ICOT Technical Memorandum: TM-0819

TM-0819

DNA Sequence Knowledge Base System (KNOA)

by
A. Konagaya & M. Yokota

October, 1989

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

DNA Sequence Knowledge Base System (KNOA)

Akihiko Konagaya and Minoru Yokota
C&C Systems Research Laboratories, NEC Corporation
4-1-1 Miyazaki Miyamae-ku, Kawasaki 213, Japan.

Abstract

Genetic information processing is focused on as a fruitful application area of logic programming in the FGCS project. A prototype DNA knowledge base system (KNOA) has been developed on an inference machine CHI. KNOA aims at providing an integrated system for genetic information processing in logic programming. DNA sequence data, protein sequence data and protein structure data have been already available with taxonomy information. Several homology search systems and a secondary structure inference verification system have been also developed experimentally. The effectiveness of stochastic inductive inference in homology search is emphasized.

1 Introduction

The Fifth Generation Computer Systems (FGCS) project has been pursuing high performance inference machines for knowledge information processing based on logic programming since 1982 [14, 13, 11]. As part of the project, we have developed an inference machine CHI [6] and also have proposed knowledge representation, inference mechanism and knowledge base construction methodologies based on logic programming [7]. To make good use of logic programming, we have developed a prototype DNA knowledge base system KNOA [8] on the inference machine CHI. In this paper, we will discuss the effectiveness of the logic programming and the inference machine architecture for genetic information processing.

The rest of this paper is as follows. At first, we will quickly review the genetics information processing, such as homology search and protein structure prediction in section 2. Next, we will discuss the advantages of logic programming approach for genetic information processing in section 3. Then, we will introduce the prototype knowledge base system KNOA in section 4.

2 Genetic Information Processing

2.1 DNA Sequence, Amino Acid Sequence and Proteins

Genetic information processing deals with issues concerning DNA sequences, amino acid sequences and proteins. A DNA sequence is a double-stranded giant molecule which consists of four kinds of nucleotides. It contains genes that produce proteins. A protein is obtained according to the following procedure. First, a messenger RNA is copied from a gene. Then, the messenger RNA is translated to an amino acid sequence by replacing every three nucleotides (codons) to one corresponding amino acid. Finally, the amino acid sequence starts folding and acts as a protein.

We especially focus on homology search and protein structure prediction from the viewpoint of knowledge information processing application, although computers are used in various ways in genetic information processing [9]. The homology search deals with ambiguous search for an unrecognized DNA sequence, and the protein structure prediction deals with structure prediction from a DNA sequence or an amino acid sequence.

2.2 Homology Search

Homology search is a kind of ambiguous search in the sense that it finds all sequences that are similar to, but not the same as, the target sequence in a databank. In ordinal homology search algorithms, a DNA sequences is considered as a sequence of characters, each of which represents an amino acid or a nucleotide. Similarity is often measured by the Hamilton length of two sequences, that is, the number of exactly matching elements. In the search, appropriate blanks (called "gaps") can be inserted to align sequences as seen in the example of Figure 1. In the example, more than half the elements can be aligned by inserting gaps, while only two elements are aligned without the gaps. In order to deal with the gaps, the dynamic programming (DP) matching algorithm and the hash-coding matching algorithm are commonly used [5].

One of the big issues in homology search is the size of databanks. Current DNA databanks, such as GenBank [2], have more than 20,000 sequences, in other words, more than 30,000,000 nucleotides; their size rapidly increases year by year. The sizes of databanks will greatly increase if the Human Genome project [1] starts. This implies homology search will take an unbearably long time unless remarkable performance improvements are made.

For this purpose, we proposed an alternative method based on stochastic inductive inference [12]. The advantages of this approaches will be discussed in sections 3 and 4.

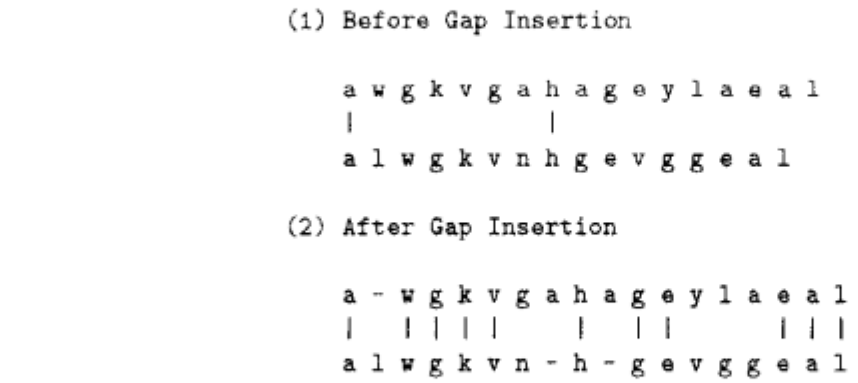

Figure 1: Gap Insertion Example

2.3 Protein structure prediction

Many attempts have been made to predict a protein structure from a given DNA sequence or an amino sequence. Roughly speaking, the methods are categorized into two ways: an energy minimization approach and an empirical inference approach.

