Machine Learning, 21, 81–101 (1995)
© 1995 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

DEXTER: A System that Experiments with Choices of Training Data Using Expert Knowledge in the Domain of DNA Hydration

DAWN M. COHEN dcohen@cs.pitt.edu
Keck Center for Computational Biology, University of Pittsburgh, Pittsburgh, PA 15260

CASIMIR KULIKOWSKI kulikows@cs.rutgers.edu
Department of Computer Science, Rutgers University, Piscataway, NJ 08855

HELEN BERMAN berman@dnarna.rutgers.edu
Department of Chemistry, Rutgers University, Piscataway, NJ 08855

Editors: David Searls, Jude Shavlik, and Lawrence Hunter

Abstract. In this paper, we describe a system, DEXTER, that uses knowledge to suggest *inductive learning experiments* in the domain of *DNA hydration pattern prediction*. These experiments vary the training data presented to a classifier learner. Such experiments are necessary in this domain, since, as in many other scientific domains, data are noisy, the relevance of particular attributes is not well established, and the number of training cases is limited. In each experiment, DEXTER chooses a set of training cases, attributes and classes to learn. To generate an experiment, it examines the results of previous experiments, and uses domain knowledge and domain independent heuristics to select and modify a previous experiment. For the domain expert interested in using the induced rules to understand data, DEXTER's explicit use of knowledge provides several advantages that other data selection techniques do not. In particular, the variation of classifiers induced in different experiments yields insights into the roles and interactions of particular attributes in determining hydration. In addition, many of the classifiers induced from DEXTER's choices of data are of accuracy greater than or equal to those induced using the entire set of available data or data chosen by several other techniques. This work is of theoretical and pragmatic importance to molecular biophysicists. The learned hydration predictors provide insights about factors influencing DNA hydration. Also, the hydration predictors could lead to a tool for automatically predicting water positions around DNA molecules for which crystallographic data are not available.

Keywords: DNA structure and hydration, inductive learning, experimentation, knowledge-based systems, exploratory data analysis

1. Introduction

Biological molecules are usually surrounded by water in nature. However, there is a growing body of biophysical studies that indicate that water provides more than an environment. Instead, it seems to play a role in the 3-D structure of macromolecules like protein and DNA. There is some evidence that water positioning is involved in interactions between DNA molecules (e.g. Schneider, et al., 1992), DNA and protein (e.g. Aggarwal, et al., 1988) and DNA and drug molecules (e.g. Neidle, Berman & Shieh, 1980). From x-ray diffraction studies, it appears that water molecules are often found in very specific arrangements around macromolecules. For these reasons, crystallographers are interested in knowing what aspects of a DNA molecule are responsible for its having a particular arrangement of waters, and thus, for its ability to interact correctly with other molecules. Information about how waters interact with macromolecules is crucial, but difficult to obtain by crystallographic analysis. Thus, there has been considerable interest in the biophysics community in identifying (1) the set of possible 3-D water arrangements or *hydration patterns*, (2) the conditions under which specific hydration patterns are observed in crystals, and (3) automated methods for predicting water positions around a given DNA molecule.

Until recently, most theories of DNA hydration have been based on a handful of known crystal structures. This was partly due to a dearth of publicly available crystal structure coordinate sets. However, with the recent expansion of the Nucleic Acid Database (NDB) (Berman, et al., 1992), there is beginning to be enough data to find the regularities in crystals. This database contains detailed information about a number of nucleic acid crystals that have been reported in the biophysics literature. The information includes data about procedures by which a crystal was obtained, the base sequence of molecules in the crystal, coordinates of the atoms in it, and various descriptions of the structure or shape within the molecules.

From the point of view of the crystallographer, the purpose of the present study is to apply machine learning techniques to data contained in the NDB, in order to induce rules for predicting the hydration pattern of a DNA molecule from its features. The usefulness of the learning process in this domain is two-fold. First, the induced rules are generally far more understandable than the raw data, and can be used by the expert to find out about the factors that contribute to DNA hydration. In this domain, where there is considerable controversy about what aspects of a DNA molecule influence its hydration, the expert is interested in finding out the underlying rules associating molecules with hydration patterns. Classifier learning is used as a kind of exploratory data analysis. The second purpose of learning is to obtain a method for classifying new cases (i.e. predicting hydration) that is simpler than the process by which the training cases were classified (i.e. solving a crystal structure). This method may be particularly useful when crystallographic data is not available for a new molecule.

In order to induce good classifiers, one must usually know a fair amount about the domain. Knowledge is *always* implicitly incorporated into the learning process by the choice of an appropriate set of training data. In particular, the knowledge engineer usually exercises some judgment to ensure that cases are chosen from a single population and that they are represented by a set of attributes believed to be useful for learning classifiers. The cases are labeled with classes that are assumed to be mutually exclusive. How data are chosen limits the classifier that can be induced from the data.

In principle, it would be possible to learn a classifier from the entire set of available data. In the hydration pattern prediction domain, as in many other scientific domains, this is undesirable for several reasons. First is the presence of noisy data which can greatly mislead greedy learning algorithms. Second, there are often several ways that data may be treated, which must be examined independently to determine the most appropriate. The expert may be able to bootstrap an understanding of the domain, by simultaneously using uncertain domain knowledge to select data to present to a learning algorithm while using the classifiers produced by the learning algorithm to extend the knowledge of the domain. By making different choices about which data to present to a learner (i.e. by *experimenting* with the inputs to the learning algorithm), an expert can explore the space of possible classifiers, in order to obtain a variety of insights about the biological system giving rise to the data.

This paper investigates the problem of automatically exploring the space of possible choices of training data or *learning experiments*, using knowledge. A framework has been developed for using domain specific and domain independent knowledge to select sets of training data automatically, in such a way as to generate better classifiers than using all possible data or to test specific "facts" of the domain. The framework has been implemented in a system called DEXTER (Data EXperimenter for Training using Expert Reasoning). The classifiers induced using DEXTER's choices of training data often have accuracy better than or equal to those produced by using all available training data. In addition, the classifiers produced using this knowledge-based method may be more plausible and give more insights into the domain than those produced using all available data. While it is necessary to generate multiple training sets in order to obtain these improved classifiers, the search space that is explored is far smaller than the combinatorial possibilities of considering all possible subsets of training data.

To be more specific, DEXTER represents and uses knowledge to select the training data to be used in a "learning experiment". A learning experiment in this framework consists of running a machine learning algorithm on a particular choice of training data. Learning experiments are generated by continually varying experimental conditions, namely the (not necessarily proper) subset of all possible training cases, the (not necessarily proper) subset of all possible attributes used to represent the cases, and the set of class labels used to classify cases.

