# How to predict it: Inductive Prediction by Analogy Using Taxonomic Information

**Takashi Ishikawa**
Kisarazu National College of Technology
2-11-1 Kiyomidai-higashi, Kisarazu, Chiba 292, JPN
takashi@j.kisarazu.ac.jp

**Takao Terano**
The University of Tsukuba
3-29-1 Otsuka, Bunkyo-ku, Tokyo 112, JPN
terano@gssm.otsuka.tsukuba.ac.jp

**Abstract**

This paper presents a novel machine learning technique in a logic programming environment: *Inductive Prediction by Analogy* (IPA). IPA learns the description of a target predicate similar to a source predicate from examples of the target predicate. A key feature of IPA is that it uses analogies to constrain the space of hypotheses using taxonomic information represented by first-order predicate logic. Typical problems addressed by IPA are to decide whether a given ground atom is valid or not, when no concept descriptions for the goal are available in a knowledge base. This is attained by the steps: 1) recognition of a candidate analogous source, 2) elaboration of an analogical mapping between source and target domains, 3) evaluation of mapping and inferences to given examples of the target predicate, and 4) consolidation of the outcome of the analogy. IPA can be applied to a wide variety of problems including classification problems in inductive learning. An experimental system of IPA is implemented in Prolog in order to use it as a knowledge acquisition tool for knowledge-based systems. The effectiveness of the technique is validated by a real world problem in molecular biology: the function prediction of proteins from their amino acid sequences.

## Introduction

Analogical reasoning is an important research area in AI as a technique to reason from incomplete knowledge. In problem solving and learning, analogical reasoning promises to overcome the explosive search complexity of finding solutions to novel problems or inducing generalized knowledge from experiences (Hall 1989). The approach of the paper utilizes analogical reasoning in concept-learning. The key issue in this approach is how to recognize automatically an analogy between a source and a target and apply it to generating hypotheses for the target domain. Previous techniques in this approach usually use oracles from user to select candidates (De Raedt & Bruynooghe 1992) (Kedar-Cabelli 1985). Whereas the technique in this paper uses taxonomic information in the knowledge base for this purpose.

This paper presents a novel machine learning technique in a logic programming environment: *Inductive Prediction by Analogy* (IPA). IPA learns the description of a target predicate similar to a source predicate from examples of the target predicate. A key feature of IPA is that it uses analogies to constrain the space of hypotheses using taxonomic information represented by first-order predicate logic. Taxonomic information describes classification of symbols of a knowledge representation language. In a logical framework the symbols are constants to represent predicates, functions, and constant terms in sentences. IPA technique is based on an analogy as a mapping between constant symbols of a source predicate and a target predicate. Requirements for the technique are that it should be easy to represent by Horn clauses and should be easy to implement in Prolog as syntactic operations.

One of main objectives to develop IPA technique is to provide molecular biologists with an easy way to predict functions of a lot of proteins from their amino acid sequences in various kinds of database. Although there are so many amino acid sequence data available, conventional methods in molecular biology require tremendous and expensive efforts to predict the functions of proteins. They need novel but easy methods. Using AI-based symbolic techniques we will solve such real world problems.

This paper is organized as follows. The second section presents the framework for the

Inductive Prediction by Analogy and defines analogy using taxonomic information. The third section describes IPA technique in detail and gives an algorithm of IPA. In the fourth section, we report an experiment on IPA technique in molecular biology applied to the function prediction of proteins from amino acid sequences. In the fifth section we discuss the strengths and limitations of IPA technique and related work. Concluding remarks will follow in the final section.

## The framework for the Inductive Prediction by Analogy

Inductive prediction consists in finding an inductive generalization of a set of examples of a concept and in applying it in order to predict whether a new instance is (or is not) a positive example of the concept (Tecuci 1993). In general, the process of inductive prediction is to generate concept-descriptions from examples. We use a logical framework for concept-learning (Genesereth & Nilsson 1987) (De Raedt & Bruynooghe 1992). In IPA technique, we assume that the knowledge base is represented by Horn clauses.

