JAIST Repository
https://dspace.jaist.ac.jp/

Title	ゲノムデータベースにおける柔軟なデータ加工およびマイニングシステムの構築に関する研究
Author(s)	布施田, 敏樹
Citation	
Issue Date	2000-03
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/631
Rights	
Description	Supervisor:佐藤 賢二, 知識科学研究科, 修士

JAIST
JAPAN
ADVANCED INSTITUTE OF
SCIENCE AND TECHNOLOGY

Japan Advanced Institute of Science and Technology

修 士 論 文

ゲノムデータベースにおける柔軟なデータ加工およびマイニングシステムの構築に関する研究

指導教官　佐藤 賢二 助教授

北陸先端科学技術大学院大学
知識科学研究科知識システム基礎学専攻

布施田 敏樹

2000 年 2 月 15 日

Copyright © 2000 by Yoshiki Fuseda

目 次

図 目 次

表 目 次

第 1 章

はじめに

1.1 研究の背景と目的

1980 年代に開始された各種ゲノム解析計画が急速に進展し、現在までに、大腸菌や酵母菌などを含む約 20 種類以上のモデル生物の DNA 配列が決定されている。同様に 1980 年代の終りに始まったゲノムプロジェクトは、これまで主に情報処理技術という観点から情報科学と生物学の融合が行われており、多大な成果をあげようとしている。このプロジェクトは、さまざま生物種のゲノムを A,T,G,C という塩基の文字列として読み出し、中に含まれる遺伝子をすべて解読することを計画しており、最終的な目的は、DNA の塩基配列から生命を理解することにある [1, 2]。ヒトに関して言えば 21 世紀初頭、2003 年頃にはひと通りの配列が解析されると言われている [3]。

このプロジェクトにより、分子生物学の実験データを格納した科学技術データベース (ゲノムデータベース) のデータ量は指数関数的に増加している。また、データベースの統合化とインターネットの普及により、データベースは格段に使いやすくなり、利用者も増加した。しかし、それに伴い色々な問題点が浮き彫りになってきた。その問題とは分子レベルの情報を集めて統合化しても、それだけからゲノムや遺伝子の構造や機能を推測するのは困難であるということである [4]。これらの問題点を克服するために近年、高速なデータ処理機能と知識獲得機能を結合したシステムの構築することの重要性が指摘されている [5, 6]。

このような背景のもと、本研究では、データベースの研究分野で近年注目されているデータマイニングの技術を用いて、大規模なゲノムデータベースからの知識発見を支援することを考える。データマイニングは、大量のデータから有用な知識を機械的に発見するための手法である [7]。例えば、代表的なデータマイニング手法である相関ルール発見

の場合、顧客の商品購入データを対象に全ての商品の組合せに関して頻度チェックを行ない、重要度が閾値を超えるものについて「パンとバターを買う人はミルクも購入することが多い」というような併買パターンを相関ルール（パン, バター $\Rightarrow$ ミルク）の形で結果として返す。簡明で意味を把握し易い形で結果が返ってくることから、流通分野に限らず各種のビジネスデータに対して実務レベルで導入が行なわれ、目覚しい成果を上げている。

相関ルール発見に限らず、各種のデータマイニング技術を応用して科学的発見を行なう研究は、ゲノム情報処理の分野で近年最も期待され活発化しているもののひとつである。しかしながら、本研究のようにゲノムデータベース全体という巨大な情報、しかもビジネスデータとは性質の異なる情報を対象にした場合、そのまま適用するにはいくつか問題点がある。その中でも、これまでゲノム情報処理の分野で行なわれたデータマイニングの研究事例では、その対象が割合小さな数百から数万程度の要素を持つ固定したデータセットに限られていたということは、重要な問題点のひとつである。これには 2 つの理由がある。まず、データマイニング手法の多くが本質的に組合せ論的な計算であり、効率化したアルゴリズムを用いても数千万や数億といったデータを扱う事が難しく、そのため厳選した少数のデータだけを対象にするケースが多い。次に、データマイニングのアルゴリズムやソフトウェアを理解して自由に解析や科学的発見を行なえるのは現状では計算機科学の研究者にほぼ限定され、実際にゲノムデーベースから知識発見を行ないたい生物学者や医学者には、利便性の点で敷居が高い。これを解決する為には日々更新されるゲノムデータベース全体から、簡単な操作で知識発見を行なうシステムやサービスが必要だが、そのようなものは未だ存在しない。

