Mutagenesis: ILP experiments in a non-determinate biological domain

A. Srinivasan and S.H. Muggleton

Oxford University Computing Laboratory,
Wolfson Building, Parks Road, Oxford.

R.D. King and M.J.E. Sternberg

Biomolecular Modelling Laboratory,
Imperial Cancer Research Fund,
Lincoln's Inn Fields, London.

Abstract

This paper describes the use of Inductive Logic Programming as a scientific assistant. In particular, it details the application of the ILP system *Progol* to discovering structural features that can result in mutagenicity in small molecules. To discover these concepts, *Progol* only had access to the atomic and bond structure of the molecules. With such a primitive description and no further assistance from chemists, *Progol* corroborated some existing knowledge and proposed a new structural alert for mutagenicity in compounds. In the process, the experiments act as a case study in which, even with extremely limited background knowledge, an Inductive Logic Programming tool firstly, complements a complex statistical model developed by skilled chemists, and secondly, continues to provide understandable theories when the statistical model fails. The experiments also constitute the first demonstrations of a prototype of the *Progol* system. *Progol* allows the construction of hypotheses with bounded non-determinacy by performing a best-first search within the subsumption lattice. The results here provide evidence that such searches are both viable and desirable.

1 Introduction

There is more to the business of scientific theory formation than just data-fitting. To be acceptable, a theory must also be understandable and open to critical analysis. This in turn, places certain requirements on programs that aim to actively assist in the theory construction process, namely, that their outputs be in terms of explicit concepts that can be easily evaluated. For some time, the field of Inductive Logic Programming (ILP) has been making steady progress towards providing automated assistance in the process of scientific discovery. In contrast to other 'discovery' systems like BACON [4] and AM [6], ILP systems have discovered new knowledge that has been refereed and published in journals of the relevant subject area. This paper mainly seeks to

investigate whether ILP algorithms still discover useful concepts when expert background knowledge is sparse or even unknown. In particular, using the ILP algorithm *Progol*, the experiments reported here are aimed at providing support to the following conjectures:

Conjecture 1. Even with poor background knowledge, *Progol* can discover concepts that are understandable and can usefully assist statistical models devised by experts in a field.

Conjecture 2. In cases where statistical models fail, *Progol* alone can still work effectively as a theory constructor.

All experiments deal with the problem of discovering rules for mutagenicity in nitroaromatic compounds. These compounds occur in automobile exhaust fumes and are also common intermediates in the synthesis of many thousands of industrial compounds [1]. Highly mutagenic nitroaromatics have been found to be carcinogenic, and often cause damage to DNA. For the experiments in this paper, the main interest lies in discovering the relationship between mutagenicity and molecular structure. In particular, we explore the possibility of determining such relationships using a much richer representation of molecules than other studies [3]. Each molecule is described in terms of atoms and bonds, with the addition of atomic and bonding properties output by a standard molecular graphics package. Since this is a highly non-determinate representation, propositional feature-based algorithms cannot be directly applied. The problem is also not accessible by ILP systems which incorporate the ij-determinate restriction, such as *Golem* [8] and LINUS [5]. A further difference from studies such as [3] is that background knowledge is here largely dispensed with.

This paper is organised as follows. Section 2 describes the mutagenicity problem, and the representation of the compounds studied. The experiments with *Progol* are in Sections 3 and 4. Complete details of the ILP system *Progol* are available in [7]. However, as the experiments here constitute the first demonstrations of this system, we include relevant details in an Appendix.

2 The Mutagenesis Problem and its representation

The problem at hand concerns identifying mutagenic compounds using only the atomic and bond structure of the compounds. Mutagenic compounds are often known to be carcinogenic and also cause damage to DNA. Clearly, it is of considerable interest to the pharmaceutical industry to determine which molecular features result in compounds having mutagenic activity. Besides directing the development of less hazardous new compounds, it also has applicability in areas such as antimicrobial agents where it is not possible to determine mutagenicity using standard tests (this is because of the toxicity of the agents to test organisms). As stated in Section 1, our intentions here are however, more general. Firstly, a capacity to cope with such a low-level, highly non-determinate representation will open up a range of arbitrary chemical structures for analysis. Second, we aim to explore the effectiveness of ILP programs when there is very little assistance from external sources.

To this end, we have chosen to study the mutagenicity of 230 compounds listed in [1]. The authors of [1] propose a linear regression model to predict mutagenicity. They use the following independent variables:

$logP$: log of the compound's octanol/water partition coefficient (hydrophobicity);

ϵ_{LUMO}: energy of the compounds lowest unoccupied molecular orbital. This is obtained from a quantum mechanical molecular model;

I_1: an 'indicator variable' that is set to 1 for all compounds containing 3 or more fused rings; and

I_a: an 'indicator variable' that takes the value 1 for "...five examples of acenthrylenes and shows that these are much less active than expected for some unknown reason" ([1], pp 788).