**Definition 1.** (Concept)
(1) A *concept* is a predicate.
(2) A *concept-description* is a set of (definite) Horn clauses defining a predicate.
(3) *Examples* are ground instances of a predicate. *Positive examples* are true and *negative examples* are false.
(4) A concept-description *covers* an example, if the example logically follows from the concept-description and the knowledge base.

A goal of the inductive prediction process is to generate hypotheses from which a target goal logically follows from the clauses in the knowledge base. The inductive prediction process can be used when a query in deductive inference fails because of the definition of the predicate is not defined in the knowledge base (Michalski & Tecuci 1994). We use a target goal as bias to restrict the form of hypotheses (Utgoff & Mitchell 1982). This type of bias focuses the concept-learner on generating hypotheses to cover a given target goal.

**Definition 2.** (Hypothesis)
(1) A *hypothesis* is a concept-description of a predicate not defined in the knowledge base.
(2) A *justified hypothesis* covers all positive examples and no negative examples.

**Definition 3.** (Inductive prediction)
(1) *Inductive prediction* is to generate justified hypotheses covering a target goal.
(2) A *target goal* is a ground instance of a predicate not defined in the knowledge base.

## Inductive Prediction by Analogy

### Analogy using taxonomic information

Analogical reasoning is a type of plausible reasoning based on the following assumption:

**Assumption.** If an analogy between a source and a target exists, then properties of the source can be projected to the target.

The notion *analogy* is defined informally as a representational mapping from the source to the target. We formalize *analogy* using taxonomic information in the following way. We use the terminology *source*, *target*, *mapping*, *analogy*, and *support* same as (Hall 1989). First, we formalize taxonomic information as a notion of *sort* (Frisch & Page 1990), then we define analogy.

**Definition 4.** (Sort)
(1) A *sort* $\tau$ is a subset of constant symbols of the domain. If constant $c$ belongs to a sort $\tau$, then we describe it as a ground clause $\tau(c) \leftarrow$.
(2) $\tau'$ is a *subsort* of a sort $\tau$ if, and only if, $\tau(X)$ can be deduced from $\tau'(X)$ and the knowledge base.

An *analogy* is considered as a mapping between elements of a source domain and a target domain. The *analogical mapping* associates or maps elements and descriptions from the source domain into the target domain. These mapped elements are analogical inferences and receive varying levels of supports from other mapped elements. This predicate mapping restriction constraints the space of possible clause mappings. IPA technique employs the observed similarity by mapping constants in concept-descriptions. Taxonomic information has the role of defining similarities among concept-descriptions. This mapping specifies the analogy among predicates. An analogy between literals is defined as follows:

**Definition 5.** (Analogy)
(1) A literal $L_1$ and a literal $L_2$ *have an analogy* when all corresponding constants in the literals belong to each common sort. The correspondence of symbols is decided according to the syntactical positions of symbols.
(2) A *common sort* of two constants $c_1$ and $c_2$, is a sort $s$, if ground clauses $s(c_1) \leftarrow$ and $s(c_2)$

← are defined in the knowledge base, or clauses $s(c_1)$ ← and $s(c_2)$ ← are deduced from the knowledge base.

(3) An *analogy* between literals $L_1$ and $L_2$ is the correspondence of their constants $\{ (c_{11}, c_{21}), (c_{12}, c_{22}), \ldots \}$ where $c_{11}, c_{12}, \ldots$ are constants in the literal $L_1$ and $c_{21}, c_{22}, \ldots$ are constants in the literal $L_2$.

Example 1 illustrates an example of an analogy between the solar system and an atomic model described in (Gentner 1983).

**Example 1.** prediction of an atomic model in which an electron revolves around a nucleus.

**Target descriptions:**

← *revolves_around(electron, nucleus)*
*attracts(nucleus, electron)* ←
*more_massive_than(nucleus, electron)* ←

**Source descriptions:**

*revolves_around(P, S)* ←
*celestial_body(P), celestial_body(S),*
*attracts(S, P), more_massive_than(S, P)*
*attracts(sun, planet)* ←
*more_massive_than(sun, planet)* ←

**Taxonomic information:**

*physical_object(X)* ← *celestial_body(X)*
*physical_object(X)* ← *elementary_particle(X)*
*celestial_body(sun)* ←
*celestial_body(planet)* ←
*elementary_particle(electron)* ←
*elementary_particle(nucleus)* ←

