Ca], [Ba], [Lu], [Hf], [Ta], [W], [Re], [Os], [Ir], [Pt], [Au],
                   [Hg], [Tl], [Pb], [Bi], [Po], [At], [Rn], [Fr], [Ra], [Lr], c, n,
                   s, o, p  (78)
Frequent:          C, N, O, c  (4)
==
LEVEL 2:
Candidates:        C-C, C-N, C-O, C-c, C=C, C=N, C=O, C=c, C#C, C#N, C#O, C#c, N-N,
                   N-O, N-c, N=N, N=O, N=c, N#N, N#O, N#c, O-O, O-c, O=O, O=c, O#O,
                   O#c, c-c, c=c, c#c  (30)
Frequent:          C-C, C-N, C-O, C-c, c-c  (5)
==
LEVEL 3:
Candidates:        C-C-C, C-C-N, C-C-O, C-C-c, N-C-N, N-C-O, N-C-c, C-N-C, O-C-O,
                   O-C-c, C-O-C, C-c-c, c-C-c, C-c-C, c-c-c  (15)
Frequent:          C-C-C, C-C-N, C-C-O, C-C-c, C-c-c, c-c-c  (6)
==
LEVEL 4:
Candidates:        C-C-C-C, C-C-C-N, C-C-C-O, C-C-C-c, N-C-C-N, N-C-C-O, N-C-C-c,
                   O-C-C-O, O-C-C-c, C-C-c-c, c-C-C-c, C-c-c-c, C-c-c-C, c-c-c-c  (14)
Frequent:          C-C-C-O, C-C-C-c, N-C-C-O, N-C-C-c, C-C-c-c, C-c-c-c, c-c-c-c  (7)
==
LEVEL 5:
Candidates:        O-C-C-C-O, O-C-C-C-c, C-C-C-c-c, c-C-C-C-c, N-C-C-c-c, C-C-c-c-c,
                   C-c-c-c-c, C-c-c-c-C, c-c-c-c-c  (9)
Frequent:          C-C-C-c-c, N-C-C-c-c, C-C-c-c-c, C-c-c-c-c, c-c-c-c-c  (5)
==
LEVEL 6:
Candidates:        C-C-C-c-c-c, N-C-C-c-c-c, C-C-c-c-c-c, C-c-c-c-c-c, C-c-c-c-c-C,
                   c-c-c-c-c-c  (6)
Frequent:          C-C-C-c-c-c, N-C-C-c-c-c, C-C-c-c-c-c, C-c-c-c-c-c, c-c-c-c-c-c  (5)
==
LEVEL 7:
Candidates:        C-C-C-c-c-c-c, N-C-C-c-c-c-c, C-C-c-c-c-c-c, C-c-c-c-c-c-c,
                   C-c-c-c-c-c-C, c-c-c-c-c-c-c  (6)
Frequent:          C-C-C-c-c-c-c, N-C-C-c-c-c-c, C-C-c-c-c-c-c, C-c-c-c-c-c-c  (4)
==
LEVEL 8:
Candidates:        C-C-C-c-c-c-c-c, N-C-C-c-c-c-c-c, C-C-c-c-c-c-c-c.
                   C-c-c-c-c-c-c-C  (4)
Frequent:          C-C-C-c-c-c-c-c, N-C-C-c-c-c-c-c, C-C-c-c-c-c-c-c  (3)
==
LEVEL 9:
Candidates:        C-C-C-c-c-c-c-c-c, N-C-C-c-c-c-c-c-c  (2)
Frequent:          C-C-C-c-c-c-c-c-c, N-C-C-c-c-c-c-c-c  (2)
==
LEVEL 10:
Candidates:        (0)
Frequency          (0)

G: C, N, O, c  (4)
S:  C-C-C-O, N-C-C-O, C-C-C-c-c-c-c-c-c, N-C-C-c-c-c-c-c-c  (4)
```

Figure 2. MOLFEA output for the first constraint of the query $freq(f, A) \geq 2 \wedge freq(f, B) \leq 1$.