DEXTER uses an iterative approach to experiment generation. It takes as input the entire set of training examples and a set of experiments that have already been run, along with their results. Knowledge and heuristics are used by the system to choose a previously run experiment which can be modified slightly to produce a new experiment. Knowledge and heuristics are also used to choose the modification in such a way that it tests a particular fact of the domain or is expected to produce an improved or distinct classifier.

A knowledge-based approach was considered appropriate in this domain because there is uncertain domain knowledge which could be brought to bear on learning. There have been studies linking qualitative, abstract hydration patterns with particular structural or chemical features of DNA molecules, usually for molecules of specific DNA types e.g., (Chuprina, et al., 1991; Eisenstein, 1990; Ho, et al., 1988; Prive, Yanagi & Dickerson, 1991). However, these studies have generally been limited in scope, taking into account only a small number of structures. Most give only anecdotal or descriptive accounts of hydration, rarely attempting quantitative definitions of hydration structure. There have also been contradictions between their conclusions (Berman, 1991). Thus, previous research has not led to effective methods for making quantitative predictions of water locations around new DNA molecules. The studies do provide some expectations, though, of attributes *likely* to influence the observed hydration patterns. It is some of these expectations that DEXTER attempts to test.

There are a number of existing techniques which address some of the learning issues studied here.

The "planning to learn" framework (Hunter, 1993) bears similarity to DEXTER in that it examines some of the decisions that must be made in order to apply learning techniques to a new problem, including choice of training set and attributes. However, unlike DEXTER, the framework is not concerned with an iterative approach to learning, where the results of one experiment are used to suggest others.

One aspect of DEXTER's task is the problem of choosing attributes for learning or "feature selection". Many techniques for this have been developed in the fields of pattern recognition and machine learning. Some carry out a search for subsets of features that are likely to produce more accurate classifiers than would be produced using the entire attribute set. These include branch and bound techniques, (Fukunaga, 1972; Narendra & Fukunaga, 1977), heuristic search (Siedlecki & Sklansky, 1988), FOCUS (exhaustive search within bounded size subsets) (Almuallin & Dietterich, 1991), bidirectional search (Salzberg, 1992), and the hill-climbing sequential forward selection approach (Weiss & Kulikowski, 1991). DT-select (Cherkauer & Shavlik, 1993) uses the attributes appearing in an induced decision tree to prune the attribute set for learning neural nets. RELIEF (Kira & Rendell, 1992) defines a quantitative notion of relevance of an attribute, and selects those attributes that have relevance above some threshold. These methods generally choose a single "best" set of attributes for the particular set of training cases, attempting, like the learning algorithms themselves, to maximize some function of the data. Unlike DEXTER, they do not provide any role for existing knowledge or for experimentation. One system which does use knowledge is F/I/E (Hunter & Klein, 1993). As one part of its classifier generation process, F/I/E makes use of the RELIEF algorithm to select attributes but uses some domain knowledge to select the most appropriate relevance threshold.

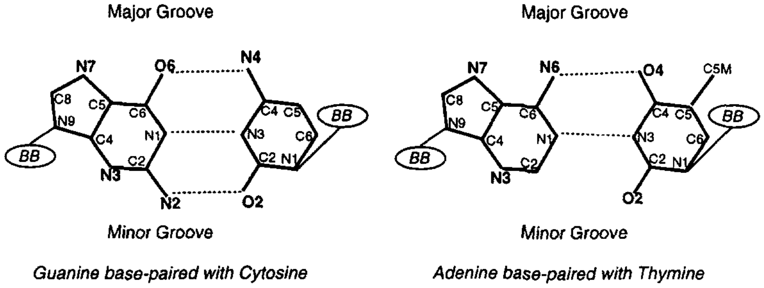
There are several rule refinement algorithms which allow for a simple type of experimentation in the form of iterative modifications to a classifier. These include SEEK2 (Ginsberg, et al., 1988), Swap-1 (Weiss & Indurkhya, 1991), RL (Provost, et al., 1993), FRINGE (Pagallo & Haussler, 1990) and APOS (Evans & Fisher, 1994). They attempt to change a previously obtained classifier or generate a new one in such a way as to improve its performance on a training set or to simplify it. Unlike DEXTER, they do not allow for any knowledge-based experimentation.

Several exploratory data analysis tools, such as EXPLORA (Klosgen, 1992) and the Knowledge Discovery Workbench (Piatetsky-Shapiro & Matheus, 1992) are meant to aid an expert to search for interesting patterns in data. However, they are mostly human-controlled and make little use of domain knowledge to aid in the search.

The rest of this paper is organized as follows. In Section 2 some background in the field of DNA structure and hydration is presented. In Section 3 DEXTER is described in detail. Section 4 presents some results of DEXTER's experiments and compares them with results of some other data selection methods. Section 5 presents some conclusions.

2. DNA Structure and Hydration

Much of the existing knowledge of DNA structure and function is derived from x-ray diffraction studies of crystals. In this type of study, a crystal is obtained of DNA molecules of a specific sequence. The crystal is composed of a repeating lattice of molecules, including DNA, water and ions. The goal of a diffraction study is to find 3-D coordinates of all of the atoms in the basic repeating unit, a process known as "solving the crystal structure". Knowledge of the coordinates of the atoms in the crystal allows the crystallographer to study the interrelationship between a molecule's structure, its physical interactions with other molecules and its function.



Guanine base-paired with Cytosine

Adenine base-paired with Thymine

Figure 1. Watson-Crick base-pairing scheme

For many years, crystallographers have reported the existence of intricate "hydration patterns" or networks of waters with hydrogen-bond connections to each other and to DNA. Some patterns of connectivity have been observed over and over again, across many crystals, like the "spine of hydration" observed in B-DNA dodecamers (Kopka, et al., 1983) and Z-DNA (Schnedier, et al., 1992). This has led researchers to examine the conditions under which particular networks do or do not arise. Most studies of hydration were based on detailed analyses of a single crystal structure, and did *not* attempt to predict water positions for new DNA molecules. If they did compare crystal structures at all, it was usually based on the presence or absence of hydrogen-bonds between specific sets of atoms.

While the connectivities of the networks seem to be consistent across crystals, the exact coordinates of the waters in each network are not identical. This presents some difficulty for any method that seeks to predict water coordinates based on previously observed patterns.

In this study, we consider a limited form of the hydration pattern prediction problem: namely, prediction of hydration near the genetic-code determining bases. By doing this, we can make use of a new technique from Schneider, et al. (1993) for describing hydration quantitatively, and consistently across crystal structures. This method identifies a set of regions where waters are often found around each type of base, with well-defined centers that can be used for coordinate prediction. To make the problem more amenable to machine learning techniques, we break it down further to predict independently the hydration around each base atom capable of hydrogen bonding with water. Thus, we transform the problem of predicting hydration around a DNA molecule to a set of subproblems. The bases are shown in Figure 1, with atoms capable of hydrogen bonding with water shown in bold.

