Finding Association Rules on Heterogeneous Genome Data

K. Satou, G. Shibayama, T. Ono†, Y. Yamamura†,
E. Furuichi‡, S. Kuhara†, and T. Takagi

Human Genome Center, Institute of Medical Science, The University of Tokyo, 4-6-1 Shiroganedai, Minato-ku, Tokyo 108, Japan.

†*Graduate School of Genetic Resources Technology, Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka 812, Japan.*

‡*Fukuoka Women's Junior College, 4-16-1 Gojo, Dazaifu, Fukuoka 818-01, Japan.*

A novel approach for discovery of knowledge from genome data, which has been recently watched with interest in the research area of database, is applied to finding unified rules spreading over sequence, structure, and function of protein. As the result of experiments using data extracted from PDB, SWISS-PROT, and PROSITE, some association rules stating sequential/structural/functional aspects of two kinds of endopeptidases were found.

1 Introduction

Intelligent management of huge amounts and heterogeneous genome data is one of the important research areas in bioinformatics. Advanced technology in AI and database are highly needed to achieve the goal, that is, providing molecular biologists with a powerful and comfortable genome analysis environment. In this research area, we studied the application of deductive database technologies to the analysis of protein structural data, and developed a database system named PACADE.[13,22] One of the results of the study was that a structural similarity search mechanism was devised and implemented on PACADE.[21] Through the similarity search function, a user can retrieve protein substructures similar to a substructure specified by the user from a viewpoint of distance, angle, and length of secondary structures.[20]

As a step to go further beyond database search which inevitably requires a user to describe query and a set of rules, we planned to study an approach which is reversal in the sense that a system generates rules from data. This direction of research meets an emerging research area in computer science, which is called *Data mining (Knowledge Discovery) from Databases* and aiming at the extraction of knowledge from a large amount of data in databases. As to genome data, *Inductive Logic Programming (ILP)* is actively applied from this

area. According to results reported by Muggleton et al., ILP seems promising, especially in some kind of prediction problems.[14] However, in case the target concept to be learned is not well-defined, another type of machine learning should be used because we can not give positive and/or negative examples.

An attractive problem of data mining called "Discovery of *association rules*" was first posed in 1993 by Agrawal et al.[1] The simplicity of its basic framework, robustness against a large amount of data, and intuitive readability on the semantics of the resulting association rules imply its potential applicability to many real-world problems. Therefore, it has been actively studied in both of the theory and application sides. Nevertheless, there are still few application studies on extraction of association rules from genome data in spite of its suitability for this research area. In this study, we tried to discover association rules concerned with the amino acid sequence, structure, and function of protein from heterogeneous genome database. As a result of these experiments using public genome database concerned with protein, some association rules stating sequential/structural/functional aspects were found.

2 Associating Large and Various Genome Information

As is commonly known, three kinds of information about protein, that is, sequence, structure, and function, are said to be closely related. However, though discovery of unified rules through this information has been actively studied for a long time, it has been solved only partly up to now because it is a large and complicated problem. Prediction of protein structure can be regarded as verification of such rules as a working hypothesis. Since establishing prediction methodology with high precision means not only knowing the principles of protein folding but also designing new proteins with a desired function, there have been many studies on protein structure prediction with various approaches. For example, molecular dynamics, 3D-1D,[6] threading, AI approaches including neural net,[18] decision tree over regular pattern,[24] etc. However, in spite of these attempts, even precision of secondary structure prediction is still not at a satisfactory level. In addition to rules associating sequence with structure, it is more important to find rules related to the function of protein. As to prediction of function from sequence, motif finding using homology search, multiple alignment, and machine learning techniques is one of the major approaches.[11] However, since the definition of protein function is obscure, there is still room for trying a new approach.

When an attempt to use such a prediction comes up against a brick wall, there may be important but unnoticed viewpoints spreading over various levels of information. Therefore, a knowledge discovery system which can point out

unnoticed association buried under large and various genome data is needed to make a breakthrough. Moreover, it also needed to pigeonhole and grasp large and complicated problems like correlation among sequence, structure, and the function of protein. These are the motivations of this study in which a data mining approach, recently watched with interest, is applied to finding association rules concerned with sequence, structure, and function of protein from heterogeneous genome database.