The complete solutions for the example query is defined by these boarders. This means, that all subfragments of fragments in S that contain one of the fragments in G, are part of the solution *sol* = {`C-C, C-C-C, C-C-C-O, C-O, C-C-O,` …}. Thus *S* and *G* together compactly represent the set of all solutions.

Results and Discussion

In this section, we briefly summarize our experiments with the DTP AIDS Antiviral Screen (http://dtp.nci.nih.gov) dataset and with toxicological *Structure-Activity Relationships* (SARs) using MOLFEA-generated features.

```
LEVEL 1:
Candidates:         C, N, O, c (4)
Frequent:           C, N, O, c (4)
Infrequent:         (0)
==
LEVEL 2:
Candidates:         C-C, C-N, C-O, C-c, C=C, C=N, C=O, C=c, C#C, C#N, C#O, C#c, N-N,
                    N-O, N-c, N=N, N=O, N=c, N#N, N#O, N#c, O-O, O-c, O=O, O=c, O#O,
                    O#c, c-c, c=c, c#c (30)
Frequent:           C-N, C-c, c-c  (3)
Infrequent:         C-C, C-O (2)
==
LEVEL 3:
Candidates:         N-C-N, N-C-c, C-N-C, C-c-c, c-C-c, C-c-C, c-c-c  (7)
Frequent:           C-c-c, c-c-c  (2)
Infrequent:         (0)
==
LEVEL 4:
Candidates:         C-c-c-c, C-c-c-C, c-c-c-c  (3)
Frequent:           c-c-c-c  (1)
Infrequent:         C-c-c-c (1)
==
LEVEL 5:
Candidates:         c-c-c-c-c  (1)
Frequent:           c-c-c-c-c  (1)
Infrequent:         ( 0 )
==
LEVEL 6:
Candidates:         c-c-c-c-c-c  (1)
Frequent:           c-c-c-c-c-c  (1)
Infrequent:         (0)
==
LEVEL 7:
Candidates:         c-c-c-c-c-c-c  (1)
Frequent:           (0)
Infrequent:         (0)
==========================================================================
G:  C-C, C-O, C-c-c-c  (3)
S:  C-C-C-O, N-C-C-O, C-C-C-c-c-c-c-c-c, N-C-C-c-c-c-c-c-c  (4)
==========================================================================
```

Figure 3. MOLFEA output for the second constraint of the query *freq(f, A)* ≥ 2 ∧ *freq(f, B)* ≤ 1. In dealing with this constraint MOLFEA starts from the results in Figure 2.

NCI DTP AIDS ANTIVIRAL SCREEN DATABASE

The NCI DTP AIDS Antiviral Screen program has checked more than 40,000 compounds for evidence of anti-HIV activity. The screen utilizes a soluble formazan assay to measure protection of human CEM cells from HIV-1 infection (12). Compounds were classified as either confirmed active (CA, providing protection), confirmed moderately active (CM, not reproducibly providing protection), or confirmed inactive (CI). In our experiments, class CA consisted of 417 compounds, class CM of 1069 compounds, and class CI of 40,282 compounds. The available database (October 1999 Release) contains the screening results for 43,382 compounds.

The aim of this experiment was to find fragments that are, statistically significant, over represented in the active class (CA) and under-represented in the inactive (CI).

The following query was posed to the system: $(freq(f,\ CA) \geq 13) \wedge (freq(f,\ CI) \leq 516)$. The thresholds in the queries were determined as follows: The minimum frequency threshold in these queries corresponds to 3 % of the active molecules. In order to determine the maximum allowable frequency in the non-active molecules, we used the X^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. In this way, we obtained a maximum frequency of 516 in inactive compounds for the first task, and a maximum frequency of 8 in the moderately actives for the second. Given these frequencies, the occurrence of a fragment in the active class is not due to chance at a significance level of 0.999.