More specifically, in this study, hydration patterns or classes are defined separately for each atom in each type of base. A hydration pattern corresponds to a subset of the possible regions of hydration identified in Schneider, et al (1993). A specific base in a specific structure has a particular hydration pattern near one of its atoms if the waters that are within hydrogen bonding distance of the atom fall exactly in the given hydration pattern's regions. If a case does not have any waters near a particular atom, its hydration pattern for the atom is *No hydration*.

In this framework, learning hydration pattern predictors consists of a set of learning subproblems: one for each type of atom in each type of base (e.g. adenine N3). A set of training cases represents individual bases from DNA molecules from a variety of crystal structures (e.g. all adenines). For any particular learning subproblem, a given case has a class label of the hydration pattern that is observed around the corresponding atom. Cases have a set of attributes that are believed to be relevant to predicting hydration.

From Schneider, Cohen and Berman (1992) and Schneider, et al. (1993), it appears that there are systematic differences between the hydration around bases in molecules of different "DNA types" (or "conformation types"). Thus, for example, the hydration regions that are found when one considers all cytosines of all crystal structures are slightly different from those found when one considers only the cytosines from molecules of type Z-DNA. In this study, we experimented with learning with cases labeled with classes defined by both kinds of regions.

A new DNA molecule's water positions may be predicted from the induced hydration pattern classifiers as follows. A hydration class is identified for one atom in a base in the molecule (e.g. adenine N3 of the adenine that is the fifth base in the molecule), using the induced classifier for the atom type (i.e. a classifier for predicting hydration near adenine N3's) for the base (i.e. the fifth base in the molecule, with its attributes). The predicted hydration class corresponds to some set of hydration regions, and waters are predicted that have coordinates that correspond to the centers of these regions. The details of this procedure are given in Cohen (1994).

The attributes for representing training cases should clearly be ones which have some impact on a base's hydration patterns. However, there is little certain knowledge of these. Indeed, there is some controversy over the factors determining hydration (see (Berman, 1991) for a review of these). Thus, one of the motivations for this study was to obtain evidence from the induced decision trees for the importance of particular attributes in DNA hydration.

The feature most commonly focused on in studies of hydration is the *DNA type* of a molecule. DNA type is a single overall description of the structure of a DNA molecule. It is correlated with a number of other attributes of DNA structure local to individual bases. It is believed to place many constraints on the free space for water around a molecule, as well as relative positions of DNA atoms that can facilitate the formation of certain hydrogen-bonding networks. It places limits on the values which may be taken by a number of other attributes. Indeed, because of this, many researchers tend to restrict studies to structures of a single DNA type. For this reason it may be appropriate to learn hydration predictors using cases of each DNA type separately. The most common types of DNA are A-DNA, B-DNA (both "right handed", coiling upwards and to the right) and Z-DNA (a "left handed" coil).

Researchers have studied the connections betweeń base *sequence* and hydration. Certain types of bases near other types of neighboring bases can allow for an exact positioning of hydrogen-bonding atoms in such a way as to allow water networks "bridging" the bases to form. Base *modifiers* have been observed to be related to hydration (e.g. (Ho, et al., 1988)). These are chemical modifications of the usual structure of the base. They may be bulky and discourage waters from hydrogen-bonding or they may be capable of hydrogen-bonding with waters, potentially allowing better networks. *Crystal packing* is another factor thought to have some role in observed DNA hydration. This describes how molecules interact with each other in crystals, leaving or omitting free space for waters between them. A number of *physical factors* are believed to affect what hydration patterns can form around a DNA molecule in a crystal. These include methods and experimental conditions used to grow crystals, such as presence or absence of ions like sodium or magnesium, crystallization solvents, etc.

Resolution, while not affecting the actual hydration present in a crystal, does affect what hydration is *observed*. Many waters may not be observable by crystallographic analysis at low resolution, though they are known to be present from independent physical measurements.

The above and a number of other attributes for describing the local structure around bases were used in learning classifiers. The details of these are given in Cohen (1994).

3. Organization of DEXTER

For DEXTER's purposes, machine learning experiments vary the set of training cases, the set of classes to learn and the set of attributes to learn on. In order to generate an experiment for a particular learning problem, DEXTER must make a selection for each of these. In principle, one could consider learning classifiers with every possible subset of training cases, every possible subset of classes to learn and every possible subset of attributes. Clearly, however, this requires time exponential in each set of choices, and is not feasible. If DEXTER is to be at all effective, it must select training sets from which classifiers can be induced that are better than using the entire training set, and it must do this by searching only a small fraction of the entire space of experiments. It does this by using explicit knowledge to prune or bias the search.

3.1. Overview of Experiment Generation

DEXTER's generation of learning experiments is driven by the user. That is to say the user specifies that a learning experiment should be generated for a particular learning problem (i.e. for an atom type whose hydration is to be predicted). From this point on, DEXTER autonomously generates the experiment.[1] Thus, the user decides *when* to generate experiments for a particular learning problem, and DEXTER, decides *how* to do this.

DEXTER generates a learning experiment in five stages:

1. Selection of a subset of cases.
2. Selection of a set of classes to learn.
3. Selection of a previous experiment whose attribute set will be modified, called the *template* experiment.
4. Selection of an attribute called the *focus attribute* of the template, to serve as a focus for changes in the attribute set.
5. Selection of an attribute set for the new experiment.

DEXTER has a set of rules for making each of these selections. The rules make use of domain knowledge as well as some common heuristics for machine learning experimentation.

The overall algorithm is as follows:

```
For each of the above stages do:
    1. Compute the set of satisfied rules.
       If there are none then signal this to the user and quit.
    2. Choose one of these rules and an instantiation of it.
    3. Apply the chosen rule, to make some selection.
    4. Check that the resulting choice will not violate some integrity
       constraint for learning experiments. If it does, go back to 2.
    5. Update the experiment information with the choice and any
       constraints on selections at later stages.
```

At each stage, DEXTER chooses randomly between the set of satisfied rules. However, the rules are ordered and those with higher precedence are chosen with higher probability than those with lower precedence. In this way, the experiment generation heuristics that one wishes to use more often can be favored, without completely neglecting the others. This is important since we wish to explore a relatively small portion of the space of possible learning experiments, and thus, would like to use mainly the rules expected to lead to interesting experiments; however, we would occasionally like to try different strategies.

DEXTER's rules have preconditions that depend on (1) selections made at earlier stages in the generation of the current experiment; (2) the set of previous experiments and their results; and (3) knowledge of the domain, stored in objects representing particular attributes or domain objects with various properties.

In addition to the rules used to select the training data, there are a number of rules which check that the selection does not create an invalid learning experiment (e.g. by selecting an empty set of training cases). If any of these "veto" rules finds that the data selection is invalid, DEXTER automatically backtracks to make a new selection.

The time needed at each stage of experiment generation is polynomial in the number of attributes, training cases, class labels and rules for making a selection. (This will be discussed below.) Thus, the overall time for generating an experiment is also polynomial.