With these 4 attributes, the authors of [1] identify 188 compounds as being amenable to a regression analysis. The remaining 42 outlier compounds were not used in constructing the regression model. The resulting equation for the 'regression friendly' subset is shown below:

$$\begin{aligned} logM = \; & 0.65(\pm 0.16)logP - 2.90(\pm 0.59)log(\beta 10^{logP} + 1) - 1.38(\pm 0.25)\epsilon_{LUMO} \\ & + 1.88(\pm 0.39)I_1 - 2.89(\pm 0.81)I_a - 4.15(\pm 0.58) \end{aligned} \quad (1)$$

where $logM$ = log mutagenicity and $log\beta = -5.48$. We confine this study to the simple problem of discriminating compounds with positive log mutagenicity from those which have zero or negative log mutagenicity.

2.1 Bond-level representation of molecules

A prominent study involving the analysis of drug structures with ILP was first reported by [3]. While it highlighted the advantages of logic-based learning, all drugs studied were variants of a basic template, and all that was required was substitutions into 3 positions on that template (see Figure 1). This is reflected by the fact that the rules obtained in that study were largely propositional.

In contrast, the compounds in this study are considerably more diverse (see Figure 2). A wider applicability to more arbitrary chemical structures requires the capacity to reason at a much lower level. The most primitive structural representation of molecules that is practical is in terms of the atomic and bonding properties of the molecule. At this level, feature-based algorithms are inapplicable, as it is usually impossible to know all relevant sub-structures for all molecules. It appears that the task of identifying such structural relationships will provide a true challenge to ILP algorithms. In fact, it is clear that the problem will not even be accessible by ILP algorithms that incorporate any determinacy restriction: a single molecule usually has several atoms, each of which can be associated in more than one bond.

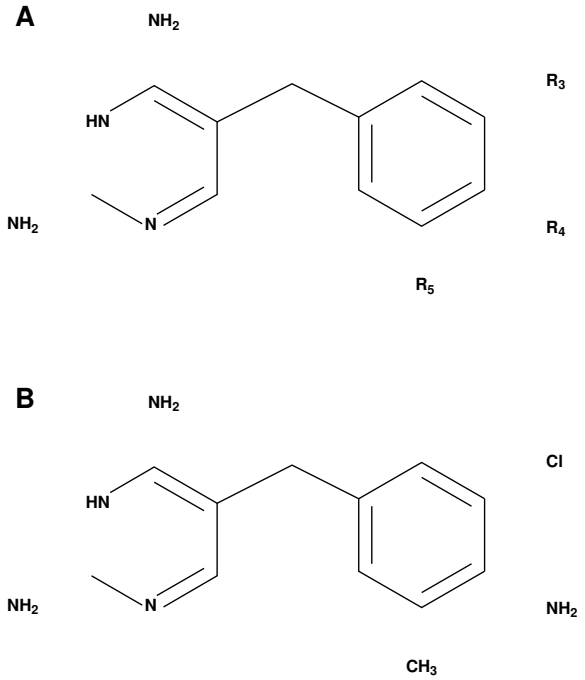

Figure 1: Typical data format for compounds used in [3]. A) Template of 2,4-diamino-5(substituted-benzyl)pyrimidines R3, R4, and R5 are the three possible substitution positions. B) Example compound: $3-Cl$, $4-NH_2$, $5-CH_3$

2.2 Data and hypothesis language for *Progol*

Bond-level data. The atom and bond structures of the 230 drugs were obtained from the standard molecular modelling package QUANTA. For each compound QUANTA automatically obtains the atoms, bonds, bond types (for example, aromatic, single, double etc.), atom types (for example, aromatic carbon, aryl carbon etc.), and the partial charges on atoms. QUANTA automatically classifies bonds into one of 8 types, and atoms into one of 233 types (most of which relate to different types of carbon atoms). The output was a set of Prolog facts of the form:

bond(compound,atom1,atom2,bondtype), stating that *compound* has a bond of *bondtype* between the atoms *atom1* and *atom2*. For example, an aromatic bond between atoms *d2_1* and *d2_2* in drug *d2* is represented by QUANTA as *bond(d2,d2_1,d2_2,7).*

atm(compound,atom,element,atomtype,charge), stating that in *compound atom* has element *element* of *atomtype* and partial charge *charge*. For example, QUANTA encodes the fact that atom *d2_1* in drug *d2* is an aromatic carbon atom with partial charge 0.067 by the fact *atm(d2,d2_1,c,22,0.067).*

The resulting 12203 facts on atomic structure and bonding generated by QUANTA form the only knowledge available for learning.