For this task, the total computation time was 19,212.31 CPU seconds (measured in CPU seconds on a Linux PC with a Pentium III 600 MHz processor). The first part (the minimum frequency query) took only 1,544.09 CPU seconds, and the second part (the maximum frequency query) took 17,668.22 CPU seconds. The boundary set *G* contained 222 elements, and *S* contained 314 elements. This contrasts with a total of 1,623 patterns in the solution space bounded by *G* and *S*, which demonstrates the utility of *G* and *S* sets (version spaces) in this kind of application.

In the minimum frequency part of the query, the longest solution fragment had a length of 24 atoms, the longest fragment found in the maximum frequency part had a length of 22 atoms. So, it has been shown that MOLFEA can search for very long patterns in a structural database of over 40,000 compounds.

Table 1.

G	*S*
`O-C-n:c:n:c=O`	`N=N=N-C-C-C-n:c:c:c:n:c=O`
`O-C-C-C-C-C-n:c=O`	`N=N=N-C-C-C-n:c=O`
`C-C-C-O-C-n:c:n:c:c:c`	`c:c:c:n:c:n-C-C-C-N=N=N`
`C-c:c:n:c:n:c=O`	`C-c:c:n:c:n-C-C-C-N=N=N`
`N-c:c:c-S`	`C-c:c:n-C-C-C-N=N=N`
`N-C-C-C-O-C-C-O`	`C-C-O-C-C-N=N=N`
`C-C-C-O-C-n:c=O`	`N=N=N-C-C-O-C-n:c:n:c=O`
`O-C-n:c:c:c:n:c=O`	`N=N=N-C-C-O-C-n:c:c:c=O`
`N=N=N`	`N=N=N-C-C-C-n:c:n:c=O`
`N-c:c:c:c:c-s`	`N=N=N-C-C-C-n:c:c:c=O`

From the boundary sets *G* and *S*, we picked fragments based on their class distribution (statistical significance and accuracy). Table 1 summarizes the most significant samples. The majority of these fragments, e.g.

`N=N=N-C-C-C-n:c:c:c=O` and `N=N=N-C-C-C-n:c:n:c=O`

indicate compounds that are derivatives of Azidothymidine (AZT, Retrovir, Zidovudine, 3'-Azido-3'-deoxythymidine, CAS 30516-87-1, see Figure 4), a potent inhibitor of HIV-1 replication, which is widely used in the treatment of HIV infection. Other fragments indicate another class of reverse transcriptase inhibitors, mainly thiocarboxanilide derivatives, which are, according to our knowledge, drugs that are still in an experimental phase. The automated rediscovery of the most important classes of anti-HIV drugs indicates the utility of the presented approach

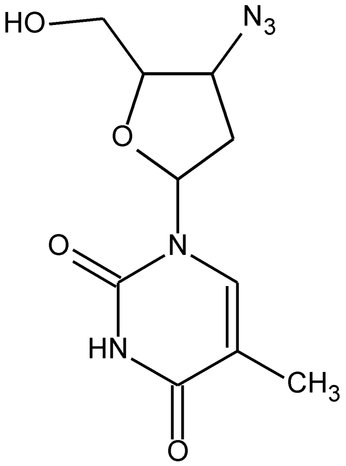

Figure 4. Chemical Structure of Azidothymidine (AZT)

USE OF MOLFEA FEATURES FOR SAR

In a series of other experiments (11), we have employed MOLFEA-generated features in structure-activity relationship prediction. SAR prediction in this context works in three steps. In a first step, MOLFEA queries are used to construct a set of fragments. These queries can be class-blind (e.g. when we require a minimum frequency on the *whole* dataset), or class-sensitive, when we consider separate criteria for active and inactive molecules.

The resulting fragments are used as binary features or fingerprints (a fragment either occurs in a molecule or it does not) to describe the molecules. The resulting data sets can be fed into a traditional data mining system [such as e.g. WEKA (13)], to obtain a predictive model for the biological effect under investigation. This method can be combined with virtually any data mining technique; we have induced decision trees, classification rules, regression models and Support Vector Machines (SVMs).

