Machine Learning, 30, 241–273 (1998)

Pharmacophore Discovery using the Inductive Logic Programming System PROGOL

PAUL FINN finnpw@pfizer.com
Computational Chemistry, Pfizer Central Research, Ramsgate Road, Sandwich, Kent CT13 9NJ, U.K.

STEPHEN MUGGLETON stephen@minster.cs.york.ac.uk
Department of Computer Science, University of York, Heslington, York YO1 5DD, U.K.

DAVID PAGE * cdpage@louisville.edu
Department of Engineering Mathematics and Computer Science, Speed Scientific School, University of Louisville, Louisville, KY 40292, U.S.A.

ASHWIN SRINIVASAN ashwin@comlab.ox.ac.uk
Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, U.K.

Editor: Ron Kohavi and Foster Provost

Abstract. This paper presents a case study of a machine-aided knowledge discovery process within the general area of drug design. Within drug design, the particular problem of *pharmacophore discovery* is isolated, and the Inductive Logic Programming (ILP) system PROGOL is applied to the problem of identifying potential pharmacophores for ACE inhibition. The case study reported in this paper supports four general lessons for machine learning and knowledge discovery, as well as more specific lessons for pharmacophore discovery, for Inductive Logic Programming, and for ACE inhibition. The general lessons for machine learning and knowledge discovery are as follows.

1. An initial rediscovery step is a useful tool when approaching a new application domain.
2. General machine learning heuristics may fail to match the details of an application domain, but it may be possible to successfully apply a heuristic-based algorithm in spite of the mismatch.
3. A complete search for all plausible hypotheses can provide useful information to a user, although experimentation may be required to choose between competing hypotheses.
4. A declarative knowledge representation facilitates the development and debugging of background knowledge in collaboration with a domain expert, as well as the communication of final results.

Keywords: inductive logic programming, pharmacophore, structure-activity prediction

1. Introduction

This paper presents a case study of a machine-aided knowledge discovery process within the general area of drug design. The case study focuses on three portions of the knowledge discovery task. The first is identification of a part of a larger task—in this case, drug design—for which machine learning might be useful. The selected subtask is *pharmacophore discovery*, which is recognized by chemists as

* Please address correspondence to this author.

an important problem in its own right. A *pharmacophore* is a 3-D substructure of a molecule that is responsible for its medicinal activity. The second portion of knowledge discovery that we address is the application of an existing machine learning algorithm to this selected task. Third, this paper considers the interaction between the users of the machine learning system and the domain expert, viewed as a client.

Our usage of the term *knowledge discovery* assumes a slightly stronger definition than the following (Fayyad et al., 1996).

> Knowledge discovery in databases is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.

In particular we assume that the knowledge involved is both *declarative* rather than procedural as well as being *necessarily* comprehensible and insightful to a domain expert. The latter condition is related to the view of the domain expert as a client who provides requirements for knowledge representation, much like the notion of requirement specification within software engineering. Within the present domain, the requirements that the knowledge should be both declarative and comprehensible stem from the fact that the knowledge involved should provide 3-dimensional structural insights to be used by a synthetic chemist involved in devising new drugs. Such relational knowledge descriptions are known within the drug design literature as pharmacophores. Successful applications of a machine learning technique to problems related to pharmacophore discovery have been discussed previously (Jain et al., 1994a, Jain et al., 1994b). The domain expert (first author) in the present study suggested a more explicit representation for pharmacophores (Section 2.2).

This paper describes a series of experiments providing insights at four different levels. First, regarding Angiotensin-Converting Enzyme (ACE) inhibition the experiments confirm an earlier proposal, generate an alternative proposal, and suggest that no other alternatives exist within the constraints provided by the domain expert. Second, the experiments suggest a general methodology for pharmacophore discovery using Inductive Logic Programming (ILP). Third, the experiments indicate that ILP can be used successfully to learn 3-dimensional concepts and to deal naturally with the *multiple instance problem* (Dietterich et al., 1997) (see Section 2.3). Fourth, the experiments support four lessons regarding knowledge discovery in general, which follow.

Rediscovery Step: An initial rediscovery step is useful when beginning work in a new problem domain. It can help build the confidence of a domain expert and can provide a thorough test of the encoding of domain or background knowledge. A *blindfold* rediscovery step, in which the users of the machine learning system do not know the correct answer in advance, provides a particularly realistic test.

Mismatch of ML Heuristics with Problem Domains: General purpose machine learning heuristics sometimes do not match particular problem domains.

In the experiments reported in the present paper, the minimum description-length principle conflicts with a domain-specific preference for more complex pharmacophores.

Complete Search and Automatic Experiment Proposal: Machine learning systems often return a single hypothesis for a given set of data, even when other plausible hypotheses are available. The present work illustrates the value of instead performing a complete search for a set of plausible hypotheses. It also highlights the need for automatic proposal of experiments to discriminate between competing plausible hypotheses.

Declarative Knowledge Representation: Using a declarative knowledge representation facilitates the development and debugging of background knowledge in collaboration with a domain expert.

These lessons are discussed in detail in Section 4, together with concomitant recommendations for research into machine learning and knowledge discovery.

The paper is organized as follows. Section 2 describes in detail the problem of pharmacophore discovery, as well as describing other work on this problem and related problems. Section 3 describes the series of experiments regarding ACE inhibition. Section 4 discusses the contributions of this paper, including general lessons for research into knowledge discovery and particular lessons for inductive logic programming, for pharmacophore discovery, and for the study of ACE inhibition.

2. Background: Domain description and prior work

2.1. Drug design process

Drug molecules work by binding (coming together in close association) to "target sites" within the body. These sites commonly are protein molecules, either enzymes or receptors. By interaction with these protein molecules drugs can modulate their actions.

Typically, the process of drug design begins with the identification of an appropriate target with which drugs could interact to modulate disease. Closely connected with this process is the identification of a "lead" molecule or molecules. These molecules may be identified in a number of ways, for example from large scale empirical testing of available chemicals. They will have some activity, i.e., ability to interact with the target, but may do so only weakly and may possess other undesirable properties, for example metabolic instability. Chemists synthesize and test related molecules, possibly generating improved leads, until ultimately a molecule of the desired activity and properties is discovered. This molecule can then begin the long process of development, including safety and efficacy testing in clinical trials. If successful—and many are not—it finally will become a marketed product.

Computational techniques are used throughout this process, but rational drug design has concentrated on the section between the identification of the lead molecule

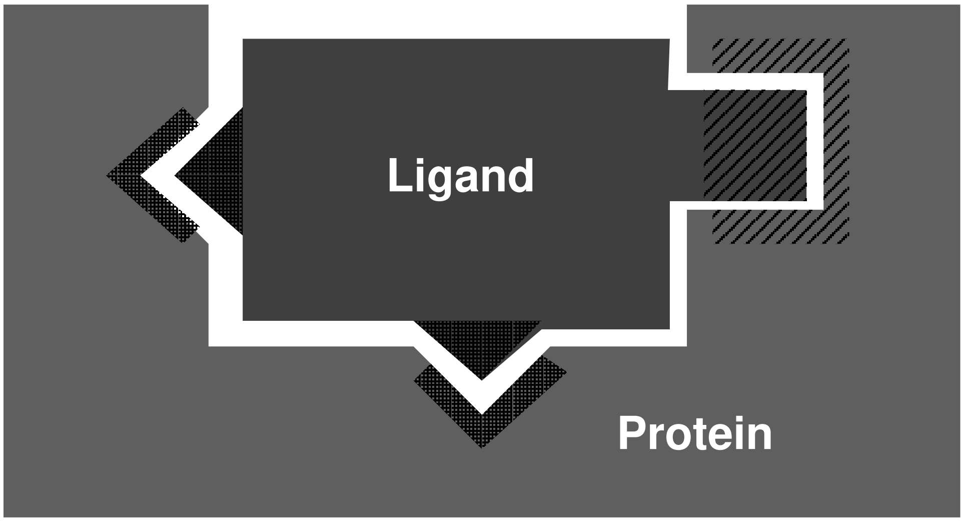

Figure 1. A schematic representation of a drug/protein interaction, illustrating complementarity of shape and three complementary property interactions, one hydrophobic (striped) and two hydrogen bonds (dotted).

and the entry of a candidate into development. This section of the process focuses on answering the question, "Why does the lead molecule possess the desired activity?" The general principles of drug activity are reasonably well understood, but answering this question for any specific case can be very difficult.

A ligand[1] will bind with its target if it is complementary to it (see Figure 1). This complementarity consists of two parts:

Complementarity of shape: This allows interaction between the drug and target over a large area. From a computational viewpoint, as we shall see below, shape complementarity is complicated by the fact that most drug molecules are flexible and thus can adopt a variety of shapes.

Complementarity of properties: The strength of the interaction is determined by the ability to form a variety of weak interactions, primarily hydrophobic and electrostatic interactions (hydrogen bonds and interaction between oppositely charged groups). These interactions are individually weak, but their effects can sum to a strong overall interaction if they are in sufficient number and appropriately positioned.

Starting from a lead molecule, how does one develop candidate drugs? In some cases, the detailed structure of the target binding site is known, and this knowledge can be used directly to design improved molecules (Whittle and Blundell, 1994). However, in the majority of cases, this information is unavailable and design must rely on what can be inferred from the structures of the ligands themselves and their biological activities. The approach, therefore, is to try to elucidate this "structure-activity relationship" (SAR) for the molecules. Analysis of the how the structural differences of a set of ligands affects their activity can lead to the discovery of an SAR. This information can be used to suggest new molecules to make, which should have enhanced activity.

2.2. Representation of the activity model

2.2.1. QSAR: Advantages and limitations Many methods have been used to represent the structure-activity relationships of molecules. The first to be developed—the so-called "traditional" quantitative structure-activity relationship (QSAR) methods—correlate properties calculated from the structures of the molecules with their activities. These properties can describe aspects of the whole molecule, such as the partition coefficient, the molecular volume, the number of rings, etc.; in cases where the molecules share a common core, these properties can describe the substructures at the positions of variation. Such models have been widely applied, and there are many examples of successful QSAR analyses (Hansch and Leo, 1995). While the correlations derived by such techniques can be used in the design of improved compounds, doing so is not always straightforward. Parameters such as electronic partial charges and connectivity index values are easy to calculate for a given structure, but it is much harder to design a molecule that will possess these values. Also, it is not always possible to relate the parameters to the principles of drug-receptor interaction described above. In addition, because QSAR methods operate on bulk molecular properties rather than explicit representations of the 3D structures of molecules, they do not take into account some structural information that can affect molecular activity. Nevertheless, one advantage of traditional QSAR methods is that they avoid having to deal with the problem of molecular flexibility, which we now describe.