In what follows, we describe the knowledge that DEXTER uses to generate experiments. Space does not permit an exhaustive discussion of all the rules, but we present many of the general strategies used. More details are given in Cohen (1994).

3.2. DEXTER's Knowledge For Selecting Sets of Training Cases

The rules that have been implemented for DEXTER to choose training cases all suggest sets that are known to be of interest to the expert. In particular, it has rules for suggesting the selection of (1) cases of a single DNA type (e.g. A-DNA cases only, B-DNA only, etc.); (2) cases from high resolution structures, only; and (3) all cases. Training sets of a single DNA type are of interest because experts would like to find out if there there are distinct rules for predicting hydration for different DNA types. Training sets including only high resolution data are of interest because it was believed that the certainty of both class labels and attribute values for these cases would be higher, and so, might lead to more accurate classifiers. All of these rules have equal precedence, and are selected with equal probability. Any selection of fewer than 20 cases is vetoed.

Clearly these rules are all domain specific. It might be possible to develop some general strategies for selecting subsets of training cases, but this has been left for future work.

The time needed to select the set of training cases with the current rules for selecting cases is proportional to the number of rules and cases. This is because the set of satisfied rules must be computed (each rule takes constant time, since the preconditions are very simple for the implemented rules) and then the selected cases are counted, to verify that there are at least 20.

3.3. Knowledge for Selecting Classes to Learn

For this domain, there is a well-defined metric of class similarity. In other words, given a particular class label, it is possible to say which is the most similar class in the entire set of class labels. Thus, when there is a class for which the data contain very few cases, it is possible to merge these cases with a better represented class. This simplifies the learning problem, making one fewer class to learn and more cases for another class. The induced rules may classify cases from the small class into the larger class that it was merged with. In this way, the learned classifier may predict more general classes with greater accuracy than one learned from cases labeled with all possible classes. The similarity measure between two classes is a function of the number of hydration regions they have in common and the euclidean distance between regions that are not common between them. The details of the class-similarity metric used here are presented in Cohen (1994).

DEXTER's rules for deciding which classes to learn always select the largest classes, and suggest merging the classes with the smallest numbers of members into larger classes.

How many classes are actually merged varies across experiments. One rule suggests using all classes for learning, since we would like to learn rules for all classes, if it is at all feasible. The second rule suggests that if it has been impossible to learn a classifier in all of the experiments that have been run so far for the current atom and current case set, then choose a number of classes to learn that has not yet been tried, but merge as few as possible. Another rule suggests that all small classes (i.e. having fewer than fifteen cases) should be merged with larger classes. None of the rules allows any class to be merged with the *No Hydration* class.

A choice of the set of classes to learn is vetoed if one class is overwhelmingly larger than all the others (i.e. if the number of members in the largest class is more than 5 times the number of members in the next largest).

The time needed for selecting the set of classes to learn, given the current set of rules, is proportional to the number of cases, experiments, rules and square of the number of classes. To determine if a rule is satisfied may require scanning the results of all of the experiments that have been run so far for the given atom. The time for finding the best large class to merge each small class into requires scanning the list of classes, and many of the classes may be small. To find out which classes are small requires counting the number of cases in each of the actual classes.

Since each small class can be merged into exactly one of the largest classes, DEXTER cannot consider learning with all possible subsets of the set of classes. It can only learn the largest classes. This bias considerably reduces the search through the space of possible experiments. While other strategies for selecting the classes to learn could be implemented, this is left to future work.

3.4. Domain Specific Knowledge Used by DEXTER

For the most part, domain specific knowledge in DEXTER is maintained in its representation of the atoms and of the attributes for learning.

Atoms are represented as objects in DEXTER. There are slots associated with each atom type which can be used when selecting training data. These slots include spatial information about the atom, including (1) whether or not it is involved in base-pairing (and if it is, with which other atoms); (2) which atoms in other bases occupy analogous physical locations (e.g. guanine N7 and adenine N7); (3) what type of base it is part of. This type of spatial information is useful on the assumption that atoms with similar spatial constraints may have similar hydration rules. Thus, information about the results of learning experiments for one atom may be used to suggest experiments for another.

Attributes for learning are represented abstractly as objects in DEXTER. Associated with each attribute is knowledge supplied by an expert about their possible roles in learning hydration classifiers. Knowledge includes:

- A qualitative measure of how important the attribute is likely to be for classification. This knowledge expresses the expert's belief that this attribute can (or is not likely to) discriminate well between hydration classes. For example, the attribute *resolution* would be expected to be very important for distinguishing between hydration and no hydration, since it determines how well waters can be crystallographically observed.

Several measures may be given, conditioned on particular learning problems or sets of training cases, if it is believed that the attribute is likely to be useful in classification to different extents. For example, *chain-length* was believed to be important for distinguishing between hydration and no-hydration for atom guanine N3 in B-DNA. It is known that B-DNA chains of length 12 are often arranged with their guanines flush against each other near N3, leaving no room for waters. On the other hand, B-DNA chains of length 10 tend to be arranged with plenty of free space for water around guanines.

- A qualitative measure of the certainty of values of this attribute. The purpose of this information is to be able to explore whether a noisy attribute's role in a classifier is noise fitting (e.g. whether an equally good classifier can be learned without it).
- A list of other attributes whose values may be constrained by the values of this attribute and might be interchanged with it to learn a similar classifier. The purpose of this information is be able to look for the "causes" of an attribute being important for classification. For example, the *guanine-N3-is-blocked?* attribute is a measure of whether a particular guanine N3 is actually flush against another DNA atom. N3 is known to be blocked more often for some chain lengths than others. Thus, one might guess that if *chain-length* appears in a learned hydration classifier for guanine N3 in B-DNA's, then so might *guanine-N3-is-blocked*, if *chain-length* is removed from the training data.
- For numeric attributes, a list of possible discretizations. The purpose of this information is to be able to explore whether the decision thresholds identified by a tree learning algorithm are "number-fitting" or whether they correspond to ranges meaningful to an expert. Classifiers induced from the discretized attributes may also be easier for the expert to understand.

3.5. DEXTER's Knowledge for Selecting Attribute Sets

DEXTER selects the set of attributes to be used for learning by modifying slightly the set of attributes used in a previous "template" experiment. Thus, it must first find an earlier experiment for the same atom with "interesting" results, and then it must find a way to modify the experiment to "tešt" these results, based on modifications to a "focus" attribute.

DEXTER begins with a set of manually generated (i.e. human designed) experiments and uses these to develop new ones. The manually generated experiments are described in section 4.2.

The rules for selecting the template experiment mainly involve scanning the set of previous experiments and their results, to find ones whose results should be tested. All of the rules prefer to suggest experiments that used the same set of training cases. Among these, the rules prefer to suggest the experiment whose classifier has the highest possible cross-validation accuracy.