これに対し本研究では、ゲノムデータベース全体を対象とするために、計算機上で日々更新されるクロスリファレンス情報を利用して、相関ルール発見を行なうことを考える。これにより、利用者は最新かつ巨大なデータ空間から知識発見を行なうことが可能になる。また、高速なデータマイニングサービス（組合せ爆発の回避）のためには、クロスリファレンス情報の全体を使うのではなく、利用者が興味を持っている特定のデータ集合に関連したもののみを動的に切り出して使うという新しい枠組を導入する。これにより、Web 経由でも十分サービス可能なほど高速に相関ルール発見を行なうことが期待できる。さらに、何を対象にデータマイニングを行ないたいか、どの単位でデータマイニングを行ないたいか（遺伝子単位、アミノ酸配列単位、タンパク質単位など)、どのレベルでデータマイニングを行ないたいか（データエントリレベル、エントリの一部であるコンテントレベル）などの指定を利用者から柔軟に受け入れ、必要に応じて加工を行ない、それを対象に処理を行なうようシステム設計を行なう。

1.2 本論文の構成

本論文は本章を含めて 6 章から構成される。第 2章では、準備として本研究で用いたマイニングシステムの検索エンジン部分であるアルゴリズムについて述べる。また現状のデータマイニングが、科学技術データベースに適していない点について述べる。第 3章ではゲノムデータベースの特徴と意義について述べ、現在利用可能なゲノムベースのサービスとデータの概念について説明する。第 4章では、本研究でゲノムデータベースからの動的データの加工法について説明する。第 5章では前章まで述べてきたアルゴリズムおよびゲノムデータを用いて構築したマイニングシステムについて述べる。最後に本論文の結論と今後の課題および展望について第 6章で述べる。

第 2 章

準備

本章では、本研究で使用したデータマイニングシステムのアルゴリズムおよび現状のデータマイニング (マーケットバスケット分析) の問題点について述べる。

2.1 データマイニング

近年、巨大データベースから知識を高速に獲得する手法として、データマイニング (Data Mining) がデータベース分野や人工知能分野など色々な分野において注目されている [8]。これは、データウェアハウス (データの倉庫) に入っているデータを採掘し、宝物である情報、仮説、知見、課題、法則性等を見つけ出す方法やプロセスのことである。

近年データマイニングが盛んに研究されるようになったのは、情報化社会において POS システムのデータや、顧客データなど多種多様なデータの蓄積が進み、それらを有効活用することが求められるようになったからである。また、データ収集技術の大幅の進歩と計算機や記憶装置の劇的な低価格化により、情報収集がたやすい作業になり、巨大なデータ (数ギガから数テラバイト) が蓄積されるようになったからである。この山のようなデータから属性やデータ間に成り立つ関係、規則、法則などを発見したいという要求が自然と生まれてくる。例えば、スーパーマーケットでの売り上げデータから「商品 A を購入した客は高い確率で商品 B を購入する」という規則が得られたならば、商品 A が商品 B の売り上げに貢献していることがわかり、売り上げの予測にもある程度つながる。また他の例として顧客データベースから顧客の購入記録の特徴をつかみ、データマイニングの結果、ダイレクトメールの送付先の絞り込みを行い、売り上げを増加することができたという事例も報告されている [9]。

データマイニングは多数の構成要素から成立しており、研究分野としては多岐にわたっ

ている。主な関連項目として KDD、エキスパートシステム、クラスタ分析、マーケットバスケット分析、遺伝的アルゴリズム、リンク分析、決定木、ニューラルネットワーク、統計処理、機械学習などの分野がある。

2.1.1 データマイニング手法の選択

データマイニング手法を決定する際にはいくつかポイントがある。

- **学習の容易さ** ある手法を用いた新しいモデル (与えられた入力に対して１つ以上の出力を与えるようなもの) 構築の手間はどの程度必要か。またどのようなデータ変換が必要かどうかなどが挙げられる。
- **適用の容易さ** モデルには、理解するのが容易なものとそうでないものがある。例えば決定木やマーケットバスケット分析は明快なルールをもたらし、意味もわかりやすい。しかし、ニューラルネットワークやクラスタリング手法は特定のモデルがなぜ得られたかについてほとんど教えてくれない。
- **一般性** どれだけ多くの問題解決やデータ型に適用することができるか。

本研究ではデータマイニング手法として、結果を明快なルールでもたらす探索型データマイニング手法の１つであるマーケットバスケット分析手法 (相関ルール抽出) を用いた。