The energy minimization approach tries to calculate a stable structure that minimizes the entire energy in the protein. This approach is very attractive since it may predict the exact molecular structure. However, it is often said that the required computation power is far beyond the ability of current computer systems for giant molecules like a protein.

Therefore, we are more interested in the latter approach, empirical inference. Empirical inference tries to predict a protein structure by applying rules obtained from the known three dimensional protein structures (tertiary structures). Currently, around 300 tertiary structures are known, and we can obtain the relations between the tertiary structures and the corresponding amino sequences (primary structures). Then, we can extract rules that map amino sequences to protein structures by analyzing the relations.

To reduce the complexity of protein structure prediction, "secondary structures" are proposed between primary structures and tertiary structures.

The secondary structure represents a characteristic part of a protein and is often categorized into an alpha helix (a helix structure), a beta sheet (a sheet structure), a turn (a turn structure) and others that connect the previous three structures.

We are very interested in this issue from the viewpoint of knowledge information processing application, such as knowledge representation, inference mechanism and learning.

3 Effectiveness of Logic Programming

Logic programming has good capabilities for genetic information processing from the following viewpoints: rapid prototyping and "logical inference". Most practical systems dedicated for genetic information processing are written in conventional procedural languages, such as Fortran and C. However, this does not mean the procedural languages suit genetic information processing, and much remains to satisfy biologist's requirements.

The problem is that little is known about genetic information processing models and new algorithms are always required to analyze more exact genetic models. In such cases, rapid prototyping is more appropriate than conventional software development methodologies, such as a water fall model, since feasibility of the algorithm is much more important than performance.

Logic programming language greatly enhances prototyping, because it provides high level programming facilities for matching, inference and database access, which are main operations of genetic information processing. The facilities enable us to concentrate our efforts mainly on exploiting new algorithms concerning genetic information processing and to keep away from data management and complex execution control. For example, we can easily develop a DNA databank using a clause database which provides advanced database facilities: data retrieval by matching (unification), automatic alternative search (backtracking), arbitrary length data structures (strings and lists), and relational definitions by predicate logic (Horn Clause).

The other important advantage of logic programming is that we can make use of "logical inference" for genetic information processing. For example, we are now interested in the application of stochastic inductive inference to find an assertion that distinguishes a unique superfamily from the other superfamilies. The stochastic inductive inference can deal with the probabilistic validity of assertions, that is, intermediate values between "true" and "false". This implies that we can make use of an assertion even if it does not completely satisfy all the sequences in a superfamily, and even if it satisfies sequences that do not belong to the superfamily.

In addition, the stochastic inductive inference has mathematical foundation of reasoning in contrast to the black-box reasoning such as back propagation learning in neuron network processing. This implies we can improve the result by analyzing the process of inference.

4 KNOA

KNOA is a prototype DNA knowledge base system developed on an inference machine which provides a fast Prolog processor (500 KLIPS) and a large scale main memory (320 MBytes). The advantage of KNOA design is in that all genetic information, such as DNA information and Protein information, are tightly integrated with expert systems, such as homology search,

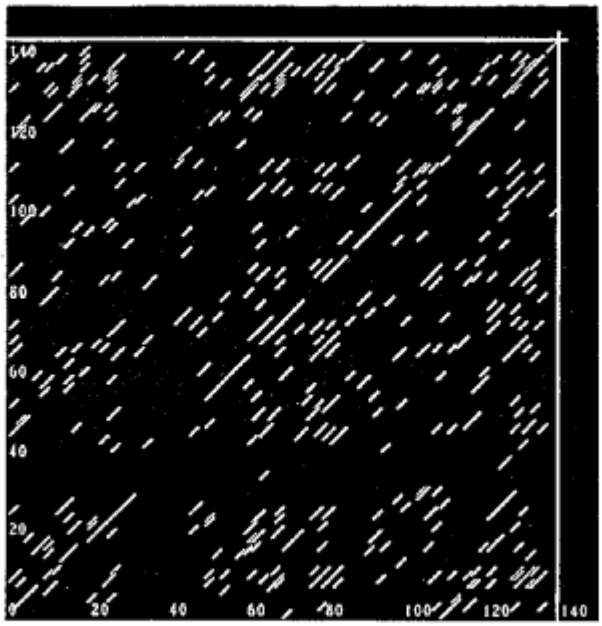

Figure 2: Dot Matrix Example (X-axis: α – *hemoglobin*, Y-axis: β – *hemoglobin*)

protein structure inference, and so on. Such integration greatly reduces the impedance mismatch of a database system and its application systems [10].