Molecules can adopt different shapes (conformations) by torsional rotations about bonds within the molecule. Molecules rapidly convert from one conformation to another, so that no single conformation can be isolated and tested for a given activity (e.g., ACE inhibition). Hence in general one does not know *a priori* which conformation of a molecule is an active conformation. Each conformation is associated with an energy level, which arises from interactions within the molecule and between the molecule and its environment. Only low-energy conformations are accessible at normal temperatures.

Any sample of a given compound contains molecules in a variety of different conformations. The concentration of a given conformation in a sample is related by an inverse exponential function to the energy of that conformation, so only very low-energy conformations appear in significant concentration. Computational techniques are available for calculating the energy of a given conformation, and for searching the conformational space to identify the low-energy conformations (Leach, 1991). Any one of these conformations could be the one which binds to the target protein. The larger and more flexible the molecule is, the more low-energy conformations it will have. One of the key problems for the approaches that have been adopted to analyze and predict biological activity is how to take this flexibility into account. Traditional QSAR techniques ignore molecular flexibility by describing only non-3D properties of molecules—properties that do not vary as a molecule changes conformation.

2.2.2. 3D QSAR techniques 3D-QSAR methods work within a correlation framework, but use an explicit representation of the molecule. The input is a series of molecules with varying levels of activity. This method represents shape and electrostatic interactions more directly via a calculated interaction with a "probe" atom or group at points on a three-dimensional grid which surrounds the molecules. Statistical methods are then used to identify those parts of the molecule which are responsible for activity. The analysis can be displayed graphically to aid in the design of new molecules. The major disadvantage with this method is that, in order to compare values at the calculated points between molecules, the conformation of the molecules and a common coordinate frame, or alignment, must be chosen in advance of the analysis. This is equivalent to deciding the manner in which the molecules interact with the target. If the molecules contain a large common structural element this alignment may be straightforward, but this is often not the case.

The COMPASS algorithm (Jain et al., 1994a, Jain et al., 1994b) overcomes these problems by using a more sophisticated representation of molecular shape, neural network learning methods and adaptation of the alignments. The models produced can be used to predict the activity of new molecules, and, again, visual representations of the results can aid compound design. However, the interpretation of the neural network may not be easy. Furthermore, the human brain has difficulty in imagining complex three-dimensional shapes, which makes it difficult to design molecules that are very different structurally from the known examples.

2.2.3. Pharmacophores A very commonly used representation of biological activity, and the one used in the present work, is the pharmacophore. This is an abstraction of the molecular structure to the, usually, small number of key features which contribute the majority of the activity, together with their geometric arrangement represented by pairwise distances (see Figure 2). These features relate directly to the interactions (hydrophobic, electrostatic, etc.) described above. In this view, the remainder of the molecule is useful only as scaffold, holding the pharmacophore groups in the correct spatial positions. Many methods of pharmacophore identification have been described in the literature. In the active analogue approach (Mayer et al., 1987), it is necessary to identify the equivalent pharmacophore groups in each molecule in advance. Conformations of each of a series of active molecules are then sought which place these groups in a common spatial arrangement. The need to identify the groups in advance limits the utility of this approach. The more general DISCO method (Martin et al., 1993) uses a clique detection algorithm (Brint and Willett, 1987) to search for common sets of inter-feature distances within a group of active molecules. Tolerances on the distance

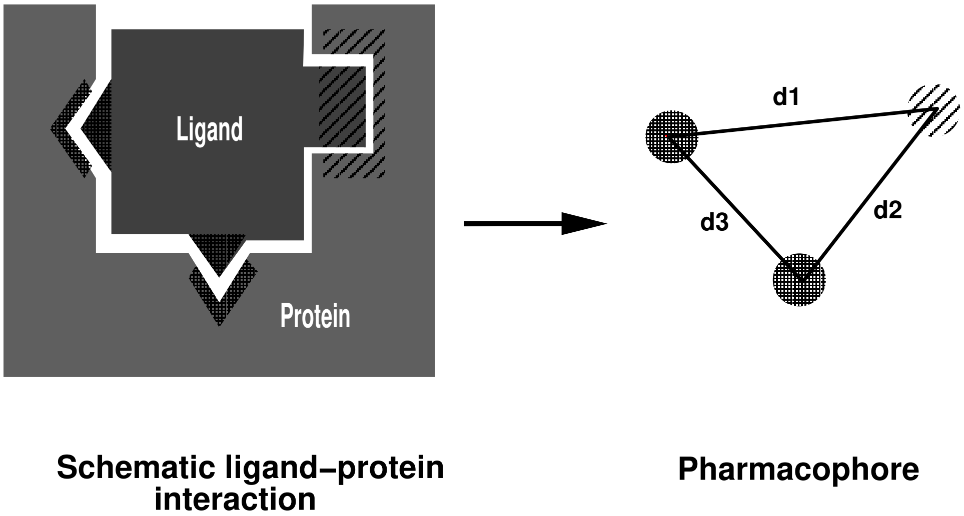

Figure 2. The pharmacophore definition on the right describes the ligand-protein interaction on the left. The pharmacophore identifies the key functional interactions (regions are coded as in Figure 1), and it expresses the geometric relationships between the sites of these interactions as distances d1, d2, and d3. For example, d1 might be 3.75 Ångstroms, d2 might be 4.5 Ångstroms, and d3 might be 5.0 Ångstroms.

matches enable the method to use discrete conformations and to model variation allowed by the pharmacophore. Our experience has been that, although DISCO can perform well in many cases, there are difficulties in cases where either the number of compounds and/or the numbers of conformations for each compound is large. The main problems have been large computation times and large numbers of returned pharmacophores. This has stimulated our exploration of alternative pharmacophore discovery methods.

An advantage of the pharmacophore representation is that it expresses biological activity in a language that is familiar to chemists within the pharmaceutical industry. These representations are also readily convertible into search queries of compound databases. Posing such queries to a database of compounds is an effective means of identifying additional active molecules (Finn, 1996). Because of these advantages, we have adopted a data input and output representation that is similar to DISCO. The input to the learning algorithm is a set of conformations for each molecule in the set. These conformations can be generated by whatever method is appropriate. The output is a pharmacophore expressed in terms of (1) certain types of atoms or functional groups in the molecule that are necessary for binding, e.g., "hydrogen bond acceptor" or "hydrophobic group", and (2) the distance relationships between these atoms or groups.

Thus far we have motivated the pharmacophore discovery problem, and we have described other approaches to this and related problems. The present paper describes the first application of ILP to pharmacophore discovery. The following subsection sets this work in the context of earlier related applications of ILP.

2.3. Related work using ILP

Pharmacophore discovery can be viewed as a particular case of structure-activity prediction, the goal of which is to learn to predict the activities of molecules based on their structures. The earliest technique used for structure-activity prediction was linear regression, first employed for this task by Hansch and colleagues in 1962 (Hansch et al., 1962). In recent years ILP has been used successfully for structure-activity prediction, first with a "1-dimensional" feature-value representation similar to that used with linear regression, and later with a "2-dimensional" chemical structure representation. The present work on pharmacophore discovery builds on these earlier applications of ILP by moving to a more complex "3-dimensional" representation of molecules. The remainder of this section sets the present work in the context of the earlier applications of ILP to structure-activity prediction.

2.3.1. Learning with a 1D representation In ILP's first successful application to a structure-activity prediction problem (King et al., 1992), the GOLEM program (Muggleton and Feng, 1990) was used to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. The training data consisted of 44 trimethoprim analogues and their observed inhibition of *Escherichia coli* dihydrofolate reductase. Eleven additional compounds were used as unseen test data. GOLEM obtained rules that were statistically more accurate on the training data and on the test data than a previously published linear regression model. We refer to the representation used in this work, as well as with linear regression, as a "1-dimensional" representation because an example molecule is represented as a vector of variables. The variables describe either whole molecule properties or substituent groups on a common structural backbone. Such a representation was possible for this problem because all of the molecules shared a common structure. Figure 3 shows the shared structure, or template, for all the molecules, as well as one particular instance of the template.

2.3.2. Learning with a 2D representation In more recent work (King et al., 1996) the 2D bond-and-atom molecular descriptions of 229 aromatic and heteroaromatic nitro compounds (Debnath et al., 1991) were given to the ILP system PROGOL. Such compounds frequently are mutagenic. It is of considerable interest to the pharmaceutical industry to determine which molecular features result in compounds having mutagenic activity. The problem of predicting mutagenicity is addressed by another paper in the present special issue (Lee et al., 1998). The study was confined to the problem of obtaining structural descriptions that discriminate drugs with positive mutagenicity from those which have zero or negative mutagenicity. A set of 8 optimally compact rules were automatically discovered by PROGOL. These rules suggested 3 previously unknown features leading to mutagenicity. The molecules

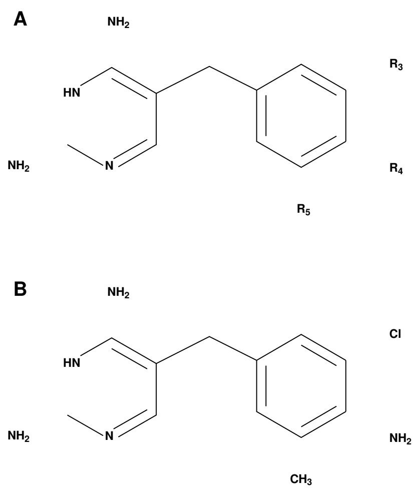

Figure 3. The family of analogues in the first ILP study. A) Template of 2,4-diamino-5(substituted-benzyl)pyrimidines: R3, R4, and R5 are the three possible substitution positions. B) Example compound: $R_3 - -Cl$; $R_4 - -NH_2$; $R_5 - -CH_3$

used in this study could not have been represented easily by a 1D feature-vector because they were a diverse set with no common structural backbone. Attempts to force the 2D representation into a set of features resulted in the generation of examples containing more than a million features each.