2.1.2 相関ルール抽出アルゴリズム

マーケットバスケット分析はビジネスデータの解析に用いられたのが始まりである [11]。マーケットバスケット分析は結果が明快で実用的である。結果は相関ルール (**association rule**) として表現される。相関ルールはどのようにある製品とサービスが他の製品やサービスと互いに関連しているか、どのようなにそれらのグループとなっているのかを表現しているので直感的に理解することができる。分析によって得られた情報は、店舗内のレイアウト計画、製品のバンドル販売や、商品の仕入れなどさまざまな目的に活用できる。

2.1.3 アプリオリアルゴリズム

本小節では、相関ルール抽出アルゴリズムについてはじめに述べる。次に相関ルール抽出の改良版であるアプリオリアルゴリズムについて述べ、具体的な例を説明する。具体的なアルゴリズムは次のようになる。アイテム集合を $\mathcal{I} = \{i_1, i_2, \cdots, i_m\}$、トランザクショ

ンデータベースを $\mathcal{D} = \{t_1, t_2, \cdots, t_m\}$、$t_i \subseteq \mathcal{I}$ とする。各要素 t_i をアイテム集合 (itemset) と呼ぶ。そして個々のトランザクションには、ユニークなトランザクション ID *(TID)* を割り当てる。この時に導出される相関ルールは次のように表現される。

$$X \Longrightarrow Y; X \subset \mathcal{I}, \mathcal{Y} \subset \mathcal{I}, \mathcal{X} \cap \mathcal{Y} = \emptyset \tag{2.1}$$

相関ルールは支持度 (**support**) および確信度 (**confidence**) の 2 つのパラメータを持ち、これらの値は相関ルールの重要さを表す。相関ルールは常に 2 進値属性上で定義する[1]。相関ルールの $X \Longrightarrow Y$ の支持度 $support(X \Rightarrow Y)$ は D 全体に対し X, Y を共に含むトランザクションの割合 $support(X \cup Y)$ により定義され、確信度 $confidence(X \Rightarrow Y)$ は X を含むトランザクションのうち、Y も含むトランザクション割合により定義される。書き換えると次のようになる。

$$\text{支持度} : \frac{\text{アイテム集合 } X \text{ と } Y \text{ を共に含むトランザクション数}}{\text{全トランザクション数}} \tag{2.2}$$

$$\text{確信度} : \frac{\text{アイテム集合 } X \text{ と } Y \text{ を共に含むトランザクション数}}{\text{アイテム集合 } X \text{ を含むトランザクション数}} \tag{2.3}$$

相関ルールの抽出問題は、ユーザによって指定された**最小支持度 (minimum support)** と**最小確信度 (minimum confidence)** を満足するすべてのルールを見つけることである。相関ルールは次の 2 つのステップで抽出される。

1. 最小支持度を満足するアイテム集合を全て見つける。見つけたアイテム集合はラージアイテムと呼ぶ。

2. (1.) で求めたラージアイテム集合から最小確信度を満たす相関ルールを導き出す。

相関ルール抽出処理のうち、第 1 ステップは基本的に、可能な全てのアイテム集合について支持度を調べる処理である。このためアイテム数が多くなると組み合わせ論的にアイテム集合のバリエーションが増え、それに伴い計算が膨大になる。一方、第 2 ステップは第 1 ステップで最小支持度を超えたラージアイテム集合だけを対象に相関ルールの生成を行うため、第 1 ステップに比べると少ない計算量で処理できる。このため、相関ルールの研究では、第 1 ステップの効率化が試みられている。アプリオリアルゴリズムは現在最も広く引用されている基本的な逐次アルゴリズムであり、本研究のシステムもこれを用いている。アプリオリアルゴリズムは 1994 年 IBM アルマデン研究所の R. Agrawal によって提

[1]現在は連続値が扱える形に拡張されているが本論文では 2 進値の場合 [12] を対象としている。

L_1 = {large 1-itemsets};
$\overline{C}_1$ = database $\mathcal{D}$;
for $(k = 2; L_{k-1} \neq 0; k++)$ **do begin**
  C_k =apriori-gen(L_{k-1}); //新しい候補アイテムの生成
  for all transactions t $\in \mathcal{D}$ **do begin**
    C_t = {c $\in C_k$ | (c - c[k]) $\in$ t.set-of-itemsets $\wedge$ (c-c[k-1]) $\in$ t.set-of-itemsets};
    for all candidates c $\in C_t$ **do**;
      c.count++;
    if($C_t \neq 0$)**then** $\overline{C}_k$ += < t.TID,C_t > ;
  end;
  L_k ={c $\in C_k$ | c.count $\geq$ *minisup*};
end