4.1 DataBank implementation

In KNOA, all DNA and protein databanks are implemented as collections of Prolog clauses, to make full use of clause database facilities. Currently, GenBank, NBRF, and Protein Structure Data Bank are available. The clause database provides a memory-based single-user relational database model with deductive facilities. The biggest advantage of logic programming is in its expandability of a clause database. Well-known logic programming methodology, called "meta programming" enables us to design any database model on a clause database by means of defining a predicate that simulates the database model. It greatly enhances database model design research for genetic information processing.

One of the interesting issues of DNA database development is to design a new database model that is more suitable to genetic information processing than a relational database model. Although a relational database model is powerful enough to cover most genetic information, it is too rigid to deal with hierarchical information like taxonomy and advanced user-interfaces such as graphical data and animation. KNOA is partly successful in solving this problem by providing a hierarchy of clause groups.

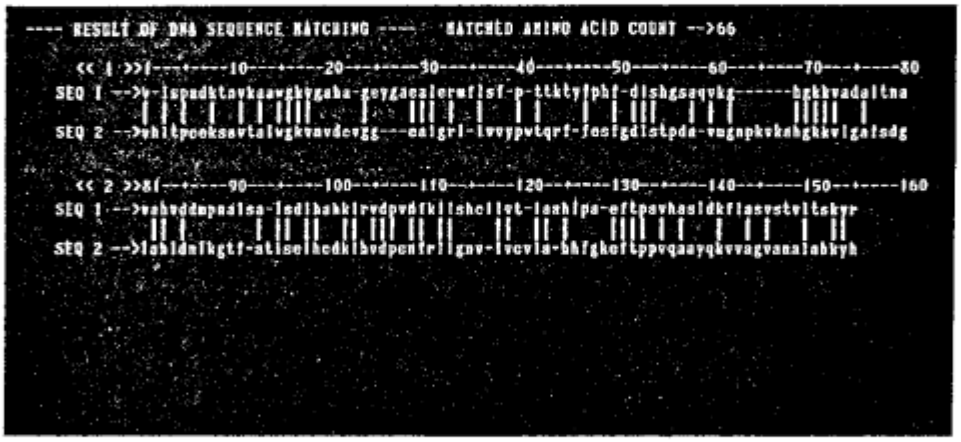

Figure 3: Alignment Example (Seq1: $\alpha-hemoglobin$, Seq2: $\beta-hemoglobin$)

4.2 Homology Search Systems

KNOA provides several homology search algorithms as well as graphical output tools (Figures 2 and 3). Well-known homology algorithms, such as DP matching and hash-coding matching, work well to some extent. However, they does not fully make use of genetic information, and require a long time to search the data bank.

To solve the problem, we have proposed a new homology search algorithm, called stochastic homology search. Stochastic homology search prob ablistically infers a specific superfamily according to the assertions obtained by the stochastic inductive inference. Appendix A shows examples of assertions for "cytochrome c". The assertions can be considered as a kind of "motif" that distinguishes "cytochrome c" from other superfamilies in the protein databank currently available.

The advantages of "stochastic homology search" are as follows. First of all, a great performance improvement can be obtained at the search time, since we can omit redundant matching against sequences in the same superfamily. In addition, we can apply the assertions for an arbitrary length of sequences, since it requires only small segments of a sequence to induce a superfamily.

The feasibility of the stochastic homology search as well as stochastic inductive inference is being evaluated. For such evaluation, the inference machine CHI is very effective, since we are almost free from the limitation of memory capacity, and a memory-based DNA databank greatly improves search performance. We believe inference machine CHI is one of the best