2.3.3. Challenges of a 3D Representation ILP has been applied successfully to structure-activity prediction problems using 1D and 2D representations of molecules. However, the 3D aspects of molecules (or their *stereochemistry*) are often crucial in determining molecular activity. Therefore the feeling of some chemists is that widespread successful application of ILP or any other machine learning technique to structure-activity prediction requires a 3D representation. The present paper reports the first attempt to use a 3D representation within ILP (a preliminary version of this work was presented at the Seventh International Workshop on Inductive Logic Programming (Muggleton et al., 1996)). The move to a 3D representation within ILP brings two important challenges, which we now summarize.

The first challenge is in choosing how to represent 3D geometric concepts in first-order logic. In summary, the approach taken in the present work is to represent geometry via pairwise distances between points. Of course if a hypothesis requires a precise floating point value for the distance between any two points, then few if any examples will be labeled *positive* by that hypothesis. Therefore we allow a degree of tolerance in the matching of distances. In the present work, the precise degree of tolerance is suggested by the domain expert, although another option is to let the learning algorithm itself vary the tolerance levels.

The second challenge is the *multiple instance problem*, a general issue first raised by Lathrop, Dietterich, and Lozano-Perez (Dietterich et al., 1997). They also were motivated by the context of drug design (their neural network-based system COMPASS was described earlier in this section), and the problem is probably easiest to explain in this concrete context. As noted in the domain description earlier in this section, a molecule can take on a variety of 3D conformations, and it is quite possible that a molecule with a desired biological activity exhibits that activity in only one of its conformations. But because molecules rapidly interconvert between conformations, no single conformation can be isolated and tested. The set of conformations of a molecule corresponds to a set of examples, but we do not have sufficient knowledge to label each individual example as *active* or *inactive*—we can label only the set as a whole. Thus a hypothesis explaining activity should label a set *active* just if it predicts that at least one example in the set is active. In general terms, the multiple instance problem occurs when individual examples cannot be labeled, but instead *sets* of examples are labeled, such that a set is labeled *positive* if and only if at least one member of the set is positive.

Lathrop, Deitterich, and Lozano-Perez raise the multiple instance problem as a general machine learning problem, although they do not cite any other real-world domains where it arises. We can support their claim of its generality by citing at least one additional domain. In recent work Saith *et al.* (Saith et al., 1997) used C4.5 to learn a decision tree for choosing embryos to return to a mother's womb after *in vitro* fertilization. Under British law only three embryos can be implanted at one time, although as many as eight may be available. To maximize the probability of success, doctors wish to choose the most viable embryos—those most likely to lead to a healthy pregnancy—and a decision tree for choosing embryos can be learned from previous cases. But again examples are grouped into sets, this time into sets of three. A set is labeled *positive* just if at least one of the embryos in the set (without our ability to know which one) results in success—parents taking home a baby—and *negative* otherwise.

The form of uncertainty embodied in the multiple instance problem did not arise in earlier ILP applications using a 1D or 2D representation of molecules, because only whole-molecule properties (e.g., molecular weight) and aspects of the bond-and-atom molecular structure were represented. A 3D representation provides additional information that can be central to molecular activity, yet this additional information is inherently disjunctive: a molecule can adopt any of several low-energy conformations. The approach to the multiple instance problem that is taken in the present work is different from approaches taken in either of the other aforementioned papers. In the work on COMPASS, an entirely new neural network algorithm was developed. In the work on *in vitro* fertilization using decision-tree techniques, examples within a set were averaged (using mean, median, or mode values of features as deemed appropriate) to yield a single example instead. It turns out that a relational representation allows the multiple instance problem to be addressed in a surprisingly straightforward way, without the need to develop a new learning algorithm tailored to the problem and without the loss of information that can come from averaging over examples in a set. Section 4.2 describes the general approach, building on details given in Section 3.6 for the specific case of pharmacophore discovery.

3. Experimentation

This section describes a sequence of four experiments using the ILP system PROGOL (Muggleton, 1995) to discover a pharmacophore for ACE inhibition. ACE (Angiotensin-Converting Enzyme) inhibitors are a widely-used form of medication for hypertension. PROGOL is used in such a way that it returns human-comprehensible pharmacophore descriptions which also can be visualized. The techniques used in these experiments appear to be of sufficient generality to apply to other problems of pharmacophore discovery. We begin this section with an overview of PROGOL—providing only the level of detail needed to understand the experiments—and an overview of the sequence of experiments.

3.1. PROGOL

Because PROGOL is an ILP system, it learns first-order definite clause theories, or logic programs.[2] PROGOL takes as input positive examples and (if available) negative examples of a target concept, as well as a background theory, all in Horn clause form. An example is redundant if it is positive and is already provable from the background theory, and an example is contradictory if it is negative and is provable from the background theory. PROGOL removes any redundant examples and expects the user to remove contradictory examples. In addition to examples and a background theory, PROGOL takes as input a description of the hypothesis space, in the form of a specification of the kinds of clauses that are acceptable. In learning a single clause, PROGOL performs a complete search of the hypothesis space, using pruning similar to that used in $A*$ (Nilsson, 1980), to find a clause c that maximizes a compression function. The compression function is $f(c) = P(c) - N(c) - L(c)$, where $P(c)$ is the number of positive examples that can be proven by the clause c taken together with the background theory, $N(c)$ is the number of negative examples that can be proven in the same way, and $L(c)$ is the size (number of literals) of c. In addition, the user may specify a maximum acceptable value of $N(c)$ (such as 0, in a noise-free case), which PROGOL will respect.

PROGOL takes a greedy set cover approach to learning multiple-clause theories. Nevertheless, this issue is of little consequence for the present paper because we adopt a one-clause representation of pharmacophores, described later in this section. A multiple-clause theory for ACE inhibition therefore would represent a disjunction of pharmacophores, such that a molecule is an ACE inhibitor just if it exhibits at least one of the pharmacophores. A disjunction of pharmacophores suggests multiple binding sites, such that binding to any of the sites produces the desired biological activity. Such a multiple-clause theory is not desired for ACE inhibition or for most pharmacophore discovery problems.

PROGOL's search is a *refinement graph search* (Shapiro, 1983), the details of which are beyond the scope of this paper. Nevertheless, two further aspects of PROGOL's operation are important for this paper. Rather than providing a technical discussion of these aspects (Muggleton, 1995), we provide a high-level description together with a discussion of their significance for the problem of pharmacophore discovery. In this discussion we consider single-clause learning only.

During learning, PROGOL focuses on a single positive example and constructs a "bottom" clause containing everything, subject to language constraints, that is true of that positive example according to the background theory. For theoretical reasons (Muggleton, 1995), the bottom clause can be used to direct PROGOL's search without sacrificing completeness. For the problem of pharmacophore discovery, the bottom clause effectively identifies all the potential pharmacophoric points in the first active molecule, and also records the distance between each pair of these points. The search then proceeds by beginning with the "empty" pharmacophore (0 points) and constructing progressively more complex pharmacophores from the points and distances in the bottom clause. The constructed pharmacophores are tested on the remaining molecules. This approach of basing a search on a "bottom" clause gives PROGOL significant efficiency advantages for the problem of pharmacophore discovery, because it substantially prunes the search space before the search even begins.

The second aspect of PROGOL that is significant for the problem of pharmacophore discovery is its complete search. PROGOL is guaranteed to find all clauses that maximize the compression function. For the problem of pharmacophore identification, this means PROGOL will find all potential pharmacophores (given the hypothesis language and background knowledge provided) that are consistent with the data. Hence, not only does PROGOL provide a user with potential pharmacophores, but it also tells the user that no other pharmacophore is possible within the constraints given.

It was the decision of the domain expert (first author) at Pfizer to use ILP for pharmacophore discovery since a pharmacophore is most easily described through the use of 3D relations, which are easily represented using logic programs. We expect that relational learning systems other than PROGOL could be applied successfully in this domain as well. PROGOL was chosen in part simply because it was developed by some of the present authors and therefore would be easier to modify if necessary. Nevertheless, PROGOL's complete search based on a bottom clause turns out to be particularly well-suited to pharmacophore discovery; this match is discussed in Sections 4.2 and 4.3.

3.2. Overview of the experiments

A pharmacophore was proposed for ACE inhibition eleven years ago by Mayer and colleagues (Mayer et al., 1987). Earlier modeling studies had investigated the ACE pharmacophore, but based on a small number of compounds (Hassell et al., 1982, Andrews et al., 1985). The first experiment described in the present paper was a blindfold test to see whether PROGOL could rediscover the pharmacophore proposed by Mayer *et al.*, given their particular assumptions (regarding the active 3D conformations and zinc-binding geometry, described in the next subsection). The test was a blindfold test in that the knowledge engineers were not told in advance the proposed pharmacophore.

In most pharmacophore discovery applications, a particular active 3D conformation for each molecule cannot be assumed, but rather it can be assumed only that for any active molecule at least one of its low-energy conformations (Section 2.1) is active. The second experiment was designed to test whether PROGOL could discover a new pharmacophore and/or re-discover the original proposed pharmacophore when provided with multiple low-energy 3D conformations for each molecule. It was not evident that PROGOL would, or even should, rediscover the original proposed pharmacophore in such an experiment, because the conformations used by Mayer *et al.* are not all energy-minimized, nor are their zinc-binding geometries ideal. In this experiment PROGOL in fact was unable to identify *any* pharmacophore common to all ACE inhibitors. Several possible explanations for this were considered. In the end, the knowledge engineers found in their encoding of chemical background knowledge an apparently incorrect assumption they had made. This assumption appeared to be the reason for failure to find any pharmacophore. This type of 'failed' experiment usually is not reported in applications papers. We include the description of this experiment, as well as the next experiment, because they are instructive regarding the knowledge engineering issues that often arise in a real machine learning application.

In the third experiment, PROGOL identified a single pharmacophore, distinct from the original proposed pharmacophore. Nevertheless, this new pharmacophore was deemed unreasonable by the domain expert. It is unreasonable because the points in the pharmacophore are too close to each other and are likely to give rise to "steric hindrance." After careful consultation with the domain expert, it was discovered that the assumption in Experiment 2 deemed incorrect by the knowledge engineers actually was correct, but that another incorrect assumption was encoded in the background theory. This incorrect assumption was responsible for PROGOL proposing no reasonable potential pharmacophore.