3 Association Rules

3.1 Terminology

The basic framework for discovery of association rules was first shown in the context of customer transaction database gathered by retail sellers.[1] Suppose that the data about a set of items are stored in a database as follows:

trans_id	`bread`	`butter`	`rice`	`milk`	`soy sauce`
1	1	1	0	1	0
2	0	1	0	0	1
3	1	0	0	0	1
4	1	1	0	1	1
5	1	1	1	0	0

In this database, binary data for five items mean whether they were bought or not, in a transaction trans_id. From such a database, sets of items called *large itemsets*, whose elements tend to be frequently bought together, are retrieved in the first step. There is a user-defined threshold value called *minimum support* on the frequency of retrieved large itemsets. For example, {`bread, butter, milk`} is retrieved under the condition `minimum support = 1` because its support is 2. Then in the second step, large itemsets are processed into association rules which have head and body in both sides of the implication. In this step, another threshold value called *minimum confidence* is used to delete inaccurate rules. This value is computed by dividing the support of the head (= frequency of head item) by the support of the body itemset. For example, an association rule `bread, butter` $\Rightarrow$ `milk` survives under the condition `minimum confidence = 60%` because it is 66.6% confident.

3.2 Related Works

The research topics of association rules can be roughly classified into the following 4 types:

[Performance improvement] From the beginning, it has been pointed out that the naive algorithm for finding large itemsets is inefficient.[1] Houtsma et al. mapped the problem of finding large itemsets into operation of relational database through SQL queries and named the algorithm SETM.[10] Agrawal et al. proposed an excellent pruning method named Apriori algorithm for eliminating useless candidate itemsets which must not be large itemsets. On the other hand, Agrawal et al. and Shintani et al. separately proposed parallel algorithms and reported results on a shared-nothing parallel computer.[2,25]

[Postprocessing of association rules] There exist apparently redundant or insignificant association rules from some viewpoints, i.e. statistic or user-defined. Corresponding to the terminology for "data mining", elimination of such rules can be called "refinement", which has been actively studied.[12,7,26] Visualization of resulting rules is another type of postprocessing having the same objective.[12,7]

[Application] Since the basic framework is not dependent on the initial application problem, it has a wide applicability in nature. An example is analysis of telecommunication network alarm databases.[8] Another example is an application to the retrieval of closely-related pages from WWW data resources,[16] where the weighted association rule is used for associating keywords. Concerning genome data, we tried to discover association rules from signals in mammalian promoter sequences.[23]

[Semantic extension of association rules] The semantics of the association rule `bread, butter` $\Rightarrow$ `milk` is limited to a propositional one. To make association rules richer, efforts to extend its semantics are needed. Srikant et al. tackled the problem of handling (interval of) numerical value, and proposed a sophisticated framework for finding quantitative association rules including numerical values.[26]

3.3 Comparison with ILP

Obviously, discovery of association rules is not a conflicting one with ILP, rather a complementing one because the latter is classified as learning from examples, and the former needs no examples. An ILP system like GOLEM [14] and FOIL [17] generates a logic program with first-order syntax and semantics, which is not achieved in association rules.

On the other hand, the framework for finding association rules has some good points:

- The cost of computation is lower comparing with ILP. This means a robustness against a large amount of data.
- The value of association rules can be estimated in intuitive measures.

3.4 Heterogeneous Data

In the example shown in subsection 3.1, there were no explicit classifications among the items and they were homogeneous in a sense. However, the simplicity of the basic framework of discovering association rules allows heterogeneous sets of items to occur in the attributes. Figure 1 illustrates our aim in this study:

protein name	`sequence feature1`	`sequence feature2`	`structure feature1`	`function1`	`function2`
name1	1	0	1	0	1
name2	0	0	1	1	0
name3	1	0	0	1	0
name4	1	0	1	1	1
name5	1	1	1	0	0

$\Downarrow$ Data Mining

`sequence feature1, structure feature1` $\Rightarrow$ `function2`
`(support=2, confidence=66.6%)`

Figure 1: Sketch of Data Mining on Heterogeneous Genome Data.