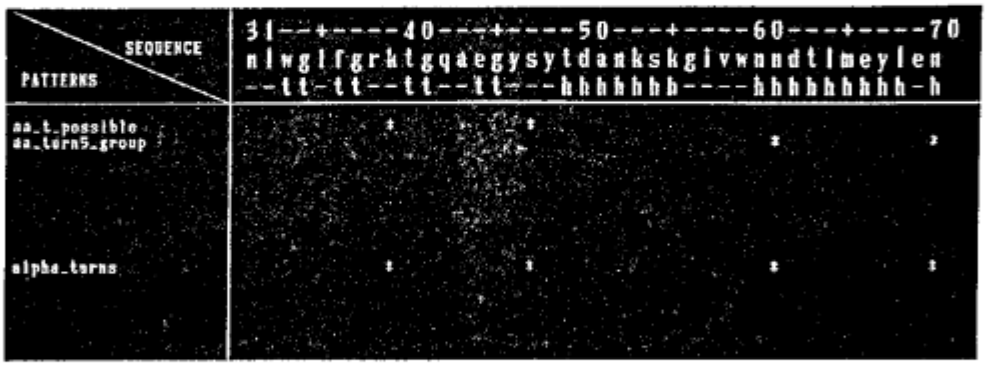

Figure 4: Secondary Structure Prediction Example (Cytochrome c of Tuna Heart 31 – 70)

machines to research genetic information processing.

4.3 Protein Structure Inference

Protein structure inference is one of the big issues in genetic information processing. As the first step of protein structure inference, an inference verification system has been developed. Figure 4 shows the result of secondary structure prediction for a cytochrome c protein of a tuna heart, using inference rules proposed by Cohen [4]. In the verification system, inference rules can be verified step by step with graphical representation of protein tertiary structures which have already known (See Figure 5). Such visualization greatly improves intuitive understanding of tertiary protein structures.

5 Conclusion

The advantages of the logic programming approach and inference machine architecture in genetic information processing is described. The expansibility and powerful language facilities for matching, inference and database access are very useful for advanced knowledge base system development. The large main memory capacity and high performance logic programming execution enable inference machine CHI to be a practical tool for this purpose. The effectiveness is especially proved in the application of stochastic inductive inference to homology search.

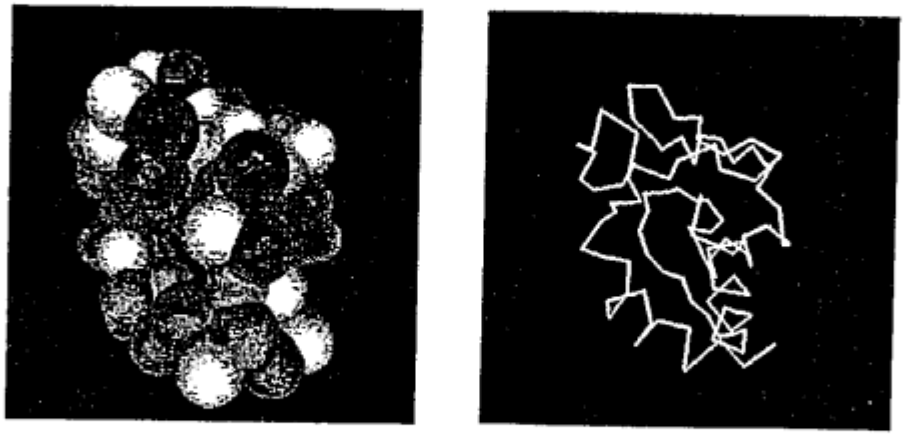

Figure 5: Tertiary Structure and Main Pass of Tuna Heart Cytochrome c

Acknowledgements We would like to thank Dr. Shun-ichi Uchida (ICOT) for his support on this project, and to Mr. Shin-ichi Habata, Mr. Atsushi Atarashi, and all project members who are engaged in research. Thanks are also expressed to Dr. Tsutomu Maruyama, Mr. Kouichi Konishi, Mr. Nobuhiko Koike and Dr. Tatsuo Ishiguro (NEC) for their continuous encouragement, valuable advice and support.

References

[1] "Mapping our Genes", Johns Hopkins Univ. Press

[2] Bilofsky, H.S., Burks, C., "The GenBank(R) Genetic Sequence Data Bank", Nucl. Acids Res., vol.16, 1987

[3] Bishop, M.J. and Rawlings, C.J. (Ed.), "Nucleic acid and protein sequence analysis - a practical approach ", IRL Press, 1987

[4] Cohen, F.E. et al, "Turn Prediction in Protein Using a Pattern-Matching Approach", American Chemical Society, vol.25, no.1, 1986

[5] Heijne, V.G., "Sequence Analysis in Molecular Biology", Academic Press Inc, 1987

[6] Habata, S., Nakazaki, R., Konagaya, A., Atarashi, A. and Umemura, M., "Co-operative High Performance Sequential Inference Machine: CHI", in *Proc. ICCD'87*, New York, 1987

[7] Ito, H., Monoi, H., Shibayama, S., Miyazaki, N., Yokota, H. and Konagaya, A., "Knowledge Base System in Logic Programming Paradigm", in *Proc. International Conference on Fifth Generation Computer Systems*, pp.37-53, Tokyo, Nov. 1988

[8] Konagaya, A. and M. Yokota, "DNA Knowledge Base System -KNOA-Logic Programming Approach", in IECE Japan AI SIG-AI 89-37, 1989 (in Japanese).