In the final experiment, with the debugged background theory, PROGOL identified 28 potential pharmacophores, but many arose from only small perturbations in the geometries of others. The set of 28 pharmacophores can be condensed into a pair of distinct proposed pharmacophores for ACE inhibition. One of these pharmacophores is the same (modulo a small geometric perturbation) as the pharmacophore found in the first experiment, thus confirming the proposal of Mayer *et al.* (Mayer et al., 1987). The second is an interesting alternative that deserves further attention. Furthermore, because PROGOL performs a complete search, this result says that there are no further alternatives to these two pharmacophores, given the constraints imposed.

3.3. Experiment 1: A blindfold test

The first experiment was a blindfold test of PROGOL's ability to learn a pharmacophore. The knowledge engineers were provided with data and relevant chemical information, but they were not told the proposed pharmacophore. This experiment tested the following hypothesis.

Hypothesis 1: Given the conformations and zinc binding sites proposed by Mayer and colleagues (Mayer et al., 1987), as well as definitions of hydrogen acceptors, hydrogen donors, and hydrophobic groups, PROGOL will re-discover the pharmacophore proposed by Mayer *et al.*

3.3.1. Data The data used in this experiment consisted of 28 compounds known to exhibit the activity of ACE inhibition. This data set was precisely the one used by Mayer *et al.* Compounds in the set vary in size from 24 atoms to 70 atoms. The compounds were identified by the names *m1*, *m2*, ..., *m28*. Their activity as ACE inhibitors was asserted to PROGOL via first-order atomic formulae, or Prolog-style facts, of the form *active(m1)*, *active(m2)*, ..., *active(m28)*. Note that no negative examples were used, although negative examples can be incorporated easily if they are available. The size of this data set is typical of data sets used by computational chemists.

3.3.2. Background knowledge The background knowledge used in this experiment can be divided into the following three types.

1. compound-specific knowledge
2. general chemical and geometric knowledge
3. constraints on legitimate pharmacophores

Compound-specific knowledge The compound-specific knowledge included the atom and bond structure of each compound, as well as its 3-dimensional conformation. This information was represented by first-order atomic formulae, or Prolog-style facts. A fact of the form

```
atm(m1,a1,o,2,3.43,-3.12,0.05)
```

asserts that molecule *m1* has an atom which we will call *a1* that is an oxygen, is *sp2-hybridized* (a detail that turns out to be of no consequence), that is at position $(3.43, -3.12, 0.05)$ in three-dimensional space, for the given conformation and orientation of the molecule. The 3D grid is in units of Ångstroms. Other atom types include *c* for carbon, *n* for nitrogen, *h* for hydrogen, *s* for sulphur, and *p* for phosphorus. A fact of the form

```
bond(m1,a2,a3,2)
```

asserts that molecule *m1* has a bond between atoms *a2* and *a3*, and that bond is a double bond.

As noted in Section 2.1, molecules can arrange themselves in a number of 3D conformations. In this experiment, for each molecule the background knowledge recorded only the conformation that Mayer and colleagues believed permits binding (and thus exhibits the activity of ACE inhibition). For later experiments, the assumption that the Mayer *et al.* conformation is the active one was removed, and multiple low-energy conformations were used for each molecule.

General chemical and geometric knowledge Binding into a metalloproteinase such as ACE often involves formation of a traditional (covalent) bond to a metal ion. But more generally, binding almost always involves the formation of weaker *hydrogen bonds*. Hydrogen bonds are simply the attraction between a hydrogen atom with a slight positive charge and an atom such as oxygen or nitrogen with a slight negative charge. A hydrogen atom bears a positive charge if it is covalently bonded to a more *electronegative* atom, such as oxygen or nitrogen, that draws the shared electron pair (constituting the covalent bond) away from the hydrogen atom; this also has the effect of giving the other atom a partial negative charge. For example, water has a higher boiling point than substances such as propane or butane partially because of hydrogen bonds that form between the oxygen atom of one water molecule and a hydrogen atom of another water molecule, holding the two molecules together. The hydrogen atom in a hydrogen bond is called a *hydrogen donor*, while the other atom is called a *hydrogen acceptor*.

We can now describe the general chemical knowledge supplied by the domain expert. The expert noted that hydrogen donors and acceptors are potentially important for any pharmacophore discovery task, and metal ion sites are potentially relevant to any task (such as this one) involving a metalloproteinase. Furthermore, hydrophobic (water repelling) groups such as benzene or alkane chains or rings are often important for pharmacophore discovery. It was straightforward to encode in definite clause form the expert's definitions for all these potentially relevant items. In addition, the expert requested that a pharmacophore be expressed as its *points* (hydrogen donors, hydrogen acceptors, and zinc sites) and their geometric arrangement, described by the pairwise distances among them. A molecule is said to exhibit the pharmacophore if it has atoms or groups that match with each of the points in the pharmacophore, such that all pairwise distances agree with the pharmacophore distances to within one Ångstrom. Finally, the expert noted that pharmacophores generally need at least three points to be useful.

Constraints on pharmacophore descriptions Given the requirements of the domain expert, the following clause would be an acceptable description of a pharmacophore.

```
active(X):- hdonor(X,A), hacc(X,B), zincsite(X,C), dist(X,A,B,3.0,1.0),
            dist(X,A,C,4.0,1.0), dist(X,B,C,5.0,1.0)
```

This clause asserts that a molecule X is active if it has a hydrogen donor A, a hydrogen acceptor B, and a zincsite C, such that the distance between A and B is 3.0 +/- 1.0 Ångstroms, the distance between A and C is 4.0 +/- 1.0 Ångstroms, and the distance between B and C is 5.0 +/- 1.0 Ångstroms.

In general, the head of any clause describing a pharmacophore should be *active(X)*, while the body should specify the points of the pharmacophore via the predicates *hdonor*, *hacc*, *hydrophobic*, and *zincsite*, and should also specify the distance between each pair of points via the *dist* predicate. Furthermore, the clause should specify at least three points for the pharmacophore. These constraints are specified within PROGOL; the PROGOL input file is available from the contact author on request.

In fact, at the outset of this experiment it was not possible to encode domain-specific constraints such as these in PROGOL. This experiment motivated a general change to the PROGOL interface and algorithm. This general change is the capability to express a *declarative bias*, a restriction on acceptable hypotheses that goes beyond the language bias imposed by the representation vocabulary. A user can now write arbitrary Prolog clauses to specify explicitly which kinds of clauses within the given vocabulary are or are not acceptable as hypotheses. Already these general constructs have proven useful in a variety of other applications of PROGOL as well as the current one. Full details of the mechanism for expressing declarative bias are beyond the scope of this paper but can be found elsewhere (Srinivasan and Camacho, 1996).

3.4. Methodology

Having specified the data and the background knowledge, the methodology for this blindfold trial was nearly as simple as, "(1) Run PROGOL with the data and background knowledge given, and (2) compare the result with the proposed pharmacophore of Mayer *et al.*" Only one modification was made to this methodology, which we now motivate. The motivation is applicable to not only PROGOL but any compression-driven or MDL-like algorithm.

A mismatch exists between PROGOL's compression heuristic (and compression-driven search in general) and pharmacophore search. In general, larger pharmacophores that are common to a set of active molecules are more interesting than smaller ones, because large shared structures are less likely to occur by chance. Furthermore, the existence of a large shared structure implies the existence of several smaller shared structures. For example, if 28 molecules share a four-point pharmacophore then they share four three-point pharmacophores which can each be obtained by deleting one point from the four-point pharmacophore. Yet although the larger pharmacophores are of greater interest, their representations are textually longer and therefore compression scores them as less interesting. The result is that PROGOL will never return the larger pharmacophores. This mismatch between compression and pharmacophore search extends beyond pharmacophore search to any application domain where textually longer hypotheses are *a priori* less likely to be consistent with random data, that is, less likely to be true by chance.

The mismatch between compression and pharmacophore search was addressed by the following change to the methodology. PROGOL was run repeatedly, first to search for three-point pharmacophores, then four-point pharmacophores, etc. In general with this approach, once a number n is reached such that no n-point pharmacophore is found, the search can be terminated—if no n-point pharmacophore exists then no $(n+1)$-point pharmacophore exists. This approach can be viewed as a *manual wrapper* around PROGOL. A manual wrapper is one of the potential solutions mentioned in Section 4 when a general machine learning heuristic does not match a particular problem domain.

3.5. The result

In the experiment PROGOL found four three-point pharmacophores and one four-point pharmacophore that are common to all 28 ACE inhibitors. The four three-point pharmacophores are all obtained from the four-point pharmacophore by omitting one point, so the four-point pharmacophore is the most interesting. The largest potential five-point pharmacophore appears in only 10 of the 28 ACE inhibitors, this arising from adding one point to the four-point pharmacophore. No other five-point pharmacophore appears in more than 4 of the 28 molecules.

It is a straightforward matter to translate clauses of a pre-defined form, such as those representing pharmacophores, into English; PROGOL has the necessary instructions to do so. Therefore, we present the PROGOL-generated English description of the four-point pharmacophore here. This four-point pharmacophore was judged equivalent to the Mayer *et al.* proposed pharmacophore by the domain expert.

```
Molecule A is an ACE inhibitor if:
   molecule A can bind to zinc at a site B, and
   molecule A contains a hydrogen acceptor C, and
   the distance between B and C is 7.9 +/- 1.0 Angstroms, and
   molecule A contains a hydrogen acceptor D, and
   the distance between B and D is 8.5 +/- 1.0 Angstroms, and
   the distance between C and D is 2.1 +/- 1.0 Angstroms, and
   molecule A contains a hydrogen acceptor E, and
   the distance between B and E is 4.9 +/- 1.0 Angstroms, and
   the distance between C and E is 3.1 +/- 1.0 Angstroms, and
   the distance between D and E is 3.8 +/- 1.0 Angstroms.
```

The ability to generate an English description of the found pharmacophore, in a form commonly used by chemists, is a significant advantage of this approach. In addition, this approach makes visualization a relatively simple matter. Because the pharmacophore is also represented as a logical clause, a logic programming query can be executed to identify the pharmacophoric points within each molecule. Once these points are identified, each molecule can be displayed with the instance of the pharmacophore labeled. Figure 4 shows two of the 28 molecules with the points of this pharmacophore labeled; notice that the labels correspond to the variable names in the English description of the pharmacophore.