4 Methods and Materials

4.1 Data for Mining

Before performing the data mining illustrated in figure 1, proteins have to be characterized from sequential, structural, and functional viewpoints. In this study, we chose the following four data sources, which are related to each other by using PDB [5] entry names as keys:

[Sequential feature] PROSITE[4] is a dictionary of protein sites and patterns. If a PROSITE pattern matches to a SWISS-PROT[3] entry, there is a description of the pattern in the entry.

[Similar substructures as a structural feature] Using PDB as data source, PACADE can perform some kinds of structural similarity searches over the data of secondary structures. The similarity searches are roughly classified based on the following two points:

Rule set
If a user is interested in a specific structure, e.g. a supersecondary struc-

ture like Greek key, jelly roll, and meander, he can use a rule set for the structure. If he does not want to stick to any specific structure, he can use a rule set for any structure that consists of continuous secondary structures including α-helices, β-strands, and/or random coils.

In such a rule set, there are some parameters defining structural similarity, which are related to distance, angle, and length of α-helix and β-strand. By tuning these parameters, a user can specify his preference about error ranges of similarity. In this study, we set 60 degrees for angle, 6 angstrom for distance, and 15,40,50 residues for length of β-strands, α-helices, and random coils, respectively.

Direct or indirect similarity
PACADE can compute indirect similarity relationships as well as direct ones.[19] In the former case, starting from a specific substructure in a specific protein, the system iteratively performs a direct similarity search until no indirectly similar structures are found.

In this study, first we computed all the directly similar substructures that consisted of continuous three α-helices or β-strands allowing random coils between two of them (we call such a substructure *3-stranded*). An example of the answers which PACADE returned is below.

```
structure(2,6,"1lya") is similar to structure(19,23,"1aaj")
structure(2,6,"1npc") is similar to structure(19,23,"1aaj")
structure(2,6,"2cdv") is similar to structure(16,20,"9wga")
                              ⋮
```

The first and the second argument of the predicate `structure` represent the sequential numbers of starting and ending secondary structures in the protein represented by the third argument. Then, the 3-stranded substructures in the answers are mapped to integers.

```
1772 is similar to 1622
1779 is similar to 1622
1818 is similar to 1306
          ⋮
```

After that, 3-stranded substructures are classified into closures of indirectly similar ones. Sequential numbers are also assigned to the closures.

closure	sequential number of it
{1622, 1772, 1779, ...}	21
{1306, 1818, ...}	23
⋮	⋮

Finally, from these data, we generated the following data, each of which states "some of substructures in a closure *num* occur somewhere in a protein *pdbcode*".

```
1aaj  21
1lya  21
1npc  21
2cdv  23
9wga  23
 ⋮     ⋮
```

To keep the database consistent, we eliminated the entries such as the entries of DNA/RNA, the entries which have no SEQRES records, and the entries whose ATOM records have no data of residues. As a result of this filtering, 187 PDB entries as source data were selected and used to generate the above data for mining. Moreover, Kabsh and Sander's method was used to make up for the mistakes and inconsistencies in secondary structure data in PDB. We think these filtering and refining for PDB data are enough to keep the quality of the structural data for mining.

The reason why we did not take already published protein structure sets (SCOP;[15] FSSP[9]) is as follows:

- These structure sets only states structural similarity from a viewpoint of the whole protein or the whole chain of amino acid. Therefore, if these structure sets are used instead of search results of PACADE, the data mining system can not discover any associations concerned about characteristic substructures buried under the whole structure of protein.

[EC number as function] Enzymes are classified, based on their functions, in four levels of hierarchy, which are represented as EC numbers. SWISS-PROT entries for enzymes have descriptions of their EC numbers. To characterize proteins, we used part of the EC number in two ways, that is, the number of the 1st-2nd level and the number of the 1st-2nd-3rd level.