**Analogy:**

*{ (nucleus, sun), (electron, planet) ,*
*(elementary_particle, celestial_body) }*

**Hypothesis:**

*revolves_around(P, S)* ←
*elementary_particle(P),*
*elementary_particle(S),*
*attracts(S, P), more_massive_than(S, P)*

In this example, the mapping *{ (nucleus, sun), (electron, planet) , (elementary_particle, celestial_body) }* is a recognized analogy. Constant terms *sun* and *planet* belong to a sort *celestial_body* as well as *nucleus* and *electron* belong to a sort *elementary_particle*. The sorts *celestial_body* and *elementary_particle* are subsorts of a sort *physical_object*. The target goal ← *revolves_around(electron, nucleus)* is deduced from the hypothesis generated with this mapping and by substituting *celestial_body* with *elementary_particle* in the source concept-description.

## The algorithm of IPA

The problem addressed by the Inductive Prediction by Analogy is formalized as follows:

**Given:**

- a target goal, which is a ground instance of a target concept
- examples of the target concept including the target goal
- background knowledge

**Find:** a hypothetical concept-description of the target concept such that the hypothesis covers all positive examples and no negative examples.

To solve the problem, IPA technique utilizes the process of analogical reasoning consisting of four main steps, *recognition*, *elaboration*, *evaluation*, and *consolidation* (Hall 1989):

(1) *recognition* of a candidate analogous source from a given target goal,
(2) *elaboration* of an analogical mapping between source and target domains, possibly including a set of analogical inferences,
(3) *evaluation* of the mapping and inferences in some context of use, including justification, repair, or extension of the mapping,
(4) and *consolidation* of the outcome of the analogy so that its results can be usefully reinstated in other context.

The steps of the algorithm of IPA technique is as follows:

**Step 1: Recognition.** Find a source ground clause which is similar to the target goal and search a source concept-description which covers the source ground clause. The recognized similarity gives a part of an analogy between the target and the source. If there exist multiple source ground clauses similar to the target goal, then IPA selects a candidate in the order of clause descriptions.

**Step 2: Elaboration.** First, transform the source concept-description using the analogy obtained in Step 1. By replacing each constant in the source concept-description with the corresponding constant in the analogy. Second, variablize all constants except for the replaced terms in the body of the transformed concept-description, and instantiate the variablized clause with the target goal and the knowledge base to get detailed analogy between the target and the source. Finally, transform the source concept-description using the detailed analogy to get a target concept-description.

**Step 3: Evaluation.** Add the target concept-description to the knowledge base and prove the target goal. If the target goal can be proved, then let the target concept-description be a candidate. If not, retract the candidate from the knowledge base, then backtrack to Step 2. If the candidate covers all the positive examples and no negative examples, then let the candidate be a hypothesis. If such a candidate cannot be found, then backtrack to Step 1.

**Step 4: Consolidation.** Let the generated hypothesis be a concept-description of the target concept.

If the Herbrand base of a target domain is finite then the numbers of substitution for constant symbols is also finite and an analogical mapping is clearly decidable. This implies the algorithm of IPA terminates.

An illustration of the process of the algorithm of IPA is shown in Example 2.

**Example 2.** Decide whether a given ground atom *family(mother, mary, kate)* is valid or not, when no concept-descriptions for the goal are available in a knowledge base.

**Given:**

- a target goal ← *family(mother, mary, kate)*, which is a ground instance of a target concept *family(mother, X, Y)*
- an example *family(mother, lucy, sara)* ← of the target concept
- background knowledge including taxonomic information for constant terms

**Find:** a hypothetical concept-description of the target concept such that the hypothesis covers all positive examples and no negative examples.

In this example we assume that the following source concept-descriptions and background knowledge are defined in the knowledge base.