One rule suggests that if the classifier induced in some experiment used an attribute that the expert believed would be unimportant in predicting hydration, then this is an interesting experiment to modify. If this rule is used, then a constraint must be recorded that states that the focus attribute should be chosen from among such attributes. Other rules suggest that if there are learning subproblems related to the current problem (such as an atom that base pairs with the current atom or that is analogous to it), then select an experiment that has been run for one of these related problems.

A choice of template experiment is vetoed if there is little overlap between the set of training cases chosen for the current experiment and the template experiment (i.e. $< 30\%$ of the cases are used in both experiments). This is done on the assumption that if there is little similarity between the training sets, there is little reason to believe that template's classifier will be at all similar to that induced in the current experiment.

The time needed for selecting a template, given the current set of rules, is proportional to the number of rules, the number of previous experiments for the current atom or related atoms and the number of attributes. This is because in order to find out whether a rule is satisfied, it may be necessary to scan the entire list of previous experiments and the list of attributes, to find out if some experiment classifier used an attribute with some property (such as being "unimportant").

Once a template experiment has been selected, it is necessary to identify the focus attribute whose use will be changed somehow between the template and the current experiment.

One rule for selecting a focus attribute suggests that if the template's classifier used any attributes that the expert considers noisy, then select one of these as the focus attribute. Two other rules suggest that if the template's classifier used any attributes that were expected to be either irrelevant or very important in predicting hydration, then select one of these to change, and test its role. Another rule suggests that if the classifier induced in the template used any numeric attributes that can be discretized, then select one of these as the focus attribute. If this rule is chosen, then DEXTER must record the constraint that the change in the final training set must be the discretization of the focus attribute.

A choice of focus attribute is vetoed if it is not applicable to bases of the given type (e.g. *guanine-N3-blocked?* is only meaningful for guanines). It is also vetoed if most of the selected cases have the same value for this attribute. (Changes to such attributes would be expected to have minimal impact on classification and make for a rather uninteresting experiment).

The time needed to choose a focus attribute, given the implemented rules, is proportional to the number of rules, cases, and attributes. To determine if a rule is satisfied may require scanning the results of the template experiment and the attributes list to find ones with appropriate properties. Checking the range of the focus attribute for the chosen training set requires checking the value of the attribute on each of the cases.

Once the focus attribute has been selected, there are several rules for using it to modify the attribute set used in the template experiment. One rule suggests that if the focus attribute is known to be noisy, then remove it to see if classifier performance can be improved without it. Another rule suggests that if the focus attribute is known to be very important, remove it from the training set, to see if classifier performance greatly decreases. Other rules suggest that if the focus is a numeric attribute then discretize it or all numeric attributes in the template's attribute set. Another rule suggests replacing the focus attribute with another attribute whose values are believed to be constrained by the focus attribute.

One domain specific rule for selecting the attribute set is the following. If the focus attribute is *resolution* then remove it from the template attribute set, since we would like to know if there are other attributes capable of discriminating between hydration classes, besides this obvious one.

The time needed to select the attribute set, for the rules implemented here, is proportional to the number of rules and attributes. To determine if a rule is satisfied requires constant time, since it generally mentions only properties of the focus attribute. To select the attribute set may require time proportional to the number of attributes, since the entire set may have to be scanned to remove attributes with particular properties.

3.6. The Space of Learning Experiments Explored by DEXTER

In summary, DEXTER's use of knowledge to select training data allows it to explore a small portion of the space of possible learning experiments. The space that it can actually search is bounded by the set of choices:

1. Case set = {A-DNA | B-DNA | Z-DNA | High-resolution | All}
2. Class set = {2 largest classes | 3 largest classes | ... | All classes}
3. Attribute set = Subsets of all possible attributes and their discretizations.

In practice, DEXTER will only search a portion of this space. Depending on the choice of cases, the number of classes that can be learned will be constrained and the ways of choosing attribute sets will be constrained. In addition, since the user decides when to generate a new experiment, it unlikely that the user will care to allow DEXTER to explore the entire space. However, the rules have been designed and ordered in hopes of DEXTER's generating the more interesting portions first.

4. Experiments and Results

There are two main domain questions which have driven this research: (1) Given a description of a DNA molecule, is it possible to predict locations of water molecules around it in a crystal? (2) Is it possible to identify factors that determine the hydration patterns around a DNA molecule? Our approach to answering these questions is to learn hydration pattern classifiers, based on various subsets of training data. We then examine the induced rules (and their accuracy) and how these change with changes in the training data. In order to address the domain questions most effectively, different sets of training data should be chosen so that the induced classifiers are as accurate as possible and/or so that the importance of specific attributes in determining hydration can

Table 1. NDB ID's and strand sequences of crystal structures included in study.

NDB ID	Sequence	NDB ID	Sequence	NDB ID	Sequence
ADDB01	CCGG	BDJB27	CCAGGCCTGG	ZDF001	CGCGCG
ADH007	GGGATCCC	BDL001	CGCGAATTCGCG	ZDF002	CGCGCG
ADH008	GCCCGGGC	BDL002	CGCGAATTCGCG	ZDF028	CGCGCG
ADH023	GTACGTAC	BDL007	CGCATATATGCG	ZDF029	CGCGCG
ADH024	GTACGTAC	BDL015	CGCAAAAATGCG	ZDFB04	CGCGCG
ADHP36	GCCCGGGC	BDL020	CGCGAATTCGCG	ZDFB05	CGCGCG
ADJ022	ACCGGCCGGT	BDL022	CGCAAGCTGGCG	ZDFB10	CGUACG
ADL025	CCCCCGCGGGGG	BDL028	CGTGAATTCACG	ZDFB11	CACGTG
BDBP23	CG	BDL029	CGTGAATTCACG	ZDFB12	CGCGUG
BDJ008	CCAAGATTGG	BDLB03	CGCGAATTCGCG	ZDFB21	CGCGCG
BDJ017	CCAGGCCTGG	BDLB04	CGCGAATTCGCG	ZDFB24	CGUACG
BDJ019	CCAACGTTGG	BDLB13	CGCGAATTCGCG	ZDFB31	CGUACG
BDJ025	CGATCGATCG	BDLB26	CGCGAATTTGCG	UDB004	CG
BDJ031	CGATTAATCG	ZDB020	CG	UDB005	CG
UDB007	TT	UDB012	AA	UDD006	ATAT
UDM010	CGCAGAATTCGCG	UDP011	CGCGCGTTTTCGCGCG		

be studied. A number of automatic feature selection techniques have been developed in the field of pattern recognition to improve the accuracy of learned classifiers, with respect to those learned using all possible data. As will be shown below, DEXTER has been able to select data that led to classifiers of accuracy superior or comparable to the data selected in several other approaches, for most of the hydration pattern prediction problems. In addition, however, DEXTER's knowledge-based, incremental approach has made it possible to understand the role of particular attributes, which other methods do not.