[SWISS-PROT keyword as function] Keywords in a SWISS-PROT entry include functional, structural, or other categorical characterization.

Part of the above four kinds of data could not be related to each other because of key mismatch. For instance, a protein `1aec` in PDB had no corresponding entry in SWISS-PROT. Starting from the 187 PDB entries, 181 out of the 187 had corresponding SWISS-PROT entries; 137 had matching PROSITE motifs; and 114 were enzymes with EC numbers. Of course, the 181 proteins had some keywords in corresponding SWISS-PROT entries. The following table illustrates these data assembled for mining.

pdb code	{1187,...,699}	SPPR=UBIQUITIN_CONJUGAT	EC3= 6.3.2	EC2= 4.2	SPKW= SIGNAL	···
1aaj	0	0	1	0	1	···
1aak	1	1	0	0	0	···
1abe	1	0	0	0	0	···
···	···	···	···	···	···	···

In this table, SPPR means PROSITE motif, and SPKW means keyword. These were extracted from SWISS-PROT entries. {1187,...,699} is an abbreviation of a set of 39 indirectly similar substructures including two substructures, 1187 and 699.

4.2 System

In this study, we adopted SETM algorithm and implemented it as Perl scripts and Sybase RDBMS on Sun 690MP. Only two threshold values, that is, minimum support (=5) and minimum confidence (=65%) were used. Since we did not use maximum support, some inconvenience about generation of a large amount of useless association rules was conjectured. For example, SPKW=3D-STRUCTURE marks too high support (=181) because this keyword means that there is a corresponding entry in PDB. Instead of using maximum support, we postprocessed the resulting association rules to delete the rules which include items of too high supports.

Besides items of high supports, there was one more conjectured inconvenience. Since the items EC2 and EC3 are apparently related to each other, useless rules such as EC3=1.2.3 => EC2=1.2 will be generated. Such rules were also deleted in the postprocessing phase.

5 Experimental Results

As the result of the mining, 182388 association rules were generated. After the postprocessing phase, it was decreased to 586 when the maximum support was set to 50. In the case of a maximum support 30, it was furthermore decreased to 381. The following 2 long rules are examples out of the 381 rules.

```
{1161,...,865}, SPKW=SERINE PROTEASE, SPKW=ZYMOGEN,
SPPR=TRYPSIN_HIS, SPPR=TRYPSIN_SER => EC3=3.4.21 (100%, 6 sp.)

{4356,...,808}, SPKW=ZYMOGEN, SPKW=ASPARTYL PROTEASE,
SPPR=ASP_PROTEASE => EC3=3.4.23 (100%, 6 sp.)
```

In this section, we analyze and discuss focusing on the 381 rules.

5.1 Classification based on EC Numbers

234 out of the 381 rules included a description of EC number. The following two tables show the breakdown.

EC number	the number of rules
EC2=1.1	2
EC2=2.1	1
EC2=2.7	1
EC2=3.1	5
EC2=3.2	2

EC number	the number of rules
EC3=1.1.1	4
EC3=3.2.1	2
EC3=3.4.21	171
EC3=3.4.23	46

We can say that most of such rules are concerned with only two kinds of EC numbers, that is, `EC3=3.4.21` and `EC3=3.4.23`. Other rules were not interesting since only the EC numbers and a few SPKW occurred in them like the following example rules.

```
SPKW=GLYCOSIDASE => EC2=3.2 (83%, 10 sp.)
SPKW=NUCLEASE,SPKW=ENDONUCLEASE => EC2=3.1 (85%, 6 sp.)
```

5.2 Serine Endopeptidases and Aspartic Endopeptidases

Rules related to `EC3=3.4.21` and `EC3=3.4.23` represent associations in serine endopeptidases and aspartic endopeptidases, respectively. These 171+46 rules consisted of items in the following tables.

items related to serine endopeptidases	sp.
{1161,...,865}	26
{1299,...,4355}	6
SPPR=TRYPSIN_HIS	9
SPPR=TRYPSIN_SER	9
EC3=3.4.21	13
SPKW=SERINE PROTEASE	13
SPKW=ZYMOGEN	21

items related to aspartic endopeptidases	sp.
{4356,...,808}	7
SPPR=ASP_PROTEASE	7
EC3=3.4.23	7
SPKW=ASPARTYL PROTEASE	7
SPKW=ZYMOGEN	21

Only the item `SPKW=ZYMOGEN` occurred in both of the two kinds of rules. The occurrence of other items in the rules agree with known biological knowledge.