Positive and negative examples. Of the 230 compounds, 138 have positive levels of log mutagenicity (as reported in [1]). These are labelled 'active'

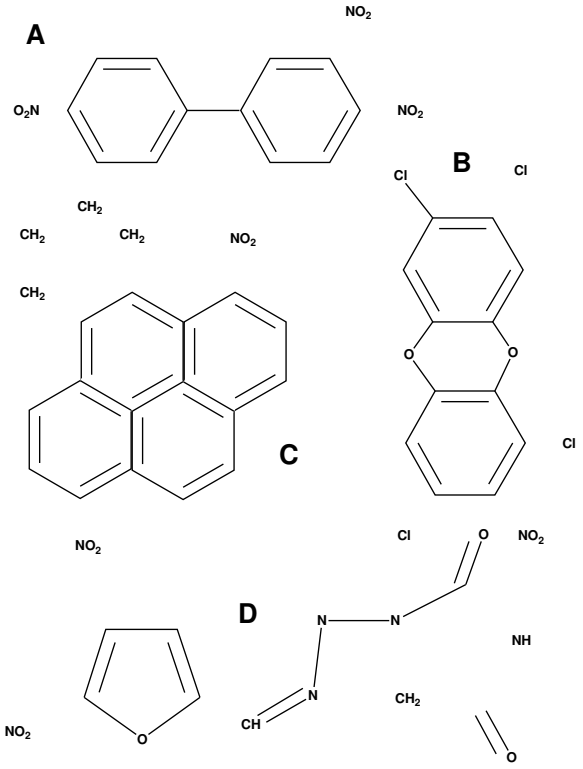

Figure 2: Examples of the diverse set of aromatic and heteroaromatic nitro compounds used in the mutagenesis study. A) 3,4,4'-trinitrobiphenyl B) 2-nitro-1,3,7,8-tetrachlorodibenzo-1,4-dioxin C) 1,6,-dinitro-9,10,11,12-tetrahydrobenzo[e]pyrene D) nitrofurantoin

and constitute the source of positive examples. The remaining 92 are labelled 'inactive' and constitute the source of negative examples. Figure 3 shows the distribution of compounds into these classes for the subsets studied here.

Compounds	'Active'	'Inactive'	Total
'Regression friendly'	125	63	188
'Regression unfriendly'	13	29	42

Figure 3: Class distribution of compounds

The hypothesis language $\mathcal{L}$. The hypothesis language $\mathcal{L}$ for *Progol* is defined in Figure 4 (see Section A.2 for more details).

Mode declarations	mode(*,bond(+compound,−atomid,−atomid,#integer)) mode(*,bond(+compound,−atomid,+atomid,#integer)) mode(*,bond(+compound,+atomid,−atomid,#integer)) mode(*,bond(+compound,+atomid,+atomid,#integer)) mode(*,atm(*,+compound,+atomid,#element,#integer,-charge)) mode(*,atm(*,+compound,−atomid,#element,#integer,-charge)) mode(1,(+charge)=(#charge))
Depth of variables	2
Maximum negatives	5
Maximum literals	4

Figure 4: Language specification for *Progol*

3 Experiment 1: *Progol* on 'regression friendly' data

Aim. To evaluate the performance of *Progol* on the 188 compounds designated by medicinal chemists as being well-suited to a regression analysis.

Materials. The bond-level representation of the 188 compounds generated by QUANTA, the log mutagenic activity of the compounds, a Prolog implementation of *Progol*, Regression Equation 1 derived in [1], and a SparcStation 10.

Method.

1. Compounds recorded as having positive log mutagenicity are labelled 'active' and form the positive examples for *Progol*. The remaining compounds are taken to be the negative examples.

2. The bond-level Prolog facts, positive and negative examples are provided to *Progol*, which returns a set of clauses explaining the examples.

3. The predictions from Equation 1 on the 188 compounds are calculated. The class predicted by the equation is taken to be 'active' if the log mutagenicity predicted is greater than 0.

4. The performances of the two approaches are evaluated for significant differences. The need for the special 'indicator variables' used in the regression analysis is contrasted against the ILP approach.

Experimental results.
Figure 5 shows the theory returned by the *Progol* algorithm. Performances of *Progol* and Equation 1 on the data are shown in Figure 6. Expected values under the hypothesis of no association between predicted and actual outcome values are shown in parentheses.