This experiment confirmed the hypothesis that PROGOL can re-discover the Mayer *et al.* pharmacophore given their original assumptions regarding the active conformation and the zinc site. In addition, another result of the experiment was a thoroughly-tested background theory for use in the further experiments.

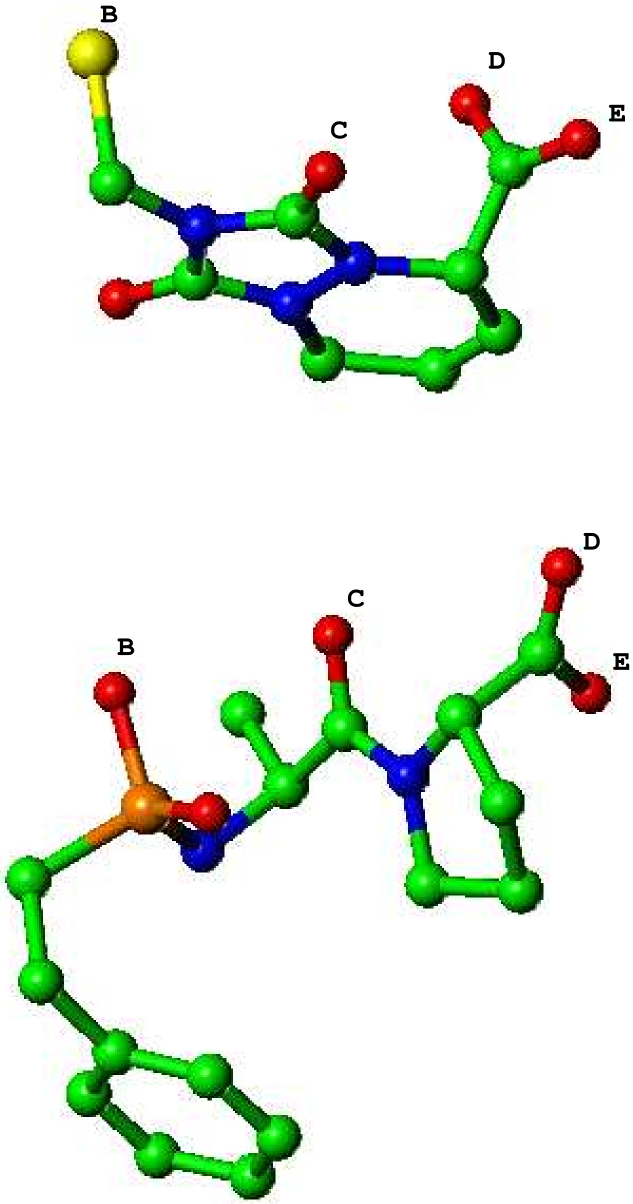

Figure 4. ACE inhibitor numbers 1 (top) and 10 with highlighted 4-point pharmacophore. Molecular structures have been simplified by the removal of hydrogens.

3.6. Experiment 2: Multiple conformations and multiple zinc sites

We have already identified the two shortcomings of the previous blindfold trial in assessing our approach to pharmacophore discovery. The first shortcoming is the unrealistic assumption of a single known active conformation for each active molecule. In a realistic experiment, a set of the lowest-energy conformations should be used instead, and the only assumption should be that the molecule will exhibit the desired activity in at least one of these conformations. These low-energy conformations can be estimated by a computational chemist using molecular modeling software. The second shortcoming is the assumption of a single known site for the zinc ion in ACE relative to the active molecule during binding. In a realistic experiment, a set of possible sites should be used instead. A zinc site is identified by first finding a *functional group* within the molecule that is capable of binding to zinc, and then calculating where the zinc ion should be in geometric relationship to this group for binding to occur. Sufficient chemical knowledge exists to identify both the functional groups that can bind to zinc and the geometry of ideal binding. The experiment we now describe implemented these changes and tested the following hypothesis.

Hypothesis 2: Given the ten lowest-energy conformations for each ACE inhibitor (as estimated by a computational chemist using molecular modeling software), and given potential zinc sites as computed according to general chemistry knowledge, PROGOL will identify at least one pharmacophore for ACE inhibition.

The data for this experiment was the same as for the previous one. Only the background knowledge changed for this experiment. We now describe this background knowledge.

3.6.1. The background knowledge Again we divide the background knowledge into three parts: compound-specific knowledge, general chemical and geometric knowledge, and constraints on the hypothesis.

Compound-specific knowledge Compound-specific knowledge was encoded in the same way as for Experiment 1, except that an additional argument was added to each fact to specify the conformation. The structure of a molecule was repeated for each conformation, and a given atom of the molecule might have a different location in each conformation. For example, the following facts give the details of atom *a1* in conformations *c1* and *c10* of molecule *m28*. The second argument of each fact provides the unique conformation identifier.

```
atom(m28,c1,a1,n,am,-0.79,-3.78,4.13).

atom(m28,c10,a1,n,am,-1.34,-4.05,0.77).
```

General chemical and geometric knowledge The additional chemical knowledge used in this experiment concerned zinc binding. The domain expert described

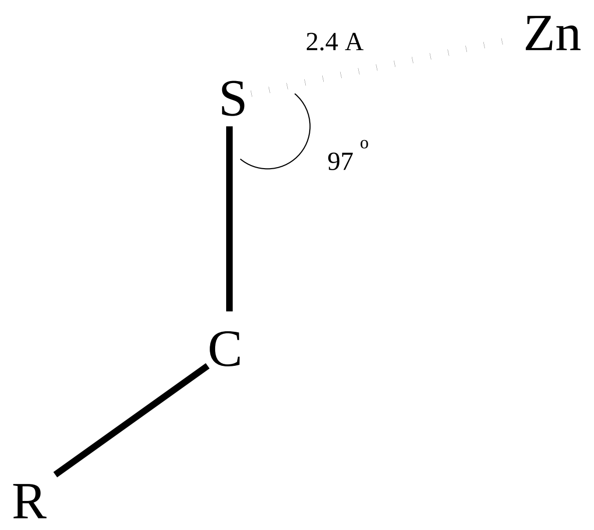

Figure 5. Ideal geometry for a thiol group binding to zinc. The length of the sulphur–zinc bond is 2.4 Ångstroms, and the carbon–sulphur–zinc angle is 97 degrees. The torsion angle is most easily described by example. A torsion angle of 0 or 180 has zinc in the plane defined by the sulphur, carbon, and R (the next connected atom in the rest of the molecule); by convention, 0 degrees is taken to have the zinc and R in the same side (half-plane) of the sulphur–carbon bond. A 90 degree torsion angle has the zinc coming directly out from the page. Thus in the figure the torsion angle is 180 degrees.

five functional groups that can bind to zinc, four of which appear in the molecules in the given data set. Functional groups turn out to be surprisingly natural to represent in definite clause form. Code for these functional groups is available from the contact author.

Further chemical knowledge from the domain expert specified where the zinc must be located relative to the functional group. For example, for a thiol group (Figure 5) the ideal binding geometry locates zinc at 2.4 Ångstroms from the sulphur atom to which it binds, at a 97 degree angle to the $C - S$ (carbon-sulphur) bond, and at a torsion angle to the $R - C - S$ plane of 0 or 180 degrees, plus or minus at most 60 degrees.

In addition to this chemical knowledge, geometric knowledge was encoded to compute the zinc site. The same code was used to compute the zinc site relative to each functional group that can bind to zinc. In each case points analogous to R, C, and S for thiol were used together with ideal bond length, bond angle, and torsion angle(s) for the given group in order to compute an ideal zinc site. Sites within 2 Ångstroms of another atom in the molecule were eliminated, because these would not be possible due to steric hindrance (crowding).

Constraints on pharmacophore descriptions Given the addition of conformational information, the following is an example of an acceptable clausal representation of a pharmacophore.

```
active(X):- hdonor(X,Y,A), hacc(X,Y,B), zincsite(X,Y,C),
            dist(X,Y,A,B,3.0,1.0), dist(X,Y,A,C,4.0,1.0),
            dist(X,Y,B,C,5.0,1.0)
```

This clause asserts that a molecule X is active if it has a conformation Y, and it has a hydrogen donor A, a hydrogen acceptor B, and a zincsite C, such that the distance between A and B *within conformation* Y is 3.0 +/- 1.0 Ångstroms, the distance between A and C *within conformation* Y is 4.0 +/- 1.0 Ångstroms, and the distance between B and C *within conformation* $Y$is 5.0 +/- 1.0 Ångstroms.

In general, the same requirements as before were placed on clauses, with one additional requirement: the conformational variable in the body literals must be the same throughout the clause. Without this requirement, PROGOL might return a clause of the following form.

```
active(X):- hdonor(X,Y,A), hacc(X,Y,B), zincsite(X,Y,C),
            dist(X,Y,A,B,3.0,1.0), dist(X,Z,A,C,4.0,1.0),
            dist(X,W,B,C,5.0,1.0)
```

Notice that the conformational variables in the last two literals are distinct from the conformational variable, Y, in the rest of the clause. Such a clause is easier to satisfy since we need only find some conformation in which the distance between A and B is roughly 3 Ångstroms, some (possibly different) conformation with a distance of roughly 4 Ångstroms between A and C and some (possibly different again) conformation with a distance of roughly 5 Ångstroms between B and C. Such a clause does not represent a true pharmacophore.

3.6.2. The result The result of the experiment was that no pharmacophore appeared in all 28 ACE inhibitors. Specifically, no pharmacophore appeared in more than 26 of the ACE inhibitors. This result seemed to indicate that Hypothesis 2 was false and that PROGOL could not successfully meet the challenges of pharmacophore prediction with multiple conformations. Nevertheless, before considering Hypothesis 2 to be disproven, an examination of the possible reasons for this result was in order. The following are the potential alternative explanations identified by the authors on first seeing the result.

1. For the two unexplained ACE inhibitors, perhaps additional *stereoisomers*[3] were included in the sample that was tested for activity. Such mixtures are common because often it is difficult to isolate a stereoisomer.