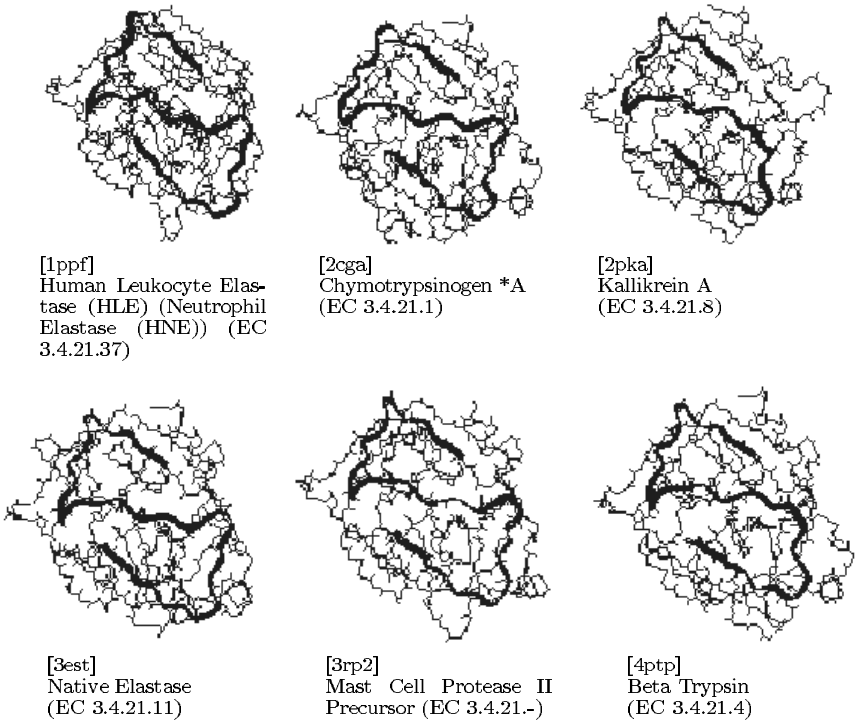

Figure 2: Graphical display of six serine endopeptidases.

Figure 2 shows the six serine endopeptidases which have similar substructures included in {1299,...,4355}, where the substructures are emphasized as dark and wide ribbons. Even if the substructure seems to be unrelated to the active site of these enzymes, it illustrates that the data mining method can be used to find substructures common to proteins which have the same function.

5.3 Discussion

There may be two reasons why almost all rules were concerned with serine and aspartic endopeptidases.

- Since these two enzyme families are among the most frequently found ones in the 187 proteins used here, this result may be biased.

- It is known that, according to experimental results on structural classification of proteins, these two enzyme families have distinctive structures compared with other proteins.

However, as to structure-structure association, there should be more association rules unrelated to serine and aspartic endopeptidases. We think that this result is caused by the loose definition of structural similarity mentioned in 4.1. Loose definition makes the resolution of classification low, which forces some of clusters to merge together. Since such a set of substructures with too high support are eliminated in the postprocessing phase, it does not occur in the resulting 381 rules focused in this section.

6 Concluding Remarks

In this study, we tried to find hidden association rules buried under large and various genome data. The experimental results were encouraging, but some problems came to the surface. For instance, background knowledge about propositionalization of source data should be used to eliminate uninteresting rules like `EC3=1.2.3 => EC2=1.2`. Moreover, visualization must also be needed to grasp a huge amount of association rules when we extend this experiment to more and various genome data.

Acknowledgments

This work was supported in part by a Grant-in-Aid for Scientific Research on Priority Areas, "Genome Science," from the Ministry of Education, Science, Sports and Culture in Japan.