Discussion.
The first question to be addressed is whether either *Progol* or the regression model has acquired any predictive power above the level of random guessing. This is adequately answered by examining the expected values in Figure 6.

```
Rule1: active(A) :-
        atm(A,B,c,195,C).
Accuracy = 100%, Coverage = 10%. In QUANTA's representation a carbon atom of Type
195 occurs in a third adjacent pair of fused six-membered rings. Thus, all
compounds with 3 fused rings will be labelled active by this rule.

Rule2: active(A) :-
        atm(A,B,c,10,C), atm(A,D,c,22,E), bond(A,D,B,1).
Accuracy = 84%, Coverage = 30%. In QUANTA's representation a carbon atom of Type
10 is aliphatic, and of Type 22 occurs in a six-membered aromatic ring. Thus, this
rule identifies compounds with a aliphatic carbon atom attached by a single
bond to a ring carbon.

Rule3: active(A) :-
        atm(A,B,c,27,C), bond(A,D,E,1), bond(A,B,E,7).
Accuracy = 90%, Coverage = 58%. In QUANTA's representation a carbon atom of Type
27 merges two six-membered aromatic rings. A bond of Type 7 is an aromatic bond. Thus,
this rule identifies compounds of two fused six-membered aromatic rings, one of which
has a further single bond with an atom of any type (that is, not necessarily a
carbon atom).

Rule4: active(A) :-
        atm(A,B,o,40,C), atm(A,D,n,32,C).
Accuracy = 71%, Coverage = 8%. In QUANTA's representation a nitrogen atom of Type
32 occurs in an amide group, and an oxygen atom of Type 40 occurs in a nitro group.
This rule labels as active all compounds that have a pair of such atoms with the same
partial charge.

Rule5: active(A) :-
        atm(A,B,o,40,-0.383).
Accuracy = 82%, Coverage = 7%. A compound is active if it has an oxygen in a nitro
group with partial charge -0.383.

Rule6: active(A) :-
        atm(A,B,o,40,-0.384).
Accuracy = 89%, Coverage = 13%. A compound is active if it has an oxygen in a nitro
group with partial charge -0.384.

Rule7: active(A) :-
        atm(A,B,o,40,-0.378).
Accuracy = 100%, Coverage = 5%. A compound is active if it has an oxygen in a nitro
group with partial charge -0.378.

Rule8: active(A) :-
        atm(A,B,h,3,0.149).
Accuracy = 88%, Coverage = 6%. In QUANTA's representation a hydrogen atom of Type 3
is aromatic. Thus, a compound with an aromatic hydrogen with partial charge of 0.149
is labelled active.

Rule9: active(A) :-
        atm(A,B,h,3,0.144).
Accuracy = 89%, Coverage = 6%. A compound with an aromatic hydrogen with partial charge
of 0.144 is labelled active.
```

Figure 5: Progol's theory for 'regression friendly' mutagenic compounds. Accuracy is 100(Positive examples covered by clause)/(Total examples covered by the clause). Coverage is 100(Positive examples covered by clause)/(Total number of positive examples)

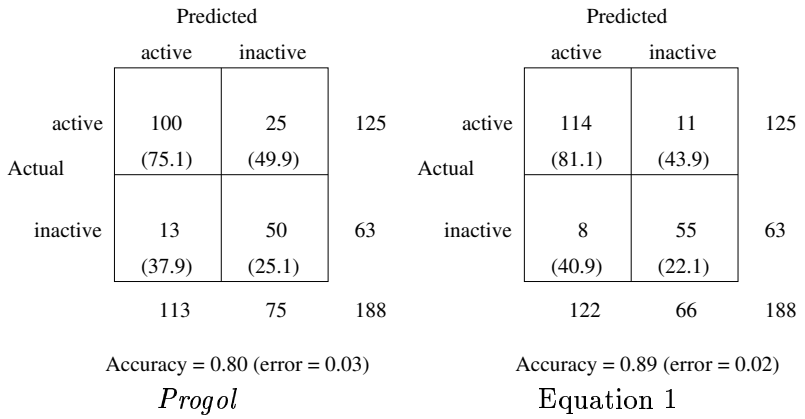

Figure 6: Performance tables for *Progol* and regression on 'regression friendly' compounds

The null hypothesis is that there is no association between the predicted and actual outcome values. With this hypothesis, the large difference in observed and expected values show that it is safe to reject the null hypothesis.

A quantitative comparison of the two approaches can exploit the fact that both algorithms were tested on the same sample. The appropriate statistical test for this is the McNemar's test for changes [2]. The null hypothesis is that the proportions of examples correctly classified by both algorithms is the same. [1] Figure 7 cross-tabulates the performances of *Progol* and Equation 1. If there is no significant difference in the performance of the two algorithms, half of the 45 cases whose classifications disagree should be classified correctly by *Progol* and the other half should be classified correctly by Equation 1. Figure 7 shows the observed number of cases that fell into these categories to be 13 and 32 respectively. To compare these observed and expected frequencies, we use the χ^2 test statistic to compare the observed frequencies against the expected frequency of $45/2 = 22.5$. Including the Yates correction for continuity, the χ^2 value of 7.2 shows that there is a significant difference in the proportions of correct predictions of the two algorithms at $P < 0.01$.