4.1. Data

The data used in this study were taken from the Nucleic Acid Database (Berman, et al., 1992). Bases from 47 DNA crystal structures were used. These are shown in Table 1. The sample of bases (i.e. cases) used in learning hydration classifiers is shown in Table 2. Though all crystals contained only DNA, and DNA generally only contains the bases adenine, cytosine, guanine and thymine, in these crystals there are also several uracils which were introduced into artificially synthesized DNA. Also, the number of cytosines does not always match the number of guanines (or adenines the thymines) because several of the structures included mismatched bases. For all learning problems corresponding to atoms from a particular base type, the cases have the same attribute values, and they differ only in their class labels, representing the hydration pattern around atoms. The number of classes to be learned for each atom type, for cases in each DNA type are shown in Table 2. All cases were represented with 11 categorical attributes and 67 numeric attributes. In addition, adenine cases had three other attributes, cytosine two more, guanine four more and thymine two more (all of these were categorical). The reader is referred to Cohen (1994) for descriptions of the attributes used.

Table 2. Number of cases and hydration classes for each atom, for training case sets used in this study.

		DNA type					High Res.
Base/Atom Type		*A*	*B*	*Z*	*U*	*All types*	*(all types)*
Adenine	cases	7	80	8	12	107	49
A-N3	classes		2			2	2
A-N6	classes		3			4	4
A-N7	classes		4			4	4
Cytosine	cases	38	121	64	17	240	160
C-O2	classes	3	3	5		5	5
C-N4	classes	3	3	4		4	4
Guanine	cases	38	126	66	17	247	165
G-N2	classes	3	3	3		4	4
G-N3	classes	2	3	1		3	3
G-O6	classes	3	4	6		8	8
G-N7	classes	3	3	7		6	5
Thymine	cases	7	79	2	12	100	46
T-O2	classes		3			4	4
T-O4	classes		4			4	4
Uracil	cases	0	0	8	0	8	8
Total	cases	90	406	148	58	702	420

4.2. Computations

In order to study DEXTER's performance and compare it to other attribute selection methods, the following learning experiments were generated for each of the learning subproblems (i.e. for each atom for which hydration pattern classifiers were to be learned).

Manually-generated experiments

Several experiments were generated manually (i.e. a human chose the data). These represented learning with different subsets of cases and attributes. For cytosine and guanine atom learning subproblems, experiments were performed using (1) all cases with all 80-82 attributes; (2) A-DNA, B-DNA and Z-DNA and all cases, using a set of 15 attributes (6 categorical and 9 numeric) and (3) using the same data as in (2) but with 5 additional noisy numeric "ν" attributes. For adenine and thymine learning subproblems, the same experiments were run, except for those using A-DNA and Z-DNA cases, since there were too few of these.

DEXTER-generated experiments

Fifteen experiments were generated by DEXTER, representing fifteen different choices of training data. DEXTER initially modified manually generated experiments to generate new experiments. It was allowed to use the same sets of training cases as were used in the manually generated experiments, and one more: those from molecules of high resolution structures (of any DNA type).

Experiments generated by SFS

Stepwise Feature Selection (SFS) is a hill-climbing technique which starts with an empty set of selected attributes and continually adds the single attribute which, when added to the selected set, leads to the most accurate induced classifier (see Weiss & Kulikowski, 1991). This is clearly a fairly computationally intensive technique, as it requires a classifier to be induced for each candidate to add to the set selected so far. Thus, in this study it was considered practical to limit the attributes considered to a set of 15 (those used in the manual experiments). Thus, 120 experiments were generated, representing learning with all cases, and attribute sets selected by SFS (15 experiments to select the best single attribute, 14 to select the best of the remaining attributes to add to the single best, and so on).

Experiment generated with RELIEF

One experiment was generated, representing learning with attributes chosen by the RELIEF feature selection method (Kira & Rendell, 1992), using all cases. This method computes a "relevance" for each attribute, which roughly represents an average difference between values of the attribute for different classes. Attributes with a tolerance above some chosen threshold, τ, are selected for learning. In this study, τ was set at the relatively low level of 0.05, leading to a comparatively large set of attributes selected from the entire set.

The data selected in each experiment was input to the CART decision tree learning system. Error rates of induced classifiers were estimated using the n-fold cross-validation procedure (also known as "leave-one-out"). In this procedure, for each case a classifier is induced using all of the other cases and tested on the omitted one. The error rate is the percentage of cases misclassified. CART provides several heuristics for determining the best attribute to split on at each stage of tree-construction. In this study, the *gini index* was used as the splitting rule. CART also automatically prunes trees, to avoid overfitting data, and provides several different pruning strategies. This study used the 0.0 S.E. pruning rule, a relatively conservative strategy (i.e. it removes few nodes). This rule prunes away subtrees in such a way as to leave a tree with minimum estimated error rate (see (Breiman, et al., 1984) for details about CART).

4.3. Comparison of DEXTER with Other Attribute Selection Methods

In order to establish that for many learning subproblems DEXTER is capable of generating data sets that give rise to more accurate classifiers than the other data selection methods studied, the lowest error rates of any classifier obtained from each method are compared. It can be seen from Table 3 that for many atoms, DEXTER was, indeed, able to obtain data sets that give rise to better error rates than the others. (Entries in this table are sorted by the improvement of DEXTER's best choice over the best manual choice of data.) In this table, "DEXTER Best Error (%)" refers to the lowest error rate for classifiers induced in any experiment generated by DEXTER for the given atom. "All Data Error (%)" refers to the error rate for the data set including all cases and all attributes, with cases classified by all classes. "Manual Best Error (%)" refers to the lowest error rate of any classifier obtained from any of the manually chosen data sets. "SFS Best Error (%)" is the lowest error rate of any classifier obtained during stepwise feature selection. "RELIEF Error (%)" is the error rate of the classifier induced from the attributes selected by the RELIEF algorithm (if any).

From this table, it can be seen that for most of the atoms, DEXTER could identify some set of training data that led to a more accurate classifier than the other methods.

Table 3. Comparison of best results of DEXTER and several other methods.

Atom	DEXTER Best Error (%)	All Data Error (%)	Manual Best Error (%)	SFS Best Error (%)	RELIEF Error (%)
ADE N7	15	36	32	27	No Attributes
CYT N4	11	31	26	28	44
GUA N7	23	39	33	34	40
ADE N3	14	24	20	17	No Attributes
CYT O2	8	28	13	25	19
THY O2	19	27	24	18	31
GUA N3	8	14	12	9	11
GUA O6	26	37	27	35	No Attributes
THY O4	24	36	23	25	No Attributes
ADE N6	24	31	22	25	31
GUA N2	15	15	10	15	14

Though the space of experiments that *could* be generated by DEXTER for each atom is quite large, only 15 experiments were generated per atom to obtain these results. For several atoms, none of the 15 experiments proposed by DEXTER could obtain more accurate classifiers than were obtained manually.