References

1. R. Agrawal, T. Imielinski, and A.N. Swami: *ACM SIGMOD*, pp.207-216 (1993).
2. R. Agrawal and J.C. Shafer: *IBM Research Report RJ 10004* (1996).
3. A. Bairoch and R. Apweiler: *NAR*, Vol.24, pp.21-25 (1996).
4. A. Bairoch, P. Bucher, and K. Hofmann: *NAR*, Vol.24, pp.189-196 (1995).
5. F.C. Bernstein, T.F. Koetzle, G.J.B. Williams, E.F. Meyer, M.D. Brice, J.R. Rodgers, O. Kennard, T. Shimanouchi, and M. Tasumi: *JMB*, **112**, pp.535-542 (1977).
6. J.U. Bowie, R. Luthy, and D. Eisenberg: *Science* ,253, pp.164-169 (1991).

7. T. Fukuda and S. Morishita: *Technical Report of IEICE*, DE95-6 (1995-05), pp.41-48 (1995).
8. K. Hätönen, M. Klemettinen, H. Mannila, P. Ronkainen, and H. Toivonen: *ICDE'96* (1996).
9. L. Holm, C. Ouzounis, C. Sander, G. Tuparev, and G. Vriend: *Protein Science*, **1**, pp.1691-1698 (1992).
10. M. Houtsma and A. Swami: *ICDE'95*, pp.25-33 (1995).
11. T. Ishikawa, S. Mitaku, T. Terano, T. Hirosawa, M. Suwa, and B. Seah: *Proc. of Genome Informatics Workshop 1995*, pp.39-48 (1995).
12. M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A.I. Verkamo: *3rd International Conference on Information and Knowledge Management*, pp.401-407 (1994).
13. S. Kuhara, K. Satou, E. Furuichi, T. Takagi, H. Takehara, and Y. Sakaki: *Proc. of the 24th HICSS*, Vol.I, pp.653-659 (1991).
14. S. Muggleton, R.D. King, and M.J.E. Sternberg: *Proc. of the 25th HICSS*, Vol.1, pp.685-696 (1992).
15. A. G. Murzin, S. E. Brenner, T. Hubbard, C. Chothia: *JMB*, **247**, pp.536-540 (1995.)
16. H. Nishimura, K. Ito, H. Kawano, and T. Hasegawa: *Seventh Data Engineering Workshop (DEWS'96)*, pp.79-84 (1996).
17. J.R. Quinlan: *Machine Learning*, Vol.5, No.3, pp.239-266 (1990).
18. B. Rost and C. Sander: *Proteins*, 19, pp.55-77 (1994).
19. K. Satou, E. Furuichi, S. Hashimoto, Y. Tsukamoto, S. Kuhara, T. Takagi, and K. Ushijima: *Journal of Japanese Society for Artificial Intelligence*, Vol.11, No.3, pp.440-450 (1996).
20. K. Satou, E. Furuichi, T. Takagi, and S. Kuhara: *Proc. of the 27th HICSS*, Vol.V, pp.160-169 (1994).
21. K. Satou, E. Furuichi, T. Takagi, S. Kuhara, and K. Ushijima: A. Makinouchi ed., *Proc. of International Symposium on Next Generation Database Systems and Their Applications (NDA'93)*, pp.130-137 (1993).
22. K. Satou, E. Furuichi, K. Takiguchi, T. Takagi, and S. Kuhara: *CABIOS*, Vol.9, No.3, Oxford Univ. Press, pp.259-265 (1993).
23. G. Shibayama, K. Satou, and T. Takagi: *Proc. of Genome Informatics Workshop 1995*, pp.108-109 (1995).
24. S. Shimozono, A. Shinohara, T. Shinohara, S. Miyano, S. Kuhara, and S. Arikawa: *Proc. of the 26th HICSS*, pp.763-772 (1993).
25. T. Shintani and M. Kitsuregawa: *Proc. of Joint Symposium on Parallel Processing 1996(JSPP'96)*, pp.97-104 (1996).
26. R. Srikant and R. Agrawal: *SIGMOD'96* (1996).