The comparison in Figure 7 can be misleading on two counts. Firstly, the estimates of accuracy are resubstitution estimates which are usually optimistic. Second, Equation 1 has access to specialist knowledge knowledge that is beyond the basic structural information produced by QUANTA. This knowledge is used by the authors of [1] to identify specific structural and chemical attributes that could be helpful to identify mutagenic compounds. In this experiment, *Progol* has been denied access to such knowledge.

To get a better idea of the predictive accuracy of *Progol* and a linear regression model on these compounds, the experiment was repeated by leaving out a randomly chosen subset of 56 compounds (constituting approximately 30% of the data). A theory for the remainder was acquired using *Progol* and a standard linear regression technique. To approximate the procedure adopted

[1] By 'correctly classified' we mean that compounds with positive log mutagenicity in the test set are classified as active, and those with zero or negative values are classified as inactive.

		Regression: correct	Regression: incorrect	
Progol	correct	137	13	150
	incorrect	32	6	38
		169	19	188

Figure 7: Compounds correctly and incorrectly classified by *Progol* and regression

by the authors of [1] the regression package had access to the same 4 attributes described in Section 2. Figure 8 tabulates the performance of the two methods on the 56 compounds set aside. Applying McNemar's test on Figure 9 and including the Yates correction for continuity, the χ^2 value is now 4.9 which is significant at $P < 0.05$. Although the significance levels were over-estimated on the training set ($P < 0.01$, see Figure 7 and associated calculations), it is clear that the regression model still performs better.

		Predicted: active	Predicted: inactive	
Actual	active	35 (32.9)	6 (8.1)	41
	inactive	10 (12.1)	5 (2.9)	15
		45	11	56

Accuracy = 0.71 (error = 0.06)

Progol

		Predicted: active	Predicted: inactive	
Actual	active	39 (32.9)	2 (8.1)	41
	inactive	6 (12.1)	9 (2.9)	15
		45	11	56

Accuracy = 0.86 (error = 0.05)

Equation 1

Figure 8: Performance tables for *Progol* and regression on a randomly chosen test set

The strengths of the *Progol* theory lie in identifying structural regularities in the data. This is highlighted by examining the the use of special (hand-crafted) structural variables for the regression analysis. Figure 10 shows the structural properties described by the first three rules in the *Progol* theory. An interesting feature is that the regression analysis utilised a special structural variable to flag the existence of three or more fused rings (the variable I_1 in Equation 1). As seen in Figure 10, *Progol* finds this constraint, albeit in a circuitous fashion, by exploiting QUANTA's representation of carbon atoms in 3 adjacent fused rings (see Rule 1 in Figure 5). The regression analysis utilised a further structural variable to flag 5 special compounds (the variable I_a) which the authors describe as "... The very low activity of the acenthrylenes

		Regression correct	Regression incorrect	
Progol	correct	39	1	40
	incorrect	9	7	16
		48	8	56

Figure 9: Compounds correctly and incorrectly classified by *Progol* and regression on the test set

is surprising in that most of the other large polycyclic aromatic compounds are reasonably well fit. This deviant group cries out for further investigation." ([1], pp 793). Figure 11 tabulates *Progol*'s classification of this group of compounds. Although the particular rules are rather shallow (especially Rules 5,7 and 9), it is still satisfying to see that an ILP algorithm can adequately classify the compounds without any special consideration.