**Target descriptions:**

← *family(mother, mary, kate)*
*sex(female, mary)* ←
*parent(mary, kate)* ←
*sex(female, lucy)* ←
*parent(lucy, sara)* ←

**Example (positive):**

*mother(lucy, sara)* ←

**Source descriptions:**

*family(father, X, Y)* ←
  *sex(male, X),parent(X, Y)*
*family(father, john, tom)* ←
*sex(male, john)* ←
*parent(john, tom)* ←
*family(sister, X, Y)* ←
  *sex(female, X),sibling(X, Y)*
*family(sister, ann, sara)* ←
*sex(female, ann)* ←
*sibling(ann, sara)* ←

**Taxonomic information:**

*family(father)* ←
*family(mother)* ←
*family(sister)* ←
*sex(male)* ←
*sex(female)* ←

From the knowledge base above the algorithm of IPA generates a hypothesis *family(mother, X, Y)* ← *sex(female, X),parent(X, Y)* in the following way:

**Step 1: Recognition.** Find a source ground clause which is similar to the target goal ← *family(mother, mary, kate)*. There are two ground clauses, *family(father, john, tom)* ← and *family(sister, ann, sara)* ←, similar to the target goal, because analogies *{(mother, father)}* and *{(mother, sister)}* can be recognized using taxonomic information. The predicate *family* of arity one defines that the constants *father* and *mother* belong to the common sort *family*, and so on. The algorithm of IPA selects *family(father, john, tom)* ← as a candidate for the source ground clause according to the order of clause descriptions. Then search a source concept-description *family(father, X, Y)* ← *sex(male, X),parent(X, Y)*, which covers the source ground clause.

**Step 2: Elaboration.** First, transform the source concept-description using the analogy obtained in Step 1. By replacing the constant *father* in the source concept-description with the corresponding constant *mother* in the analogy. Second, variablize the constant *male* in the body of the transformed concept-description except for the replaced terms, and instantiate the variablized clause with the target goal and the knowledge base to get detailed analogy between the target and the source. Then the following target concept-description is generated: *family(mother, X, Y)* ← *sex(female, X),parent(X, Y)*.

**Step 3: Evaluation.** Add the target concept-description to the knowledge base and prove the target goal. As the target goal can be proved, then let the target concept-description be a candidate. As the candidate covers all the positive examples and no negative examples, then let the candidate be a hypothesis. Negative examples of a concept *family(mother, X, Y)* are all ground instances that satisfy *family(father, X, Y)* and *family(sister, X, Y)*.

**Step 4: Consolidation.** Let the generated hypothesis be a concept-description of the target concept.

## An experiment in molecular biology

In order to demonstrate the usefulness of IPA technique we apply it to the function prediction of proteins from amino acid sequences. An extremely important task in molecular biology is to predict the functions of a protein given its amino acid sequence (Shavlik et al. 1995) (Hunter 1993) (Schulz & Schirmer 1979). The problem of this experiment is to predict the functions of a protein having an amino acid sequence similar to that of bacteriorhodopsin (Henderson et al. 1990) in the SWISS-PROT protein sequence database (Bairoch & Boechmann 1994). Our experiment shows that the Inductive Prediction by Analogy technique is useful at least for domains with many structurally related predicates.

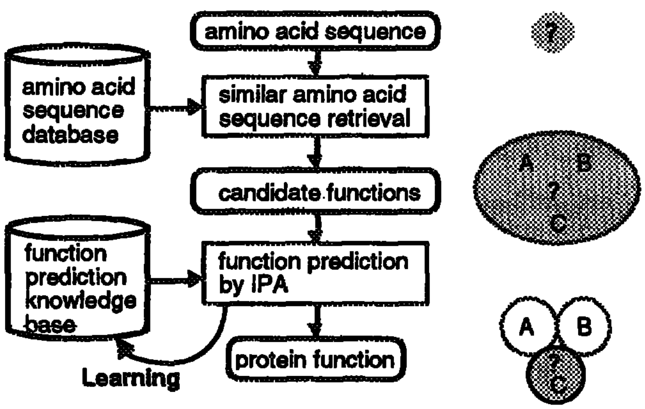


Figure 1. An overview of the system for function prediction of proteins

**System.** The system for the function prediction of proteins as shown in Figure 1 inputs an amino acid sequence of a protein with unknown functions and outputs a function of the protein. The system comprises two processes: *similar amino acid sequence retrieval* and *function prediction by IPA*. The similar amino acid sequence retrieval finds proteins having similar amino acid sequences with the inputted amino acid sequence from the amino acid sequence database. The system outputs a candidate function of retrieved proteins when their class falls into one function class. If the function classes are more than one class, then the system executes the next function prediction by IPA to refine function prediction of the protein. Since it is not sure that the target protein has the function same as the function of the protein having the retrieved amino acid sequences, the function prediction by IPA refines candidate protein functions to a specific protein function. The process generates concept-descriptions for the candidate functions, and apply them as classification rules to determine a specific protein function with the knowledge base. In the process, the system learns new knowledge and extends the knowledge base.