4.4. Results of Interest in the Domain

Several of the more interesting results from the point of view of the domain are described below.

Almost all of the induced classifiers for predicting hydration around adenine N3 and thymine O2 have included the attribute *3'-neighbor*. This is interesting because these are the base atoms which are involved in the B-DNA spine of hydration, and it had been hypothesized that local sequence can affect the formation of this network.

It was initially expected that learning experiments using cases from high resolution structures would give rise to classifiers with greater accuracy than similar experiments using mixtures of high and low resolution cases. This was not found to be the case in this study. In almost all experiments, learning with high resolution cases (cases from structures of resolution $0-2.1\AA$) led to *poorer* classifiers than using all data. The reasons for this will have to be investigated further. It is possible that it is due to the fact that the high resolution cases tend to be dominated by the Z-DNA's which are hard to learn because they generally have cases distributed over many small classes. An exception to this observation is the atom adenine N3. In this case, it was possible to learn even more accurate rules for high resolution cases than for other training sets.

The location of water around cytosine N4 is related to the presence of bulky *modifiers*, such as bromine or a methyl group on the nearby C5. This attribute appeared in most classifiers for this atom, regardless of case set.

As was expected, crystal structure *resolution* was quite important in predicting whether or not hydration is observed. This attribute was included in classifiers for many of the subproblems. Also, *chain-length* and *resolution* were often interchangeable in inducing classifiers.

One example of a case where a specific fact was tested in an experiment led to a surprising result is as follows. In one experiment for learning to predict hydration near guanine N3 in B-DNA, a tree was induced consisting of a single decision based on the attribute *chain-length.* DEXTER proposed an experiment to substitute the attribute *chain-length* with *guanine-N3-is-blocked.* The result was a classifier with a single decision based on *resolution.* A subsequent experiment sought to test this result, by removing *resolution*, but leaving in *guanine-N3-is-blocked.* Surprisingly, in this experiment, no classifier could be induced (i.e. the tree was pruned away to a single node). This indicates that *packing* is not a sufficient explanation for lack of hydration near guanine-N3 in B-DNA.

Somewhat surprisingly, among the experiments in which DEXTER proposed using template experiments from "analogous" atoms, almost none seemed to lead to either very accurate or similar classifiers. When they did lead to accurate classifiers, it was generally the case that the number of classes to learn had been greatly reduced. However, there were only about half a dozen experiments of this type, and it is hard to draw conclusions about whether the strategy of using related subproblems to select templates could be more effective than is apparent from this study.

Classifiers induced using training cases of a single DNA type often had very different accuracies and rules from those induced using all cases. This partially supports the belief that molecules of different DNA types have distinct hydration patterns and factors influencing hydration. However, *DNA-type* itself rarely appeared explicitly in induced rules.

5. Conclusion

In any machine learning application, the choice of training data is critical in being able to induce accurate and understandable classifiers. Considerable effort may be spent during a learning study to select appropriate data, which is sometimes called "defining the learning problem" or "massaging the data". A researcher may carry out many learning experiments in the course of a study, varying the attributes for learning, the set of cases, or even the classification problem itself. Typically, these learning experiments are designed by a human researcher, who knows enough about the domain to decide how to select data in such a way as to induce better classifiers or help understand the importance of their choices in classifier induction. In this paper we have described a system, DEXTER, which carries out this process automatically, using explicitly encoded knowledge.

There are a number of aspects of the DNA hydration pattern prediction domain which make a knowledge-based approach desirable. The data are noisy, training sets are small, and a large number of *potentially* relevant attributes are available for learning. However, there is uncertain knowledge which can guide the search for "interesting" choices of training data.

DEXTER is meant to select sets of training data in such a way as to obtain more accurate, more plausible and more varied classifiers than are obtained by using all possible data. In addition, it is meant to allow various pieces of domain knowledge to be tested, by suggesting specific changes in training data from one experiment to another. It was not possible to achieve these goals for every learning subproblem in the hydration pattern prediction domain. However, the results indicate that each goal has been achieved for some learning subproblems. For most of the learning subproblems, DEXTER could identify some training sets that led to more accurate classifiers than either learning on the entire set of training data or on sets chosen by several other methods. It has provided new evidence about what features of a DNA molecule may be important in predicting hydration and it has exposed several domain issues which should be studied in more detail. Clearly, however, these results come at the price of knowledge engineering effort and the computational cost of inducing a variety of classifiers in the experiments proposed by DEXTER.

One difficulty which arises in the evaluation of DEXTER is that some of the insights which have been obtained are not due to single experiments generated by DEXTER, but rather emerge from the results of several experiments. In addition, DEXTER allows several changes to take place between a template experiment and the one being generated. While the rules it uses have been biased against this sort of behavior, it is sometimes difficult to determine which training data change was most responsible for changes in the induced classifier. In addition to these limitations in DEXTER, it is also important to note that the induced rules are only as good as the training data. The sample of cases is fairly biased (for example there are a few distinct groups of sequences, but within each group, the sequences are quite similar). Also the classification of training cases is subject to considerable noise, since many waters may not be crystallographically observable.

It is *not* our claim that the experiments generated by DEXTER could not also be generated by an expert manually. In fact, DEXTER *cannot* generate any experiments that could not (at least in principle) be generated by an expert. This is because it relies on explicitly stated knowledge provided by the expert. However, one advantage of DEXTER's automatic experiment generation in the hydration pattern prediction domain has to do with the fact that there are several related learning subproblems, rather than a single learning task. For example, the same discretizations of an attribute apply to all subproblems, and the expert need not *explicitly* remember to try each discretization for each problem. Another potential advantage of using DEXTER is that it allows experiments to be generated which might be ruled out immediately by an expert, that could nevertheless lead to interesting results. Finally, by its systematic use of explicitly coded knowledge, DEXTER can combine knowledge and experimental results in ways unexpected by the expert. This can lead to useful experiments that might not have occurred to the expert.

While there are other systems that are capable of selecting training data from a given set, they generally make use of quantitative, general-purpose heuristics, rather than knowledge of the domain or of learning experiments. DEXTER, on the other hand, allows the kinds of knowledge used by an expert to be represented and used automatically to generate learning experiments that attempt to improve classifier performance or give insights into the meaning of the data.

DEXTER has only been tested in the DNA hydration pattern prediction domain. However, knowledge from other domains could, in principle, be engineered as input to DEXTER, in order to facilitate learning experimentation for other scientific problems. Future work includes development of a truly domain-independent tool with a knowledge acquisition component to simplify the knowledge engineering for an expert wishing to experiment with learning.