2. Perhaps some low-energy conformations were omitted for the 2 unexplained ACE inhibitors.

3. An error may have been made in the background theory.

The first possible explanation was ruled out by the domain expert after a review of the methodology for testing the 28 compounds for ACE inhibition. The second possible explanation also was ruled out by the domain expert, for the following reason. While some additional conformations could be added for some of the molecules in the data set, the two unexplained molecules are small and rigid, so that no other relatively low-energy conformations are possible.

Within the background theory the following assumption had been made by the knowledge engineers without checking with the domain expert.

Co-reference Assumption A hydrogen acceptor within a pharmacophore cannot also be the atom that binds to zinc.

The co-reference assumption could have dramatic consequences, because oxygen is generally a hydrogen acceptor and also can bind to zinc when it appears in one of two common functional groups—carbonyls and carboxylates. Recognition of the co-reference assumption led to Experiment 3, which we now describe.

3.7. Experiment 3

The data and background knowledge for Experiment 3 were the same as for Experiment 2, with the exception that an atom could now be used as both a hydrogen acceptor and the atom that binds to zinc within a pharmacophore. The result of this Experiment was that a single four-point pharmacophore was found; it is shown here.

```
Molecule A is an ACE inhibitor if for some conformation B:
        A contains a hydrogen acceptor C,
        A contains a hydrogen acceptor D,
        the distance between C and D within conformation B is 3.2 +/- 1.0 Angstroms,
        A contains a hydrogen acceptor E,
        the distance between C and E within conformation B is 4.0 +/- 1.0 Angstroms,
        the distance between D and E within conformation B is 2.2 +/- 1.0 Angstroms,
        A can bind to zinc at a site F,
        the distance between C and F within conformation B is 3.9 +/- 1.0 Angstroms,
        the distance between D and F within conformation B is 2.0 +/- 1.0 Angstroms,
        the distance between E and F within conformation B is 3.1 +/- 1.0 Angstroms.
```

Nevertheless, this pharmacophore was judged by the domain expert to be unreasonable because the zinc site is too close to the hydrogen acceptors. Further conversation revealed that the co-reference assumption, although not checked with the domain expert, actually is sensible. Therefore this assumption was reinstated.

On closer examination, a problem was identified with the definition of thiol. The definition of thiol initially required that the carbon atom single-bonded to sulphur must also be bonded to two hydrogen atoms. Such bonding is unnecessary and the definition was amended accordingly. Both of the unexplained ACE inhibitors contain thiol groups in which the carbon is not bonded to two hydrogens. This error was found while sitting with the chemist in front of the computer and together going through the code defining the functional groups that can bind to zinc. Such a mode of debugging was facilitated by the declarative nature of the code. This correction to the background theory lead to Experiment 4.

3.8. Experiment 4

In Experiment 4, the data and background knowledge were the same as for Experiment 3, except for the changes just described. The result of Experiment 4 was that 28 pharmacophores were found to appear in all 28 ACE inhibitors. These form two groups that can be represented by one pharmacophore each. The criterion for grouping is that any pharmacophore in the group can be converted to the representative pharmacophore by modifying its distances by less than one Ångstrom. The two pharmacophores are as follows.

```
Molecule A is an ACE inhibitor if for some conformation B:
        A contains a hydrogen acceptor C,
        A contains a hydrogen acceptor D,
        the distance between C and D within conformation B is 3.2 +/- 1.0 Angstroms,
        A contains a hydrogen acceptor E,
        the distance between C and E within conformation B is 4.0 +/- 1.0 Angstroms,
        the distance between D and E within conformation B is 2.2 +/- 1.0 Angstroms,
        A can bind to zinc at a site F,
        the distance between C and F within conformation B is 5.5 +/- 1.0 Angstroms,
        the distance between D and F within conformation B is 7.1 +/- 1.0 Angstroms,
        the distance between E and F within conformation B is 8.5 +/- 1.0 Angstroms.

Molecule A is an ACE inhibitor if for some conformation B:
        A contains a hydrogen acceptor C,
        A contains a hydrogen acceptor D,
        the distance between C and D within conformation B is 3.2 +/- 1.0 Angstroms,
        A contains a hydrogen acceptor E,
        the distance between C and E within conformation B is 4.0 +/- 1.0 Angstroms,
        the distance between D and E within conformation B is 2.2 +/- 1.0 Angstroms,
        A can bind to zinc at a site F,
        the distance between C and F within conformation B is 3.9 +/- 1.0 Angstroms,
        the distance between D and F within conformation B is 6.1 +/- 1.0 Angstroms,
        the distance between E and F within conformation B is 7.3 +/- 1.0 Angstroms.
```

The two pharmacophores differ in the position of the zinc site relative to the hydrogen acceptors. This result provides three items of information about ACE inhibition. First, it confirms the Mayer *et al.* result. Even though the Mayer *et al.* zinc sites do not obey ideal binding geometry and even though some of their conformations are not energy-minimized, their result is valid when using ideal binding geometries and energy-minimized conformations. Second, the result provides an alternative potential pharmacophore that merits further investigation. A recently published modeling study of ACE inhibition based on superposition of a small number of conformationally constrained inhibitors (Bohacek et al., 1996) also identified two models differing primarily at the zinc site. The models appear similar to ours, but there is insufficient information in the paper to enable a detailed comparison. The third item of information the result provides is that these two pharmacophores are the only ones possible (modulo small perturbations of the distances) given the assumptions encoded in our background theory.

For machine learning methods that perform a complete search, as PROGOL does, computational complexity is a particular concern. Indeed the worst-case time complexity for PROGOL on a pharmacophore discovery problem is exponential in the size of the target pharmacophore; a worst-case time complexity analysis is available in an on-line appendix. (In fact, it is straightforward to prove an idealized version of pharmacophore discovery to be NP-complete, although it is unclear whether the idealization can be modified to take into account all the relevant constraints from chemistry and biology.) To be more specific, the time complexity of PROGOL on a pharmacophore discovery problem involving a target pharmacophore with k points is $O(n^k cmp^k)$, where n is the number of potential pharmacophoric points in the molecule with the fewest such points, c is the number of conformations for this molecule, m is the total number of conformations for all other molecules, and p is the number of potential pharmacophoric points in the molecule with the most such points. Nevertheless, in practice PROGOL's search space often is much smaller than the worst-case search; this occurs in the present work, where the total run time of PROGOL in Experiment 4 (the most time-consuming) was only 20.4 minutes on a Sun SPARC 20. In addition to PROGOL's search strategy, two decisions helped keep the runtime low. First, zinc sites and pairwise distances between potential pharmacophoric points were precomputed. This avoided unnecessary repetition of costly (though still polynomial-time) computations. Second, three-point pharmacophores were first discovered by PROGOL, and these were combined by PROGOL into four point pharmacophores. It should be noted that this second decision actually increases the worst-case time complexity, although in practice it yields reduced computation times.

It is worth noting that large total numbers of molecules and conformations are not terribly damaging to the PROGOL approach even in the worst case, since the algorithm's time complexity is linear in the variable m. Clearly k is the most damaging number, but for many pharmacophore discovery problems a relatively low value of k can be assumed (pharmacophores of more than four or five points are rarely found by existing techniques). If k is taken to be a constant (say 3 or greater), then the most damaging variable in the worst case becomes p, the number of potential pharmacophoric points in the molecule with the most such points. This number is particularly high when multiple hypothesized points, such as zinc sites, must be computed, since this can raise p from a typical value of around 10 or 15 to a typical value of around 50 or 60. To the extent that conclusions can be drawn from a worst-case bound, this bound indicates that pharmacophore discovery problems where hypothesized sites are unnecessary should lead to lower run times than in the present work.

4. Scientific contributions

The series of experiments described in this paper provide insights at four different levels. First, regarding Angiotensin-Converting Enzyme (ACE) inhibition the experiments confirm an earlier proposal, provide an alternative proposal, and suggest that no other alternatives exist within the constraints provided by the domain expert. Second, the experiments suggest a general methodology for pharmacophore discovery using Inductive Logic Programming (ILP). Third, the experiments indicate that ILP can be used successfully to learn 3-dimensional concepts and to deal naturally with the *multiple instance problem* (Dietterich et al., 1997) (see Section 2.3.3). Fourth, the experiments support four lessons regarding knowledge discovery in general. We now discuss these contributions in further detail in reverse order, beginning with the most general contributions.

4.1. Lessons and recommendations for knowledge discovery

4.1.1. Lesson 1: Rediscovery step The first lesson from the experiments reported in this paper is the value of an initial rediscovery step when beginning work in a new problem domain. In the present work, Experiment 1 is an attempt to rediscover a proposed pharmacophore for ACE inhibition, given particular assumptions made by the original proposers. A rediscovery step is useful for at least two reasons. First, it builds the confidence of the domain expert. Even if the domain expert is not skeptical, a successful rediscovery can help to generate further enthusiasm about potential benefits. Second, a rediscovery step provides a thorough test of the encoding of domain or background knowledge. Both of the potential benefits of a rediscovery step can be fully achieved only if it is carried out as a "blindfold trial," in which the users of the machine learning algorithm do not know the target concept. Experiment 1 in the present work was conducted as a blindfold trial. If a blindfold trial initially is unsuccessful, the failure may be due simply to bugs in the background knowledge. If so, this will be easier to determine with a test case where the domain expert knows what the answer should be than in a case of *de novo* discovery.

The use of blindfold trials is not original with this work. To our knowledge the first use of a blindfold trial in knowledge discovery by machine learning occurs in the work on META-DENDRAL (Buchanan et al., 1972), where the task was to learn to identify particular organic compounds from within a family of compounds (e.g., estrogens) given their mass spectrometry readings. Although the authors do not explicitly state that their trials were blindfold trials, this seems evident given their discussion. Initial information (background knowledge) was elicited from experts and provided to META-DENDRAL, and the expert was then asked to judge the acceptability of the META-DENDRAL results. We suspect there are a number of other cases where blindfold rediscovery steps have been used, but they simply do not get discussed explicitly very often (Provost and Aronis, 1996).

Recommendation 1: We recommend further study into the efficacy of blindfold trials when beginning to apply machine learning to a new application domain. It is hoped that such study also will lead to recommendations regarding the best procedure to follow. For example, if the rediscovery step initially is unsuccessful, how should the researchers proceed? Also, should a new application begin with two rediscovery tasks, the first of which is not blindfold and is used to develop the approach, while the second is a blindfold test?