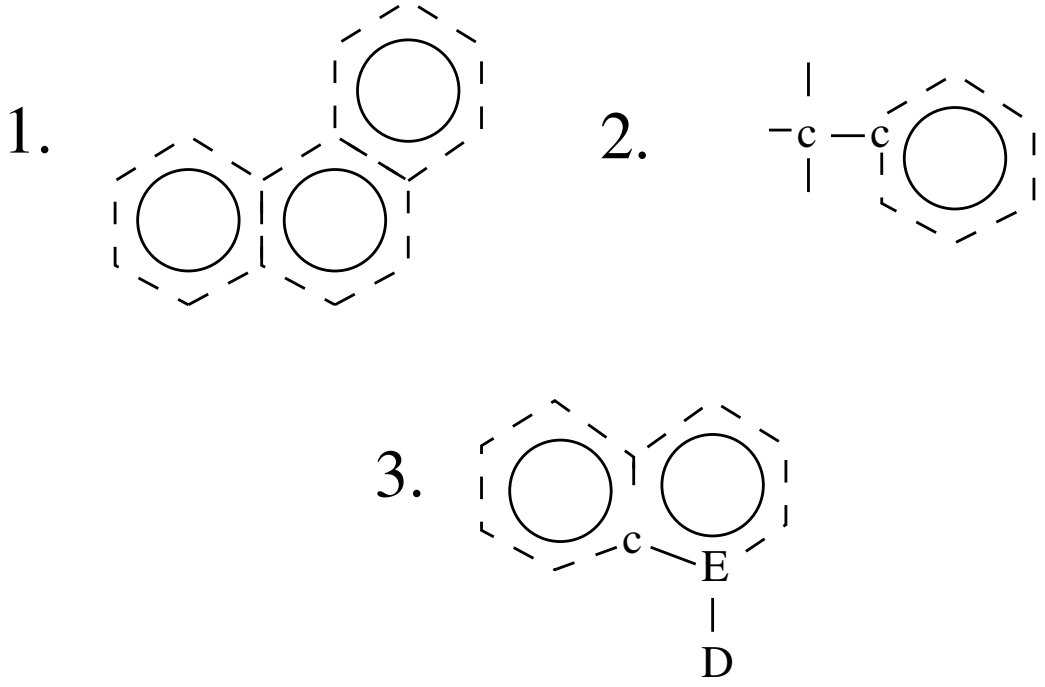

Figure 10: Properties discovered by *Progol* in 'regression friendly' data set

Compound	log Mutagenicity	*Progol* classification (rules used in parentheses)
2-nitrobenz[j]acenthrylene	0.86	active (1,9)
4-nitrobenz[k]acenthrylene	0.67	active (9)
2-nitrobenz[l]acenthrylene	0.26	active (9)
6-nitrobenz[e]acenthrylene	0.04	active (5)
5-nitrobenz[k]acenthrylene	0.92	active (7)

Figure 11: Progol's classification of deviant group of acenthrylene compounds.

4 Experiment 2: *Progol* on 'regression unfriendly' data

Aim. To evaluate the performance of *Progol* on the 42 compounds designated by medicinal chemists as being not amenable to a regression analysis.

Materials. The bond-level representation of the 42 compounds generated by QUANTA, the log mutagenic activity of the compounds, a Prolog implementation of *Progol*, and a SparcStation 10.

Method.

1. The classification of 42 compounds into positive and negative examples are as in Experiment 1. Along with the bond-level facts, these are provided in turn to *Progol* as before.

2. Given the relatively small size of the data-set, the predictive accuracy of the *Progol* theory is assessed by adopting a leave-one-out procedure.

Experimental results.
Figure 12 shows the theory returned by the *Progol* algorithm.

```
active(A) :-
        bond(A,B,C,2), bond(A,D,B,1), atm(A,D,c,21,E).
Accuracy = 100%, Coverage = 62%. In QUANTA's representation
a carbon atom of Type 21 is a member of a five-membered aromatic
ring. The rule identifies compounds with a double bond
conjugated to a five-membered aromatic ring.
```

Figure 12: Progol's theory for 'regression unfriendly' mutagenic compounds. Accuracy is 100(Positive examples covered by clause)/(Total examples covered by the clause). Coverage is 100(Positive examples covered by clause)/(Total number of positive examples)

The result of a leave-one-out analysis is shown in Figure 13. As before, expected values are in parentheses.

		Predicted		
		active	inactive	
Actual	active	8	5	13
		(2.5)	(10.5)	
	inactive	0	29	59
		(5.5)	(23.5)	
		8	34	42

Accuracy = 0.88 (error = 0.05)

Figure 13: Table from leave-one-out analysis of 'regression unfriendly' compounds using *Progol*

Discussion.
Clearly, the distribution of class values is highly skewed (there are more than twice as many 'inactive' compounds as 'active'). To check whether *Progol* has acquired any predictive power above the level of random guessing, the null hypothesis is that there is no association between the predicted and actual outcome values. Including the Yates correction for continuity, the χ^2 value of 18, shows that it is safe to reject the null hypothesis. Figure 14 shows some example compounds in this subset, and the structural property expressed by the *Progol* rule. No structural alerts have previously been proposed for mutagenesis in the 42 compounds. The predictive power of the simple relational property as indicated by the accuracy estimated from the leave-one-out procedure provides a good indication of the potential of an ILP algorithm to construct theories in situations where there is little knowledge of useful attributes that can be used in a standard statistical analysis. We have verfied that the the attributes used for the regression friendly subset are of little value here. With these attributes, the predictive power of both linear regression and linear discrimination is no better than a default rule that classifies all compounds in the set of 42 as 'inactive'. As for *Progol*, the predictive accuracy for these two techniques is estimated from a leave-one-out procedure.