[9] Lesk, A.M. (Ed.). "Computational Molecular Biology", Oxford University Press, 1988

[10] Maier, D., Stein, J., Otis, A., Purdy, A., "Development of an Object-Oriented DBMS", in Proc. Object-Oriented Programming Systems, Languages and Applications, 1986

[11] Nakazaki, R., Konagaya, A., Habata, S., Shimazu, H., Umemura, M., Yamamoto, M., Yokota, M. and Chikayama, T., "Design of a High-speed Prolog Machine (HPM)", in *Proc. of the 12th Annual Int'l Symposium on Computer Architecture*, Boston, Jun 1985

[12] Ohno, K., "Stochastic Inductive Inference and Model Entropy for Hypothesis Evaluation", in Proc. 3rd Japanese Artificial Intelligence Conference, 1988 (in Japanese).

[13] Taki, K., "Performance and architectural evaluation of the PSI machine", in *Proc. of ASPLOS II*, 1987

[14] Uchida, S., "Inference Machine: From Sequential to Parallel", in *Proc. of Int. Symp. on Computer Architecture*, Jun 1983

Appendix Some of the assertions for cytochrome c:

```
% contain(Patterns,SuperFamilyName,Total,Success,Validity).
contain(['TKM','CHT'],'cytochrome c',69,68,0.9855073).
contain(['ANK','CHT'],'cytochrome c',61,59,0.9672131).
contain(['GTK','CHT'],'cytochrome c',71,68,0.9577465).
contain(['GPN','CHT'],'cytochrome c',70,67,0.9571428).
contain(['TKM','ANK'],'cytochrome c',67,64,0.9552239).
contain(['TKM','NPK'],'cytochrome c',73,69,0.9452055).
contain(['PNL','CHT'],'cytochrome c',71,67,0.943662).
contain(['AYL','CHT'],'cytochrome c',61,57,0.9344263).
contain(['NPK','GPN'],'cytochrome c',73,67,0.9178082).
contain(['TKM','PGT'],'cytochrome c',83,76,0.9156627).
contain(['PGT','CHT'],'cytochrome c',75,68,0.9066667).
contain(['GPN','ANK'],'cytochrome c',67,60,0.8955224).
contain(['TKM','GTK'],'cytochrome c',85,76,0.8941177).
contain(['NPK','CHT'],'cytochrome c',75,66,0.88).
contain(['IPG','CHT'],'cytochrome c',75,65,0.8666667).
contain(['GTK','NPK'],'cytochrome c',81,70,0.8641976).
contain(['TKM','PNL'],'cytochrome c',80,69,0.8625).
contain(['TKM','GPN'],'cytochrome c',80,69,0.8625).
contain(['TKM','IPG'],'cytochrome c',79,68,0.8607595).
contain(['PNL','ANK'],'cytochrome c',73,61,0.8356164).
contain(['NPK','AYL'],'cytochrome c',70,58,0.8285714).
contain(['GTK','ANK'],'cytochrome c',78,64,0.8205128).
contain(['TKM','AYL'],'cytochrome c',77,63,0.8181818).
contain(['GTK','GPN'],'cytochrome c',84,68,0.8095238).
contain(['PNL','IPG'],'cytochrome c',82,66,0.8048781).
contain(['IPG','GPN'],'cytochrome c',82,66,0.8048781).
contain(['PGT','NPK'],'cytochrome c',87,69,0.7931035).
contain(['AYL','GPN'],'cytochrome c',77,60,0.7792208).
contain(['NPK','PNL'],'cytochrome c',86,67,0.7790698).
contain(['GTK','IPG'],'cytochrome c',88,68,0.7727273).
contain(['PGT','ANK'],'cytochrome c',83,64,0.7710843).
contain(['AYL','ANK'],'cytochrome c',72,55,0.7638889).
contain(['GTK','AYL'],'cytochrome c',80,61,0.7625).
contain(['PGT','GPN'],'cytochrome c',91,68,0.7472528).
contain(['NPK','ANK'],'cytochrome c',82,61,0.7439024).
contain(['AYL','IPG'],'cytochrome c',79,58,0.7341772).
```