4.1.2. Lesson 2: Mismatch between ML heuristics and problem domains General purpose machine learning heuristics sometimes do not match particular problem domains. In the present work the MDL principle conflicts with chemists' intuition that the most complex pharmacophore common to all active molecules is most likely to be correct, since it is the least likely to be present due to chance alone. The following three approaches can be taken when a mismatch occurs between general heuristics and a particular problem domain.

Wrapper: This is the approach taken in the present work. We have used the machine learning algorithm in a way that allows us to give preference to more complex pharmacophores. A wrapper can either be automatic, in the form of a program that repeatedly calls the machine learning algorithm, or manual, as in the present work.

Domain-specific Heuristic: An algorithm can be modified to make use of an alternative, domain-specific heuristic. More generally, machine learning systems can be built which allow a user to plug in domain-specific heuristics. These heuristics might modify how the search is done or might simply modify the scoring of hypotheses. Srinivasan and Camacho describe an ILP algorithm that minimizes a user-defined cost function (Srinivasan and Camacho, 1997).

New General Heuristic: Work within one or more domains might cause an algorithm designer to see an alternative general-purpose heuristic that is better suited to a number of domains. For example Muggleton recently has designed a heuristic that trades coverage of positive examples with specificity, measured according to a random, unlabeled data set (Muggleton, 1996). Desirable hypotheses have high coverage over the positive examples but relatively low coverage over the random data. It appears likely that use of this heuristic in place of MDL would eliminate the mismatch in the present work and possibly other problem domains, although this has not yet been tested.

Recommendation 2: We recommend further investigation into the mismatches that arise when applying general-purpose machine learning heuristics to new problem domains. Can the heuristics be improved to circumvent some such mismatches? Would we obtain better knowledge discovery systems if we allowed specialized problem-dependent heuristics to be specified for each new problem domain, or are there major benefits that come from committing a system to a general-purpose heuristic?

4.1.3. Lesson 3: Complete search and automatic experiment proposal Many machine learning algorithms return a single hypothesis for a given set of data, even when other plausible hypotheses are available. In some cases it may be possible to encode user-defined criteria for acceptable hypotheses or to score hypotheses according to user-defined criteria. In such cases it can be advantageous to return multiple hypotheses that meet the given criteria or that achieve a high score. This occurs in the present paper, where two pharmacophores for ACE inhibition are found. In addition, if the algorithm performs a complete search, it is possible to assert that no other reasonable hypotheses exist, given the user's criteria. When multiple hypotheses are returned, however, we run the risk of providing a client with too many solutions. It might be possible in such situations for the machine learning algorithm to propose experiments that can distinguish between competing hypotheses. For example, in the present work if the ILP system PROGOL had access to a database of compounds, it potentially could search the database for compounds possessing one but not both of the competing pharmacophores for ACE inhibition. It could then propose testing one or more of these compounds for ACE inhibition, to distinguish between the two competing pharmacophore hypotheses.

Recommendation 3: We recommend a research program to test the following claim.

Claim: Automatic proposal of experiments by a machine learning system to distinguish between competing hypotheses is feasible for real-world applications.

4.1.4. Lesson 4: Declarative knowledge representation Because of the nature of ILP systems, the background knowledge we employed was largely declarative. We consider it only "largely declarative" because even Prolog code can sometimes be written is a less declarative, more procedural form. For example, one part of our background knowledge computes potential sites for a zinc atom given the descriptions of chemical groups that can bind to zinc. This portion of the background knowledge is highly procedural, although written in Prolog, and includes code for intersecting planes and spheres, for Gaussian elimination in 3 variables, etc. Fortunately, writing and debugging this portion of the code did not require interaction with the chemist. By contrast, the part of the background knowledge that defines chemical groups that can bind to zinc was highly declarative, being almost a direct translation of pictorial descriptions provided by the chemist. As a result, where ambiguities or questions arose, or where debugging was necessary, these issues could be addressed by actually going through that portion of the background knowledge with the chemist. In addition to a declarative representation for background knowledge, having a declarative representation for hypotheses made it a single day's task to write code both to translate hypotheses into English and to display molecules with the pharmacophore highlighted.

The preceding discussion raises the point that expressions and portions of code cannot be labeled *declarative* or *non-declarative* simply because of the representation language employed. As a first attempt at an improved definition, which we hope may stimulate further discussion, we propose the following.

Definition 1. A portion of code in some representation language is *declarative* if it is isomorphic to a comprehensible expression in natural language.

To make crisp the notion of *isomorphic*, we require the existence of an algorithm to translate statements from the representation language into natural language statements. The notion of *comprehensible* is inherently subjective, but a relatively objective test could be achieved by (1) asking a sample of people to read the natural language description and (2) testing their comprehension. Notice that with this formulation the relationship to natural language is a property of the representation language, whereas comprehensibility is a property of both the representation language (with its translation procedure) and the particular portion of code.

We have admitted that part of the background knowledge in the present work was not particularly declarative. Another shortcoming in the present work is in the *declarative bias*, or the specification of the form of acceptable hypotheses. This was declarative to the extent that it specified the form of Prolog clauses, but it could have been written in a more declarative style that described pharmacophores directly. We see improvement in the form of the declarative bias as an area for further work with PROGOL in particular.

For machine learning systems that take explicit background knowledge (e.g. META-DENDRAL, AQ (Michalski et al., 1986), ILP systems), it is widely accepted that a substantial knowledge engineering effort often is required to encode this knowledge. But we believe that a substantial knowledge engineering effort is required for the application of many machine learning systems that do not take explicit background knowledge as input, such as decision tree learners. In the case of such algorithms, knowledge engineering issues still arise with code that users invariably write for precomputing various features. We believe that the value of a declarative representation applies to these types of machine learning algorithms as well, since developing the code that precomputes interesting domain-specific features may require a high degree of interaction with a domain expert.

Recommendation 4: We recommend further study into the relative merits of declarative vs. non-declarative background knowledge. This investigation should not be limited to ILP systems; we believe it could involve all manner of machine learning systems. Such an investigation requires a carefully designed definition of *declarative*—it appears to be much easier to label pieces of code as "declarative" or "non-declarative" than to formulate a definitive set of criteria distinguishing between the two. In addition, we recommend further investigation into general kinds of background knowledge that might be re-usable across domains and across different types of learning algorithms, thus easing the burden of the knowledge engineering effort. For example, we hypothesize that background knowledge for geometric concepts might be useful across a wide variety of domains. Even more specific background knowledge, such as chemical knowledge, might apply to a variety of problem areas.

4.2. Lessons for PROGOL *users and for ILP*

The first lesson from the present work for ILP is that 3D concepts can be learned if geometry is represented logically by pairwise distances between points of interest. This could be useful in a variety of other domains as well, for example, the machining of tools. It should be noted that as the geometric concepts grow more complex, additional detail may be necessary, for example, to capture orientation or to distinguish between mirror images.

The second lesson is that PROGOL, and probably other relational learning algorithms, can be used without modification to address the multiple instance problem (Section 2.3.3). The present work shows that the multiple instance problem can be addressed naturally within ILP, without the need to develop a new learning algorithm tailored to the problem and without the loss of information that comes from averaging over examples in a set. The general approach is based on an "instance" predicate. The following generic form for hypotheses for multiple instance problems can be used.

$$positive(X) \leftarrow instance(X, Y), pos\text{-}properties(Y)$$

A set X is asserted to be *positive* just if it has an instance Y such that Y has the properties required for a label of *positive*. For illustration, in the present work the variable X would stand for an example molecule under consideration, and the variable Y would stand for a conformation of that molecule. In some cases it may be possible to avoid the need for an explicit instance predicate, by including both of the variables X and Y as arguments in the literal(s) describing properties necessary for a *positive* label. This was done in the present work (Section 3.6).

Recommendation for ILP research: Both PROGOL's success with a 3D problem and its natural fit to the multiple instance problem depend on the first-order definite clause representation, rather than on the details of PROGOL itself. Therefore we expect that these properties would apply to a variety of other ILP systems as well. We recommend the application of ILP systems to other 3D problems and to other domains that exhibit the multiple instance problem.

4.3. Lessons for pharmacophore discovery

In addition to PROGOL's general suitability to 3D concepts and the multiple instance problem, and its output of declarative hypotheses, PROGOL has one additional feature that makes it particularly effective for pharmacophore discovery. This feature is PROGOL's focus on a "seed" positive example. The use of a seed example can eliminate many needless portions of a search space, particularly if the smallest or "simplest" example is chosen as the seed. For illustration, one type of atom that sometimes is of interest in a pharmacophore is a *hydrogen donor*, for our purposes defined simply as a hydrogen bonded to an atom other than carbon. But in the present work the seed example actually had no hydrogen donors at all. As a result all hypotheses involving hydrogen donors were automatically omitted from the search.[4] We expect that the use of a seed molecule is a general idea that can be incorporated into other approaches to pharmacophore discovery to reduce their time complexities as well.

Recommendations for pharmacophore discovery: We recommend investigation into whether "seed molecules" can be used to reduce the time complexities of other approaches to pharmacophore discovery. In addition, we recommend (and intend to pursue) application of the methodology detailed in Section 3 to other pharmacophore discovery problems, as well as an investigation of automatically-proposed experiments where multiple plausible pharmacophores are generated. We encourage other machine learning researchers to repeat our results with PROGOL, to apply their own algorithms to the problem of ACE inhibition, or to use the ideas in this paper to approach other pharmacophore discovery problems, many of which can be found in the biochemistry literature. To this end the contact author is pleased to provide all the data and background knowledge code in electronic form, on request.

4.4. Lessons regarding ACE inhibition

As is the case for many enzymes, it has not yet been possible to determine the binding site for ACE. Hence the question of the most likely pharmacophore for ACE inhibition remains open and of interest to researchers. The present work provides further confirmation for an earlier proposal by Mayer et al. (Mayer et al., 1987), using energy-minimized conformations (according to Tripos' Sybyl and Advanced Computation packages) and using ideal binding geometries for zinc. But the present work also proposes an alternative. Specifically, the alternative places the zinc site at a location more than one Ångstrom closer to the triangle of hydrogen acceptors than does the original pharmacophore. Furthermore, given the assumptions provided by the chemists, no other potential pharmacophore exists.