As seen in Figure 12, the rule only explains about 60% of the data (8 out of the 13 compounds pre-classified as 'active'). *Progol* did find other rules for the remaining compounds. These were, however, discarded as being non-compressive. More data is required to confirm these patterns.

5 Discussion

It is important that theoretical developments in Inductive Logic Programming are validated by practical application. The experiments reported here are aimed at this. We believe that they go some way towards establishing that programs like *Progol* can indeed actively aid the theory construction process. By providing little or no background knowledge, we have under-utilised the actual power offered by the ILP framework. That the results are favourable even in this sit-

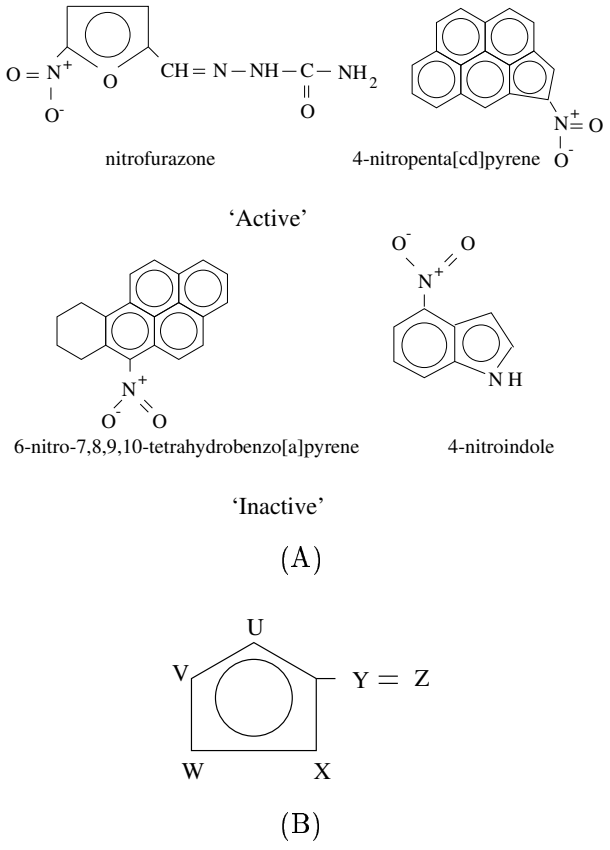

Figure 14: (A) Some 'regression unfriendly' compounds and (B) the structural property found by *Progol*

uation is a good sign for ILP algorithms. Turning to the actual experiments, a next step could be to provide the attributes used in the regression analysis (in particular, $logP$ and ϵ_{LUMO}) as background knowledge to *Progol*. Along with predicates for numerical reasoning, we would expect that the resulting theory to at least match the regression equation in predictive power. The significant edge of understandability of an ILP solution would of course, remain.

Although the solutions found by *Progol* appear particularly simple, the mutagenesis problem is not a trivial one. The representation is obviously relational, highly non-determinate, and involves a large database of facts. That *Progol* is able to perform well is largely due to its strategy of generalisation, namely, using a ground-plan (provided by the most-specific clause) to direct the search for acceptable clauses. Other experiments that we have conducted provide evidence that complete search carried out within this setting runs faster than greedy heuristic searches and returns more compact theories [9]. To this extent, we expect this domain will act as an important test-bed for ILP algorithms.

Acknowledgements

This paper has been helped greatly by discussions with Donald Michie. This research was supported partly by the Esprit Basic Research Action ILP (project

6020), the SERC project project 'Experimental Application and Development of ILP' and an SERC Advanced Research Fellowship held by Stephen Muggleton. Stephen Muggleton is a Research Fellow of Wolfson College Oxford. Thanks are also due to Rui Camacho and David Page for interesting discussions on *Progol*.

References

[1] A.K. Debnath, R.L Lopez de Compadre, G. Debnath, A.J. Schusterman, and C. Hansch. 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, 1991.

[2] B.S. Everitt. *The analysis of contingency tables.* Chapman and Hall, London, second edition, 1992.

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

[4] P. Langley, G.L Bradshaw, and H. Simon. Rediscovering chemistry with the Bacon system. In R. Michalski, J. Carbonnel, and T. Mitchell, editors, *Machine Learning: An Artificial Intelligence Approach*, pages 307–330. Tioga, Palo Alto, CA, 1983.

[5] N. Lavrac and S. Dzeroski. *ILP: Techniques and Applications.* Ellis Horwood, London, 1994.

[6] D.B. Lenat. On automated scientific theory formation: a case study using the AM program. In J.E. Hayes and D. Michie, editors, *Machine Intelligence 9.* Horwood, New York, 1981.