Acknowledgments

Debbie Smith and Jordi Bella helped greatly with using some of the crystallography software. Some code for superimposing bases was provided by A.R. Srinivasan. The staff of the Nucleic Acid Database was extremely helpful in making data available and accessible. Thanks to Haym Hirsh for helpful comments on previous drafts of this paper and the research. Thanks also to Mick Noordewier for helpful comments on the research. Much of the foundational work for this study was done with Bohdan Schneider.

This material is based on work supported by the National Institutes of Health under grants GM21589 and GM08319. The Nucleic Acid Database is supported by the National Science Foundation under grant NSF BIR 90 12772.

Notes

1. In about a quarter of the experiments, DEXTER did solicit input from the user, asking for the number of classes to learn. It did this because in generating these experiments, it could not use any of its other rules for selecting the number of classes. In two other experiments, the user deliberately made choices for DEXTER.

References

Aggarwal, A.K., Rodgers, D. W., Drottar, M., Ptashne, M. & and Harrison, S.C. (1988). Recognition of a DNA operator by the repressor of Phage 434: A view at high resolution. *Science*, 242:899–907.

Almuallin, H. & Dietterich, T.G. (1991). Learning with many irrelevant features. In *Proceedings of the Ninth National Conference on Artificial Intelligence*, pages 547–552. Anaheim, CA: AAAI Press.

Berman, Helen. (1991). Hydration of DNA. *Current Opinions in Structural Biology*, 1(3).

Berman, H.M., Olson, W.K., Beveridge, D.L., Westbrook, J., Gelbin, A., Demeny, T., Hsieh, S.-H., Srinivasan, A.R. & Schneider, B. (1992). The nucleic acid database: A comprehensive relational database of three-dimensional structures of nucleic acids. *Biophysical Journal*, 69:751–759.

Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, C.J. (1984). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

Cherkauer, K.J. & Shavlik, J.W. (1993). Protein structure prediction: Selecting salient features from large candidate pools. In *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology*, pages 74–82. Bethesda, MD: AAAI Press.

Chuprina, V.P., Heinemann, U., Nurislamov, A.A., Zielenkiewicz, P. & Dickerson, R.E. (1991). Molecular dynamics simulation of the hydration shell of a B-DNA decamer reveals two main types of minor-groove hydration, depending on groove width. *Proceedings National Academy Science*, pages 593–597.

Cohen, Dawn M. (1994). *Knowledge-Based Generation of Machine Learning Experiments: Learning to Predict DNA Hydration Patterns*. PhD thesis, Rutgers University.

Eisenstein, M., Frolow, F., Shakked, Z. & Rabinovich, D. (1990). The structure and hydration of the A-DNA fragment d(GGGTACCC) at room temperature and low temperature. *Nucleic Acids Research*, 18(11):3185–3194.

Evans, B. & Fisher, D. (1994). Process delay analysis using decision tree induction. *IEEE Expert*, 9:60.

Fukunaga, K. (1972). *Introduction to Statistical Pattern Recognition*. Academic Press, New York.

Ginsberg, A., Weiss, S.M. & Politakis, P. (1988). Automatic knowledge base refinement for classification systems. *Artificial Intelligence*, 35:197–226.

Ho, P.S., Quigley, G.J., Tilton, R. F. & Rich, A. (1988). Hydration of methylated and nonmethylated B-DNA and Z-DNA. *Journal of Physical Chemistry*, 92(4):939–945.

Hunter, L. (1993). Planning to learn about protein structure. In L. Hunter, editor, *Artificial Intelligence and Molecular Biology*, pages 259–288. AAAI Press, Menlo Park, CA.

Hunter L. & Klein, T. (1993). Finding relevant biomolecular features. In *Proceedings of the First International Conference on Intelligent Systems for Molecular Biology*, pages 190–197. Bethesda, MD: AAAI Press.

Kira, K. & Rendell, L.A. (1992). The feature selection problem: Traditional methods and a new algorithm. In *Proceedings of the National Conference on Artificial Intelligence*, pages 129–134. San Jose, CA: AAAI Press.

Klosgen, W. (1992). Problems for knowledge discovery in databases and their treatment in the statistics interpreter EXPLORA. *International Journal of Intelligent Systems*, 7(7):649–673.

Kopka, M.L., Frantini, A.V., Drew, H.R. & Dickerson, R.E. (1983). Ordered water structure around a B-DNA dodecamer. a quantitative study. *Journal of Molecular Biology*, 163:129–146.

Narendra, P.M. & Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Trans. Comp.*, 26:917–922.

Neidle, S., Berman, H.M. & Shieh, H.S. (1980). Highly structured water networks in crystals of a deoxydinucleoside-drug complex. *Nature*, 288:129–133.

Pagallo, G. & Haussler, D. (1990). Boolean feature discovery in empirical learning. *Machine Learning*, 5:71–99.

Piatetsky-Shapiro, G. & Matheus, C.J. (1992). Knowledge discovery workbench for exploring business databases. *International Journal of Intelligent Systems*, 7:675–686.

Prive, G.G., Yanagi, K. & Dickerson, R.E. (1991). Structure of the B-DNA decamer CCAACGTTGG and comparison with isomorphous decamers CCAAGATTGG and CCAGGCCTGG. *Journal of Molecular Biology*, 217:177–199.

Provost, F.J., Buchanan, B.G., Clearwater, S.H., Lee, Y. & Leng, B. (1993). Machine learning in the service of exploratory science and engineering: A case study of the RL induction program. Technical Report ISL-93-6, Computer Science Department, University of Pittsburgh.

Salzberg, S. (1992). Improving classification methods via feature selection. Technical Report JHU-TR-92-12, Johns Hopkins University.

Schneider, B., Cohen, D. & Berman, H. (1992). Hydration of DNA bases: Analysis of crystallographic data. *Biopolymers*, 32:725–250.

Schneider, B., Cohen, D.M., Schleifer, L., Srinivasan, A.R., Olson, W.K. & Berman, H.M. (1993). A systematic method for studying the spatial distribution of water molecules around nucleic acid bases. *The Biophysical Journal*.

Schneider, B., Ginell, S.L., Jones, R., Gaffney, B. & Berman, H.M. (1992). Crystal and molecular structure of a DNA fragment contalning a 2-aminoadenine modification: The relationship between conformation, packing and hydration in Z-DNA hexamers. *Biochemistry*, 31:9622–9628.

Siedlecki, W. & Sklansky, J. (1988). On automatic feature selection. *International Journal of Pattern Recognition and Artificial Intelligence*, 2:197–220.

Weiss, S. & Indurkhya, N. (1991). Reduced complexity rule induction. In *Proceedings of IJCAI-91*, pages 678–684. Sydney: Morgan Kaufmann.

Weiss, S.M. & Kulikowski, C.A. (1991). *Computer Systems That Learn*. Morgan Kaufmann, San Mateo, CA.

Received October 12, 1993
Accepted August 18, 1994
Final Manuscript November 7, 1994