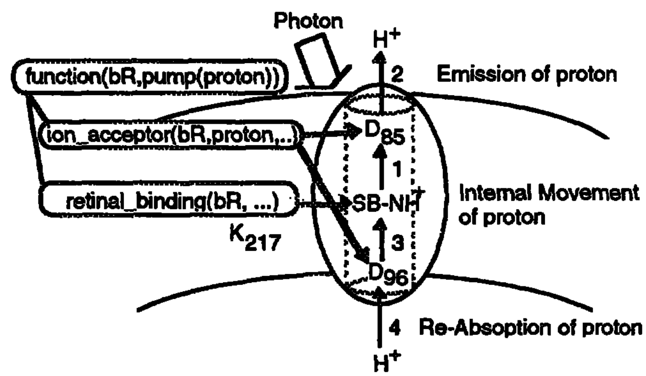


Figure 2. An abstract model of bacteriorhodopsin

**Implementation.** We have implemented a Prolog program based on the proposed algorithm of IPA. We have applied the program to the function prediction of proteins having amino acid sequences similar to bacteriorhodopsin (abbreviated as bR). bR is one of a few proteins whose structures and functions are well studied compared to other proteins. That is, we know its amino acid sequence, tertiary structure, and biological functions (Henderson et al. 1990). bR is a trans-membrane protein and has a function of proton pump that transports proton ($H^+$) across the membrane of cells. It is known that the structure of bR has seven alpha-helices and retinal as working material in it as shown in Figure 2. There are several proteins with amino acid sequences similar to that of bR in the SWISS-PROT database. Conventional method for the function prediction of proteins such as PROSITE (Bairoch & Boechmann 1994), however cannot discriminate these functions. Also bR is only one protein whose concept-description for its protein function can be given. In this experiment, the target protein with an amino acid sequence similar to bR is halorhodopsin (abbreviated as hR) chosen from these proteins. hR transports chloride ion ($Cl^-$) instead of proton ($H^+$) in bR as an ion pump. Here we assume that we only know that the function of hR is one of functions of retrieved proteins including chloride pump. In the following, we will explain the process of IPA technique for the function prediction of hR. In the experiment, a hypothesis for the concept-description of chloride pump is generated by analogical reasoning from the concept-description of proton pump, and the function of hR is predicted by justifying the hypothesis.

**Problem.** The problem of the function prediction of hR is defined using Prolog descriptions as follows:

**Given:**

- a target goal `function(hR,pump(chloride))`
- examples of the target concept including the target goal
- background knowledge

**Find:** a hypothetical concept-description of the target concept such that the hypothesis covers all positive examples and no negative examples.