図 2.1: アプリオリアルゴリズム

案された [13, 14]。ここで k 個のアイテムの組合わせを k-itemset、長さ k のラージアイテム集合を L_k、長さ k の候補アイテム集合を C_k とする。長さ $k(\geq 2)$ の場合の処理は次のようになる。

1. 長さ k-1 のラージアイテム集合 L_{k-1} から、長さ k の候補アイテム集合 C_k を作成する。

2. トランザクションデータベースを探索し、支持度を求める。

3. 最小支持度を満足するものを取り出し、長さ k のラージアイテム集合 L_k とする。

アプリオリアルゴリズムにおける候補集合の生成 (apriori-gen()) は、アイテム集合 L_{k-1} から次のアイテム集合 C_k を生成する。この候補集合を生成する時には最小確信度は考慮されない。

1. $p, q \in L_{k-1}$ を用意する。

2. $p.item_1 = q.item_1, \cdots, p.item_{k-2} = q.item_{k-2}, \cdots, p.item_{k-1} \leq q.item_{k-1}$ を満たすのであれば $\{p.item_1, \cdots, p.item_{k-2}, q.item_{k-1}\}$ は求める次候補集合 C_k に追加する。なお $item_1, \cdots, item_{k-2}, item_{k-1}$ は昇順に並んだ個々のアイテムを表す。

$k-itemset$	k 個のアイテムから成る集合 (以下、k アイテム集合と呼ぶ)
L_k	L_k は k アイテム集合に関する集合である。この集合は k 個のアイテムを持つ集合で、各々のメンバーは最小支持度を満たす以下の２つの要素から成る。 1. アイテム集合 2. サポートカウント すなわち L_k は長さ k のラージアイテム集合を生成した結果 (最小支持度による足切りで閾値を越えた k アイテム集合) である。
C_k	C_k は k アイテム集合に関する集合である。C_k は k アイテム集合に関する集合で、各々のメンバーは２つの要素から成る 1. アイテム集合 2. サポートカウント この候補集合から最小支持度を満たすメンバーで L_k が構成される
$\overline{C}_k$	$\overline{C}_k$ はトランザクション ID とアイテム集合の組から成る集合である。このため同一のアイテム集合であっても、トランザクション ID が異なれば別々のメンバーとなる。

表 2.1: アプリオリアルゴリズムで使用する記法一覧

3. 追加された候補アイテムから任意のアイテムを一つ取り除いたものが、L_{k-1} に含まれているか調べる。もし含まれていなければ、候補集合 C_k から取り除く。そして最終的に残るものが次のアイテム集合への候補集合 C_k となる。

２つの入力値（最小支持度、最小確信度）を満たす相関ルールを導出するアルゴリズムについて具体例を挙げる。対象となるトランザクション集合は表 2.2の Database D を用いる。トランザクション集合は、コンビニエンスストアのトランザクションデータを仮定している。

顧客番号 1 { ジュース, 弁当, おでん, カップラーメン }

顧客番号 2 { 雑誌, 弁当, カップラーメン }

顧客番号 3 { ジュース, 雑誌, 弁当, カップラーメン }

顧客番号 4 { 雑誌, カップラーメン }

の購入記録がある。このままでは非常に扱いづらいので以下のように数値データに置き換えるが、計算方法のようになる。

顧客番号 1 {1 3 4 5}

顧客番号 2 {2 3 5 }

顧客番号 3 {1 2 3 5}

顧客番号 4 {2 5 }

閾値については最小支持度を 50 %、最小確信度を 75 %とする。実行例は、表 2.2にまとめてある。

1. Database$D$ より L_1 を生成する。
   今、与えられたデータのトランザクション数が４、最小支持度が 50 %であるから、$minisup$=2 となる。トランザクションから 1 つのアイテムを持つ集合とそのアイテムがトランザクション内に存在している数を数え上げ、$minisup$ を満たすラージアイテム集合 L_1 を生成する。

2. Database$D$ より $\overline{C}_1$ を生成する。
   この場合 $\overline{C}_1$ は Database$D$ そのものになる。

3. 次に L_1 から C_2 を生成する。L_1 のアイテム集合 {{1},{2},{3},{5}} からメンバーが２となる組でアイテムを持つ集合 C_2 を生成する時の組合わせは
   {{1,2},{1,3},{1,5},{2,3},{2,5},{3,5}} となる。