In the experiments sketched in (11), we have investigated the effects of class-sensitive vs. class-blind fragment construction on benchmark datasets for biodegradability, mutagenicity and carcinogenicity prediction (14, 15, 16). Class-sensitive feature construction is performed using combined minimum and maximum frequency queries as described above. Class-blind feature construction is performed by a simple minimum frequency query.

The predictive accuracies obtained with MOLFEA generated features turned out to be at least competitive with the best published results in the literature so far. Support Vector Machines were able to take advantage of a large number of features (fragments) constructed in a class-blind manner, whereas classical inductive Machine Learning approaches (decision trees and rules) seemed to benefit from class-sensitive feature construction. Summing up, these experiments clearly demonstrated the utility of MOLFEA-generated features in the induction of SARs.

RELATED WORK

Molecular fragments are, among other purposes, useful and important for the the induction of Structure-Activity Relationships. The use of automatically derived structural fragments in SARs originates from the CASE/MultiCASE systems developed by (17).With more than 150 published references, the CASE/MultiCASE systems are the most extensively used SAR and predictive toxicology systems. Previous approaches in these areas are based on the "decomposition" of individual compounds: these methods generate *all* fragments occurring in a given single compound. In this regard, our contribution is a language that enables the formulation of complex queries regarding fragments – users can specify precisely which fragments they are interested in. We also implemented a solver to answer queries in this language. Thus, from the algorithmic point of view, it is no longer necessary to process the results of queries post-hoc.

Molecular fragment finding has also been studied within the context of inductive logic programming and knowledge discovery in databases. For instance, WARMR (18) and the approach by Inokuchi *et al.* (19) have been used in this context. WARMR is a system discovering requently succeeding Datalog queries, and thus is not restricted to fragments. The approach by Inokuchi *et al.* deals with arbitrary frequent subgraphs, and thus is not restricted to linear ragments. Both approaches differ in that their pattern domain is more expressive, but finding requent patterns is likely to be more expensive and complex than for linear fragments.

Finally we want to stress again, thatMOLFEA is only one instance of the general *Inductive Database Framework*. It is quite easy to adapt Inductive Databases for a new application domain. We have presently implemented PROFEA(6), that analyses the secondary structure of proteins and we are working on a further instance for gene expression data.

Table 2.

	Learning	Class	
Domain	Algorithm	Blind	Sensitive
Carcinogenicity	C4.5	64.3	65.9
	PART	**67.4**	65.9
	Log	65.6	65.3
	1. SVM	65.3	65.0
Mutagenicity	C4.5	90.4	87.2
	PART	91.5	93.1
	Log	94.7	**95.7**
	1. SVM	94.7	92.0
Biodegradation	C4.5	77.7	76.5
	PART	77.1	79.9
	Log	80.2	75.9
	1. SVM	**81.1**	75.9

FURTHER DEVELOPMENTS

MOLFEA is presently capable to find linear sequences of atoms in databases with chemical structures. It is presently impossible, to identify stereochemical effects, or arrangements in three-dimensional space. We are therefore working on several extensions to the MOLFEA framework.

Abstractions The concept of fragments is not limited to sequences of elements. It is fairly easy to define abstract *atom types*, that can be more general (e.g. H-bond donor/acceptor) or more specific (e.g. oxygen in a carbonyl group) than the elements. Another addition will be the introduction of wildcards for atoms. With these extensions it will be possible to find fragments like "two H-bond donors separated by 5 heavy atoms".

Branched Fragments The extension towards branched fragments is conceptually easy, from a computer science viewpoint, but it might result in increased search times. The use of branched fragments is particularly attractive from a chemist's viewpoint, because with their help it will be possible to identify stereochemical effects.

3D Fragments Another extension of MOLFEA is the consideration of three-dimensional arrangement of atoms in fragment finding. Work on this topic is almost completed and will be the subject of a separate publication.