**Knowledge base.** First we describe the concept-description of proton pump in the knowledge base. The description has the head literal `function(X,pump(proton))` and the body literals including predicates `ion_acceptor` and `retinal_binding`. The predicate `ion_acceptor` represents that an amino acid with opposite charge of transporting `Ion` exists in `helix(I)` of the amino acid sequence of a protein `X`. The predicate `retinal_binding` represents that an amino acid `` `K` `` (retinal binding) exists in `helix(I)` of the amino acid sequence of a protein X. The terms `helix(1), ..., helix(7)` represent indices of alpha-helices as a secondary structure of proteins. A part of Prolog descriptions of the knowledge base for this experiment is shown below.

```
%% Prolog descriptions of proton pump
function(X,pump(proton)) :-
  ion_acceptor(X,proton,helix(3),Pos1),
  ion_acceptor(X,proton,helix(3),Pos2),
  Pos1<Pos2,
  retinal_binding(X,helix(7)).

ion_acceptor(X,Ion,helix(I),Pos) :-
  trans_mem_sequences(X,SQ),
  member(NSQ,SQ,I),
  charge(Ion,C),opposite(C,AC),
  charge(Res,AC),
  string_member(Res, NSQ, P),
  in_membrane(P),length(NSQ,L),
  (member(I,[1,3,5,7]) ->
    Pos = P ; Pos is L-P+1).

retinal_binding(X,helix(I)) :-
  trans_mem_sequences(X,SQ),
  member(NSQ,SQ,I),
  string_member(`K`,NSQ,P),
  in_membrane(P).

in_membrane(Pos) :- Pos>3,Pos<20.

%% positive examples
function(bR,pump(proton)).
function(aR,pump(proton)).
function(sR,sensor(light)).

%% amino acid sequences of seven alpha-
helices
trans_mem_sequences(bR,
  [`WIWLALGTALMGLGTLYFLV`,
  `AITTLVPAIAFTMYLSMLLG`,
  `RYADWLFTTPLLLLDLALLV`,
  `ILALVGADGIMIGTGLVGAL`,
  `FVWWAISTAAMLYILYVLFF`,
  `TVVLWSAYPVVWLIGSEGAG`,
  `ETLLFMVLDVSAKVGFGLIL`]).
trans_mem_sequences(hR,
  [`LLSSSLWVNVALAGIAILVFVYMG`,
  `WGATLMIPLVSISSYLGLLSGLTV`,
  `SQWGRYLTWALSTPMILLA`,
  `SLFTVIAADIGMCVTGLAAAMTTS`,
  `FRWAFYAISCAFFVVVLSALVTDWA`,
  `AEIFDTLRVLTVVLWLGYPIVWAV`,
  `VTSWAYSVLDVFAKYVFAFILLRW`]).
trans_mem_sequences(aR,
  [`LWLGIGTLLMLIGTFYFIVKGW`,
  `SITILVPGIASAAYLSMFFGIGLTEV`,
  `ADWLFTTPLLLLDLALLA`,
  `IGTLVGVDALMIVTGLVGAL`,
  `WLFSTICMIVVLYFLATSLRA`,
  `LTALVLVLWTAYPILWIIGT`,
  `LGIETLLFMVLDVTAKVGFGFILL`]).
trans_mem_sequences(sR,
  [`TAYLGGAVALIVGVAFVWLLYRS`,
  `SPHQSALAPLAIIPVFAGLSYVGMAY`,
  `GLRYIDWLVTTPILVGYVGYAA`,
  `IIGVMVADALMIAVGAGAVV`,
  `ALFGVSSIFHLSLFAYLYVIF`,
  `QIGLFNLLKNHIGLLWLAYPLVWLFGP`,
  `GVALTYVFLDVLAKVPYVYFFYARRR`]).

%% taxonomic information
protein(bR).
protein(hR).
protein(aR).
protein(sR).
ion(proton).
ion(chloride).
protein_function(pump(proton)).
```

```
protein_function(pump(chloride)).
protein_function(sensor(light)).
helix(1).
helix(2).
helix(3).
helix(4).
helix(5).
helix(6).
helix(7).
```

**Reasoning process.** The algorithm of IPA starts from the following goal literal representing that the function of hR is chloride pump:

```
Goal = function(hR,pump(chloride))
```

**Step 1: Recognition.** Find a ground source clause (analogue) `function(bR,pump(proton))` similar to the target goal `function(hR, pump(chloride))` by making an analogy between the analogue and the target goal using taxonomic information defined in the knowledge base, and search a source clause which covers the ground source clause.

```
Analogue = function(bR,pump(proton))
Analogy = [[hR,bR],[chloride,proton],
    [pump(chloride),pump(proton)]]
Source_clause =
function(X,pump(proton)) :-
  ion_acceptor(X,proton,helix(3),Pos1),
  ion_acceptor(X,proton,helix(3),Pos2),
  Pos1<Pos2,
  retinal_binding(X,helix(7)).
```

**Step 2: Elaboration.** Apply the analogy to generate a candidate hypothesis for the target concept-description. To obtain a valid concept-description of `function(X,pump(chloride))`, `helix(3)` in the source clause should be transformed by variablizing two constants '3' into different variables and instantiating the variables with the knowledge base. The IPA program generates the following hypothesis for the concept-description of chloride pump.

```
Target_clause =
function(X,pump(chloride)) :-
  ion_acceptor(X,chloride,helix(3),
    Pos1),
  ion_acceptor(X,chloride,helix(6),
    Pos2),
  Pos1<Pos2,
  retinal_binding(X,helix(7)).
```

**Step 3: Evaluation.** Next, the generated hypothesis for the concept-description of chloride pump is refined so that the functions of proteins having similar amino acid sequences can be classified correctly. The refinement is executed by backtracking Step 2 until the condition is satisfied if needed.