4. C_2 から $\overline{C}_2$ を生成する。
   C_2 のアイテム集合が $\overline{C}_1$ のどのトランザクション ID(TID) をもつアイテムから構成されているかを検索し、TID ごとにアイテム集合を生成し、C_2 のアイテム集合が $\overline{C}_1$ 内に存在している数を数えあげる。

5. C_2 から L_2 を生成する。
   C_2 のアイテムのうち、$minisup$ を満たすものとして L_2 を生成する。

6. L_2 から C_3 を生成する。
   L_2 のアイテム集合 {{1,3},{2,3},{2,5},{3,5}} から、メンバーが３となるアイテムを持つ集合 C_3 を生成するときの組合わせは {2,3,5} となる。

7. C_3 から $\overline{C}_3$ を生成する。
   C_3 のアイテム集合が、$\overline{C}_2$ のどのトランザクション ID(TID) をもつアイテムから構成されているかを検索し、TID ごとにアイテム集合を生成し、C_3 のアイテム集合が $\overline{C}_2$ 内に存在している数を数えあげる。

Database D	
TID	Items
顧客番号 1	ジュース、弁当、おでん、カップラーメン
顧客番号 2	雑誌、弁当、カップラーメン
顧客番号 3	ジュース、雑誌、弁当、カップラーメン
顧客番号 4	雑誌、カップラーメン

Database D	
TID	Items
1	1 3 4 5
2	2 3 5
3	1 2 3 5
4	2 5

L_1	
Itemset	Support
{1}	2
{2}	3
{3}	3
{5}	4

$\overline{C}_1$	
Itemset	Set of Itemsets
1	{ {1},{3},{4},{5}}
2	{ {2},{3},{5} }
3	{ {1},{2},{3},{5}}
4	{ {2},{5}}

C_2	
Itemset	Support
{1,2}	1
{1,3}	2
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

L_2	
Itemset	Support
{1,3}	2
{1,5}	2
{2,3}	2
{2,5}	3
{3,5}	3

$\overline{C}_2$	
Itemset	Set of Itemsets
1	{ {1,3} {1,5} {3,5}}
2	{ {2,3} {2,5} {3,5} }
3	{ {1,2} {1,3} {1,5}
	{2,3} {2,5} {3,5}}
4	{ {2,5}}

C_3	
Itemset	Support
{2,3,5}	2

$\overline{C}_3$	
Itemset	Set of Itemsets
2	{ {2,3,5} }
3	{ {2,3,5} }

L_3	
Itemset	Set of Itemsets
{2,3,5}	2

表 2.2: アプリオリアルゴリズムによるラージアイテム集合の抽出例

8. C_3 から L_3 を生成する。
   C_3 のアイテム {2,3,5} は $minisup$ を満たすので L_3 を構成する。

9. 同様の手順で L_3 から C_4、$\overline{C}_3$、L_4 を生成しようとしても 4 以上のメンバーが存在しないので終了となる。

以上より、L_1、L_2、L_3 が最小支持度を満たすラージアイテム集合となる。
次に、これらのアイテム集合の確信度を求める。最小支持度を満たしているアイテム集合は L_1、L_2、L_3 である。表 2.2より L_1={{1},{2},{3},{5}}、L_2={{1,3},{1,5},{2,3},{2,5},{3,5}}、L_3={{2,3,5}} と書き換えることができる。最小支持度と最小確信度を満たしているものとして、$1 \Rightarrow 3$ や $2 \Rightarrow 5$ や $2,3 \Rightarrow 5$ など合計 16 のルールが導出される。これらの例をもとのコンビニエンスストアの例に当てはめ直してみると

カップラーメンを購入する顧客の 75 %は雑誌を購入する。
雑誌と弁当を購入する顧客は 100 %カップラーメンを購入する。

という相関ルールが得られたことになる。以上が相関ルール抽出のアルゴリズムである。Agrawal らアプリオリ以前に AIS というアルゴリズムを提案している [15]。このアルゴリズムは候補アイテム集合の絞り込みがなされていなかったが、アプリオリアルゴリズムは AIS に比べて大きく改善されている。

2.2 マーケットバスケット分析の問題点

本章では、マーケットバスケット分析の問題点について実例を挙げながらいくつか紹介する。マーケットバスケット分析は、結果が相関ルールとして抽出されるので明確で実用的に表現される。しかし、得られたルールが常に有用であるとは限らない。以下の 3 例はマーケットバスケット分析によるルールの事例である [10]。