CONCLUSIONS

We have presented a novel database and data mining approach based on the concept of inductive databases. Even though our framework was presented for string-like patterns, it should be clear that one could easily adapt it towards richer data structures such as e.g. graphs, or towards other application domains in bio- and chemo-informatics. We have also argued that the inductive database framework is useful for knowledge discovery in databases in general and in bio- and chemo-informatics in particular. The authors hope that the work on MOLFEA and PROFEA will stimulate other researchers to add inductive query languages to the many existing databases in bio- and chemo-informatics. This in turn should allow scientists to understand their data more easily and to discover new knowledge more effectively.

ACKNOWLEDGMENTS

This work is partly supported by the European Union IST FET project cInQ.

REFERENCES

[1] Fayyad, U., Piatetsky-Shapiro, G., Smmyth, P. (1996). From Data Mining to Knowledge Discovery: An Overview. In *Advances in Knowledge Discovery and Data Mining*, AAAI Press, Menlo Park, Calif., pp. 1-30.

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

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

[4] De Raedt, L. (2000). A logical database mining query language. In *Proceedings of the 10th Inductive Logic Programming Conference*, Lecture Notes in Artificial Intelligence, Vol. 1866, Springer Verlag.

[5] Boulicaut, J. F., Klemettinen, M., Mannila, H. (1998). Querying Inductive Databases: A Case Study on the MINE RULE Operator. In *Proceedings of PKDD-98*, Lecture Notes in Computer Science, Vol. 1510, Springer Verlag, pp. 194-202.

[6] Fischer, J. (2002). *Objektorientiertes Design einer induktiven Datenbank und eine Anwendung des Levelwise Version Space Algorithmus auf die Sekundärstruktur von Proteinen*. Studienarbeit at the Machine Learning Lab, University of Freiburg, Germany.

[7] De Raedt, L., Kramer, S. (2001). The level wise version space algorithm and its application to molecular fragment finding. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, Morgan Kaufmann.

[8] Kramer, S., De Raedt, L., Helma, C. (2001). Molecular feature mining in HIV data, in: *Proc. of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-01)*, 136-143.

[9] Agrawal, R., Imielinski, T., Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of ACM SIGMOD Conference on Management of Data*.

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

[11] Kramer, S. & De Raedt, L. (2001). Feature construction with version spaces for biochemical applications. *Proceedings of the 18th International Conference on Machine Learning*, 258-265, Morgan Kaufmann.

[12] Weislow, O. S., Kiser, R., Fine, D. L., Bader, J. P., Shoemaker, R. H., Boyd, M. R. (1989). 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.

[13] Witten, I. & Frank, E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

[14] D¡zeroski, S., Blockeel, H., Kompare, B., Kramer, S., Pfahringer, B., Van Laer, W. (1999). Experiments in predicting biodegradability. In *Proceedings of the 9th International Workshop on Inductive Logic Programming (ILP-99)*, 80-91, Springer Verlag.

[15] Srinivasan, A., Muggleton, S., King, R. D., Sternberg, M. (1996). Theories for mutagenicity: a study of first-order and feature based induction. *Artificial Intelligence* **85**(1-2):277-299.

[16] Srinivasan, A., King, R. D., Bristol, D. W. (1999). An assessment of submissions made to the predictive toxicology evaluation challenge. *Proc. of IJCAI-99*, 270-275.

[17] Rosenkranz, H. S., Cunningham, A. R., Zhang, Y. P., Clayhamp, H. G., Macina, O. T., Sussmann, N. B., Grant, S. G., Klopman, G. (1999). Development, characterization and application of predictive toxicology models. SAR and QSAR in *Environmental Research*, **10**:277-298.

[18] Dehaspe, L. & Toivonen, H. (1999). Discovery of frequent datalog patterns. In *Data Mining and Knowledge Discovery Journal*, **3**(1):7-36.

[19] Inokuchi, A., Washio, T., Motoda, H. (2000). An Apriori-based algorithm for mining frequent substructures from graph data. In D. Zighed, J. Komorowski, and J. Zyktow (eds.) *Proceedings of PKDD 2000*, Lecture Notes in Artificial Intelligence, Vol. 1910, Springer Verlag.