**Step 4: Consolidation.** By adding the obtained concept-description to the knowledge base, we succeed in deciding the function of hR as a chloride pump from its amino acid sequence.

**Result.** By applying same procedure above, IPA technique successfully classifies protein functions of all proteins similar to bR, which are aR (archrhodopsin), sR (sensory-rhodopsin), and hR (halorhodopsin) (Ishikawa et al. 1995). Table 1 summarizes the classification features of the bacteriorhodopsin-like proteins.

Table 1. Classification features of bacteriorhodopsin-like proteins

| protein | function | features |
|---|---|---|
| bR, aR | proton pump | two amino acids with negative charge in helix(3) |
| hR | chloride pump | one amino acid with positive charge in helix(3), one amino acid with positive charge in helix(6) |
| sR | not ion pump | no above features |

Instead of using expensive biological efforts, the system using IPA technique learns these classification features from amino acid sequences and back-ground knowledge including taxonomic information. Although the discovered features for hR and sR have not yet completely certified in molecular biology, a recent research strongly suggests that these results are probable to be valid (Futai 1991). This means that the proposed technique is of use to support novel scientific discovery from biological database.

## Discussion and related work

The proposed technique IPA is effective for generating classification rules from a very few number of training examples. Instead of applying the proposed technique using analogical reasoning to generate hypotheses of classification rules, it is difficult to generate the same classification rules by the direct applications of inductive inference techniques to amino acid sequences. Because the proposed technique generates functional models for protein functions in top down manner, so obtained rules have abstract structures easy to understand for domain experts.

The use of analogical reasoning prunes meaningless generations of hypotheses in generating hypotheses. Therefore, the proposed technique improves the efficiency of learning. IPA technique generates a valid hypothesis for a target literal simply by variablizing constant terms in the explanation of the target literal and by instantiating the explanation without searching for generalizations and specializations of the terms.

IPA technique is applicable to domains with structurally related predicates, specifically to predicates of same syntactic structures. This limitation of IPA technique may be overridden by introducing the notion of abstraction-based analogy (Greiner 1988) (Ishikawa & Terano 1993). It is one direction of future research and we are conducting experiments for the purpose.

The technique Constructive Induction by Analogy (De Raedt & Bruynooghe 1992) uses second order schema, which is first introduced by (Yokomori 1986), whereas IPA technique uses taxonomic information to find analogies. However, the algorithm in (De Raedt & Bruynooghe 1992) requires asking the clause question to the user, whereas IPA technique automatically constrains candidates using taxonomic information.

The concept-learning by analogy in (Tecuci 1993) uses analogy based on *determination rules*, which are introduced in (Davies & Russell 1987). Determination rules are higher order rules and give too strong information to transform source knowledge to target knowledge. In IPA technique, simple taxonomic information plays a role of mapping the source symbols to the target symbols.

A paper concerned with logic program synthesis from examples (Sadohara & Haraguchi 1995) uses abstraction-based analogy (Greiner 1988) for explanation structures of logic programs. However, the algorithm is impractical since the enumeration of analogical mappings is computationaly explosive. Another paper concerned with analogical reasoning for logic programming (Tausend & Bell 1992) uses mechanisms of *Inductive Logic Programming* (Muggleton & Buntine 1988). However the input of the algorithm at least requires two analogical examples, whereas IPA technique automatically can find analogous examples from even only one example.