There is no *a priori* reason to believe one of the proposed pharmacophores over the other. Subsequent to the experiments described in Section 3, we have tested eight additional ACE inhibitors published more recently (Lombaert et al., 1996). These tests confirm our earlier results in that all eight new ACE inhibitors exhibit both pharmacophores. Unfortunately, the new test cases do not distinguish between the two pharmacophores. The reason it has been difficult to distinguish between the pharmacophores is that the binding geometries for chemical groups that can bind to zinc are symmetric. Every ACE inhibitor we have examined thus far has a chemical group which can bind to zinc either at a location oriented away from the triangle of hydrogen acceptors—and thus corresponding to the first pharmacophore—or oriented toward the hydrogen acceptors and corresponding to the second pharmacophore. It should be possible, though time-consuming, to carefully design and synthesize compounds with only one of the two pharmacophores, and then to test such compounds for ACE inhibition. If successful, such experimentation would provide basic knowledge about the ACE binding site; this knowledge could possibly lead to improved ACE inhibitors.

Recommendation for the study of ACE inhibition: We recommend the design and synthesis of molecules that can be tested for ACE inhibition in order to distinguish between the two competing pharmacophores described in the present paper. We hope to carry out such testing ourselves as a first step into investigating the requirements for the automatic proposal of experiments within pharmacophore discovery.

5. Conclusion

This paper has presented a case study of a machine-aided knowledge discovery process within the general area of drug design. Within drug design, the particular problem of *pharmacophore discovery* was isolated, and the ILP system PROGOL was applied to the problem of identifying potential pharmacophores for ACE inhibition. The domain of pharmacophore discovery presented a natural "next step" beyond previous applications of ILP to structure-activity prediction within drug design. The case study reported in this paper supports four general lessons for knowledge discovery, as well as more specific lessons for pharmacophore discovery, for ILP, and for ACE inhibition. The general lessons for knowledge discovery are as follows.

1. An initial rediscovery step is a useful tool when approaching a new application domain.
2. General ML heuristics may fail to match the details of an application domain, but it may be possible to successfully apply a heuristic-based algorithm in spite of the mismatch.
3. A complete search for all plausible hypotheses can provide useful information to a user, although experimentation may be required to choose between competing hypotheses.
4. A declarative knowledge representation facilitates the development and debugging of background knowledge in collaboration with a domain expert, as well as the communication of final results.

Acknowledgments

We thank the editors, anonymous referees, and an internal Pfizer review board for suggestions that greatly improved the paper. We thank the Pfizer review board and Alan Frisch for independently noting that more complex 3D concepts may require an encoding that goes beyond pairwise distances, in order to distinguish mirror images. We thank Donald Michie for his comments on an earlier draft, including his observation that the representation of 3D concepts used here might be applicable to a variety of other domains. Much of this work was carried out while Stephen Muggleton and David Page were at the Oxford University Computing Laboratory. This research was supported partly by the Esprit Basic Research Action ILP2 (project 20237), the EPSRC project 'Experiments with Distribution-Based Machine Learning', the SERC project 'Experimental Application and Development of Inductive Logic Programming' and a SERC Advanced Research Fellowship held by Stephen Muggleton. Stephen Muggleton also was supported by a Research Fellowship at Wolfson College, Oxford.

Notes

1. The term *ligand* is used for molecules which bind to the protein binding site. A ligand might be highly active against a target, but not a "drug," because of a lack of other required properties such as metabolic stability (it may be broken down within the body), safety (it may be toxic), or an ability to diffuse from the gut into the bloodstream (if a requirement is that the drug can be taken by mouth).
2. The experiments reported in this paper were run using the Prolog version of PROGOL, P-PROGOL 2.3. This and a C version of PROGOL are available by anonymous ftp to the site ftp.comlab.ox.ac.uk, in the directory pub/Packages/ILP.
3. Two molecules are stereoisomers if they have the same atom and bond structure but different 3D arrangements, and it is not possible to convert from one to the other without breaking a bond.
4. We simply chose the first example supplied by the chemist as the seed example, and it was fortuitous that this happened to be one of the smaller molecules and to have no hydrogen donors. But a program could be written to examine a data set initially and select a seed example that is likely to yield a reduced search.

References

Andrews, P., Carson, J., Caselli, A., Spark, M., and Woods, R. (1985). Conformational analysis and active site modelling of angiotensin-converting enzyme inhibitors. *Journal of Medicinal Chemistry*, 28:393–399.

Bohacek, R., Lombaert, S. D., McMartin, C., Priestle, J., and Grutter, M. (1996). Three-dimensional models of ACE and NEP inhibitors and their use in the design of potent dual ACE/NEP inhibitors. *Journal of the American Chemical Society*, 118:8231–8249.

Brint, A. and Willett, P. (1987). Algorithms for the identification of three-dimensional maximal common substructures. *J. Chem. Inf. Comput. Sci.*, 27(152):152–158.

Buchanan, B., Feigenbaum, E., and Sridharan, N. (1972). Heuristic theory formation: data interpretation and rule formation. In Meltzer, B. and Michie, D., editors, *Machine intelligence 7*, pages 267–290. Edinburgh University Press.

Debnath, A., de Compadre, R. L., Debnath, G., Schusterman, A., and Hansch, C. (1991). Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of Medicinal Chemistry*, 34(2):786 – 797.

Dietterich, T., Lathrop, R., and Lozano-Perez, T. (1997). Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (1996). *Advances in knowledge discovery and data mining*. AAAI Press / MIT Press.

Finn, P. (1996). Computer-based screening of compound databases for the indentification of novel leads. *Drug Discovery Today*, 1:363–370.

Hansch, C. and Leo, A. (1995). Exploring QSAR. *ACS*.

Hansch, C., Maloney, P., Fujita, T., and Muir, M. (1962). Correlation of biological activity of phenoxyacetic acids with Hammett substituent constants and partition coefficients. *Nature*, 194:178–180.

Hassell, C., Krohn, A., Moody, C., and Thomas, W. (1982). The design of a new group of angiotensin-converting enzyme inhibitors. *FEBS Letters*, 147:175–179.

Jain, A., Dietterich, T., Lathrop, R., Chapman, D., Critchlow, R., Bauer, B., Webster, T., and Lozano-Pérez, T. (1994a). Compass: a shape-based machine learning tool for drug design. *Journal of Computer-Aided Molecular Design*, 8:635–652.

Jain, A., Koile, K., Bauer, B., and Chapman, D. (1994b). Compass: Predicting biological activities from molecular surface properties. *Journal of Medicinal Chemistry*, 37:2315–2327.

King, R., Muggleton, S., Lewis, R., and Sternberg, M. (1992). Drug design by machine learning: The use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. *Proceedings of the National Academy of Sciences*, 89(23):11322–11326.

King, R., Muggleton, S., Srinivasan, A., and Sternberg, M. (1996). Structure-activity relationships derived by machine learning: the use of atoms and their bond connectives to predict mutagenicity by inductive logic programming. *Proceedings of the National Academy of Sciences*, 93:438–442.

Leach, A. (1991). A survey of methods for searching conformational space of small and medium sized molecules. In Lipkowitz and Boyd, editors, *Reviews of Computational Chemistry, Vol. 2*. VCH USA.

Lee, Y., Buchanan, B., and Aronis, J. (1998). Knowledge-based learning in exploratory science: Learning rules to predict rodent carcinogenicity. *Machine Learning*, vv:xxx–xxx.

Lombaert, S. D., Chatelain, R., Fink, C., and Trapani, A. (1996). Design and pharmacology of dual angiotensin-converting enzyme and neutral endopeptidase inhibitors. *Current Pharmaceutical Design*, 2:443–462.

Martin, Y., Bures, M., Danaher, E., DeLazzer, J., Lico, I., and Pavlik, P. (1993). A fast new approach to pharmacophore mapping and its application to dopaminergic and benzodiazepine agonists. *Journal of Computer-Aided Molecular Design*, 7:83–102.

Mayer, D., Naylor, C., Motoc, I., and Marshall, G. (1987). A unique geometry of the active site of angiotensin-converting enzyme consistent with structure-activity studies. *Journal of Computer-Aided Molecular Design*, 1:3–16.

Michalski, R., Mozetic, I., Hong, J., and Lavrac, N. (1986). The AQ15 inductive learning system: an overview and experiments. In *Proceedings of IMAL 1986*, Orsay. Université de Paris-Sud.

Muggleton, S. (1995). Inverse entailment and Progol. *New Generation Computing*, 13:245–286.

Muggleton, S. (1996). Learning from positive data. In *Proceedings of the Sixth Inductive Logic Programming Workshop*, Lecture notes in artificial intelligence, Berlin. Springer-Verlag.

Muggleton, S. and Feng, C. (1990). Efficient induction of logic programs. In *Proceedings of the First Conference on Algorithmic Learning Theory*, Tokyo. Ohmsha.

Muggleton, S., Page, C., and Srinivasan, A. (1996). An initial experiment into stereochemistry-based drug design using ILP. In *Proceedings of the Sixth Inductive Logic Programming Workshop*, Lecture notes in artificial intelligence, Berlin. Springer-Verlag.

Nilsson, N. (1980). *Principles of Artificial Intelligence*. Tioga, Palo Alto, CA.

Provost, F. and Aronis, J. (1996). Scaling up inductive learning with massive parallelism. *Machine Learning*, 23:33–46.

Saith, R., Srinivasan, A., Michie, D., and Sargent, I. (1997). The relationship between embryo, oocyte and follicular features and the developmental potential of human IVF embryos. *Human Reproduction (Submitted)*.

Shapiro, E. (1983). *Algorithmic program debugging*. MIT Press.

Srinivasan, A. and Camacho, R. (1996). Experiments in numerical reasoning with ILP. Technical Report PRG-TR-22-96, Oxford University Computing Laboratory, Oxford.

Srinivasan, A. and Camacho, R. (1997). Experiments in numerical reasoning with ILP. *Journal of Logic Programming (accepted)*.

Whittle, P. and Blundell, T. (1994). Protein structure-based drug design. *Annu. Rev. Biophys. Biomol. Struct.*, 23:349–375.