[7] S. Muggleton and A. Srinivasan. Mode-directed inverse resolution. In D. Michie K. Furukawa and S. Muggleton, editors, *Machine Intelligence 14.* Oxford University Press, 1994. (to appear).

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

[9] A. Srinivasan, S.H. Muggleton, R.D. King, and M.J.E. Sternberg. Discovering rules for mutagenesis. Technical Report PRG-TR-4-94, Oxford University Computing Laboratory, Oxford, 1994.

A *Progol*

A.1 Specification for *Progol*

The formal specification for *Progol* is as follows.

- $B = C_1 \wedge C_2 \wedge \ldots$ is background knowledge consisting of a set of definite clauses.
- $E = E^+ \wedge E^-$ is a set of examples where
  - $E^+ = e_1 \wedge e_2 \wedge \ldots$ are definite clauses and
  - $E^- = \overline{f_1} \wedge \overline{f_2} \wedge \ldots$ are non-definite horn clauses.
- $H = D_1 \wedge D_2 \wedge \ldots$ is an hypothesised explanation of the examples in terms of the background knowledge.

Progol can be treated as an algorithm A such that $H = A(B, E)$ is a minimal complexity hypothesis from predefined language $\mathcal{L}$ for which each D_i in H has the property $B \wedge D_i \models e_1 \vee e_2 \vee \ldots$, $\{e_1, e_2, \ldots\} \subseteq E^+$ and $B \wedge H \wedge E^- \not\models \square$. If more than one such minimal H exists then *Progol* returns the first one in an arbitrary lexicographic ordering $\mathcal{O}(\mathcal{L})$.

A.2 Defining a hypothesis language for *Progol*

The language $\mathcal{L}$ is defined in terms of

- Mode declarations which state the 'forms' that atoms in hypothesis can take in terms of
  - the places where variables are allowed an whether they are inputs or outputs (indicated by $+$ or $-$);
  - the places where constants are allowed (indicated by $\#$);
  - the types of these variables and constants; and
  - the degree of indeterminacy when making such a call to the background knowledge. This is either a number or $*$ meaning finite but unbounded recall of the goal.
- the maximum number of layers of variables introduced by atoms in the body of the clause from variables in the head of the clause;
- the acceptable level of consistency in terms of the maximum number of negatives that can be covered by any clause; and
- the maximum cardinality of any clause.

A.3 Algorithmic description of *Progol*

The top-level algorithm for *Progol* is Algorithm BestClauseSet. Given an example e, the algorithm for finding the best clause is Algorithm BestClause.

$\rho(C)$ is a set of clauses $\{D_1, .., D_m\}$ such that for each i, $1 \leq i \leq m$ the clause C subsumes D_i and D_i subsumes $\bot(B, e)$. The closure ρ^* contains every clause $C \in \mathcal{L}$ for which C subsumes $\bot(B, e)$. The extent of the completeness guarantee for *Progol* only extends to this property of ρ, and thus a *set* of clauses returned by *Progol* is guaranteed to be optimal only for single-clause target concepts.

Algorithm: $BestClauseSet(B, \langle E^+, E^- \rangle)$
Let Seeds $= E^+$
Let $H = \{\}$
While Seeds $\neq \{\}$
    Remove an arbitrary example e from Seeds
    Let $C = BestClause(B, e, \langle E^+, E^- \rangle)$
    Let $H = H \cup \{C\}$
Return H

Algorithm: $BestClause(B, e, \langle E^+, E^- \rangle)$
$\perp(B, e)$ is the most specific clause in $\mathcal{L}$ such that $B, \perp(B, e) \models e$
Let $p(C) = |\{e : e \in E^+ \text{ and } B, C \vdash e\}|$
Let $n(C) = |\{e : \overline{e} \in E^- \text{ and } B, C \vdash e\}|$
Let $c(C) = |C|$, the number of literals in C
Let $h(C) =$ the minimum number of additional literals
        needed complete the input/relations in C
Let $f(C) = p(C) - (c(C) + n(C) + h(C))$
Let $C = \{\}$, $Open = \langle \langle f(C), C \rangle \rangle$
Let $Best = C$
While not terminated($Best$,$Open$)
        $BestOpen$=removefirst($Open$)
        $Open = \rho(BestOpen).Open$
        sort $Open$ in descending order
        $Best$=max($Best$,$BestOpen$)
Return $Best$

terminated(Best,$Open$) when
        $Open = \langle C_1, \ldots, C_n \rangle$
        and $n(C) = 0$ and $(p(C_1) - c(c_1)) \geq (p(C_i) - c(C_i))$ for $1 \leq i \leq n$