## Conclusion

This paper has presented a novel machine learning technique *Inductive Prediction by Analogy* (IPA), which learns the descriptions of a target predicate similar to a source predicate from a few examples of the target predicate. IPA technique allows the learner to use analogical reasoning in generating hypotheses using taxonomic information represented by first-order predicate logic. The usefulness of the technique has been validated by a real world problem in molecular biology: the function prediction of proteins from their amino acid sequences. From the experiment, we have observed that the proposed technique generates *interesting* biological hypotheses from protein database. In this paper, although we have focused on molecular biology, however, IPA technique is applicable in various domains. We have succeeded in solving such problems as discovery of the Pythagorean theorem and prediction of the Rutherford model of atom (Ishikawa & Terano 1994). Therefore, we believe IPA technique will provide a simple but strong way in knowledge system development.

## References


Bairoch, A and Boechmann, B. 1994. *Nucleic Acids Res.* 22:3578-3580.

Davies, T. R. and Russell, S. J. 1987. A logical approach to reasoning by analogy. In Proceedings of the International Joint Conference on Artificial Intelligence 1987, 264-270.

De Raedt, L. and Bruynooghe, M. 1992. Interactive concept-learning and constructive induction by analogy. *Machine Learning* 8:107-150.

Frisch, A. M. and Page Jr., C. D. 1990. Generalization with taxonomic information. In Proceedings of AAAI-90, 755-761.

Futai M. eds. 1991. *Bio-membrane Engineering* (in Japanese).:Maruzen.

Genesereth, M. and Nilsson, N. J. 1987. *Logical Foundations of Artificial Intelligence.*:Morgan Kaufmann.

Gentner, D. 1983. Structure mapping: A theoretical framework for analogy. *Cognitive*

*Science* 7:155-170.

Greiner, R. 1988. Learning by understanding analogies. *Artificial Intelligence* 35:81-125.

Hall, R. P. 1989. Computational approaches to analogical reasoning: A comparative analysis. *Artificial Intelligence* 39:39-120.

Henderson, R. et al. 1990. J. Mol. Biol. 213:899-929.

Hunter, L. ed. 1993. *Artificial Intelligence and Molecular Biology*.:AAAI Press.

Ishikawa, T. and Terano, T. 1993. Analogy by Abstraction: Theory of Case Retrieval and Adaptation in Inventive Design Problems. In Proceedings of AAAI-93 CBR workshop. also to appear in *Expert Systems with Applications* 10.

Ishikawa, T. and Terano, T. 1994. A Formulation of Analogy by Abstraction and Its Implementation with Logic Programming (in Japanese). In Proceedings of Knowledge Reformation Symposium 1994, 156-167.

Ishikawa, T., Mitaku, S., Terano, T., Hirokawa, T., Suwa, M., and Seah, B-C. 1995. Building a knowledge-base for protein function prediction using multistrategy learning. In Proceedings of Genome Informatics Workshop 1995, 39-48.

Kedar-Cabelli, S. T. 1985. Purpose Directed Analogy, In Proceedings of the Cognitive Science Society, Irvine, CA, August 1985.

Michalski, R.S. and Tecuci, G. eds. 1994. In *Machine Learning: A Multistrategy Approach Vol. IV* .:Morgan Kaufmann. 3-62.

Muggleton, S. and Buntine, W. 1988. Machine invention of first-order predicates by inverting resolution. In Fifth International Conference on Machine Learning:Morgan Kaufmann.

Sadohara, K. and Haraguchi, M. 1995. Analogical logic program synthesis from examples. In ECML-95, 232-244.

Schulz, G. E. and Schirmer, R. H. 1979.*Principles of Protein Structure* .:Springer-Verlag.

Shavlik, J., Hunter, L., and Searls, D. eds. 1995. Special Issues on Applications in Molecular Biology, *Machine Learning* 21.

Tausend, B. and Bell, S. 1992. Analogical reasoning for logic Programming. In *Inductive Logic Programming*.:Academic Press.

Tecuci, G. 1993. Plausible justification trees: A framework for deep and dynamic integration of learning strategy. *Machine Learning* 11:237-261.

Utgoff, P. E. and Mitchell, T. M. 1982. Acquisition of appropriate bias for concept learning. In Proceedings of Second National Conference on Artificial Intelligence, 414- 418.

Yokomori, T. 1986. Logic Program Forms. *New Generation Computing* 4:305-320.