- 木曜日にスーパーマーケットでビールと紙おむつを一緒に買う人が多い。
- 製品保証契約をつけた顧客は大型の家電製品を買う傾向がある。
- 日曜大工店の新規オープンでよく売れる物の１つがトイレットリングである。

この３つの事例は、マーケットバスケット分析でもたらされるルールの３つのタイプ:「有益なルール」「とるに足らないルール」「説明不可能なルール」を示している。

「有益なルール」は品質の高い実行可能な情報である。木曜日のビールと紙おむつの事例は、木曜日の晩に子供の紙おむつと父親のビールとを週末用に準備することを示してい

る。店側がおむつをビールのある通路の近くに置いておくと、２つの商品は売り上げをさらに伸ばすことができるのである。ルールが理解しやすいので次のような仮説を立てることも可能である。顧客がどんな品物も「忘れない」ようにビールの見える範囲に他のベビー向け商品を置くとか、他のレジャー関連の食品をベビー用品売り場に置くとか、先ほどはおむつの近くにビールを置いたが、ビールの見える範囲にベビー用品を置くなどが考えられる。

「とるに足らないルール」とは、その業界の人なら誰でもすでに知っているルールのことである。我々は、すでに大型家電製品の購買と同時に保証契約をつけることを知っている。保証契約は大型家電製品といっしょに広告されており、めったに単独で扱われることはない。この様なルールは、データ上は正しいが、使い途がないのである。同様な結果もたくさんある。ペンキを買う人はペンキ用のブラシを買う、オイルとオイルフィルターは同時に買われる、といったことと同じである。

「説明不可能なルール」とは不可解な結果であり理解することができず、それにたいして対応することが難しいものである。3 番目の事例は、新事実の発見ではないかという魅力的なものであるが、消費者行動や商品についての洞察が欠けており、次の行動を示唆することができない。開店セールの間、他の商品に比べてトイレットリングが安かったとかあるいは、少数店舗だけのデータでの例外的な結果なのかなど色々と考えられる。このようなルールは、原因が何であれ、マーケットバスケットデータからの追加分析でも確実に説明ができない。マーケットバスケット分析を行うときは、多くの結果が「とるに足らないルール」であるか「説明できないもの」であることが多い。どのルールが価値があるものかを知るためには、事前にマーケティングプランや他の外的要因、時間的な要因に関する知識を持っていることが必要とされる。

次に大規模データベースにおける相関ルールの組み合わせ問題について述べる。これもビジネスデータの事例と共に説明する。一般にファミリーレストランのメニューには数十から 100 種類前後ほどのアイテムが載っている。このためトランザクションデータから相関ルールを生成すると、アイテムのそれぞれの組合わせごとに頻度をカウントしなければならず、組み合わせの数は指数的に増大する (100 アイテムの中から 3 アイテムの組合わせの種類は 161,700 存在する)。また、典型的なコンビニエンスストアでは約 3000 のアイテムを取り扱っている。この場合、2 アイテムの組み合わせだけでも約 450 万の組み合わせとなり、3 アイテムだと約 45 億通りの組み合わせになる。表 2.3は組み合わせの数がどれほど指数関数的に増加するかを示したものである。コンピュータの価格が劇的に向上したといっても、このような組み合わせについて頻度を計算するのはほとんど不可能である。もし計算することができても現実的に使用するのは非常に難しい。またビジネスデー

組み合わせの中のアイテム数	存在する組み合わせ数	存在する組み合わせ数
1	100	3,000
2	4,950	4,498,500
3	161,700	4,495,501,000
4	3,921,225	$6.7 * 10^{12}$
5	75,287,520	$2.0 * 10^{15}$

表 2.3: 100、1000 アイテムでの組み合わせ爆発の例

タの場合トランザクションの数も大変大きい。小規模なスーパーマーケットでさえ年間に数百万のトランザクションを生成する。この各トランザクションでは１つ以上、しばしば数十のアイテムが購入される。したがって、特定のトランザクションにアイテムの特定の組み合わせがあるかどうかを求めるためには、全てのトランザクションについて 100 万倍した計算量が必要になる。

以上にビジネスデータを例に挙げてマーケットバスケット分析が抱える本質的な問題点について説明を行ってきた。大量のデータに対していかに計算の爆発をおさえるかという意味では、本研究で取り扱うゲノムデータベースでも同様な事が言える。しかしながら、通常のビジネスデータを対象としたデータマイニングでは起らないような問題も研究を進める過程で発見された。これについては第 5章の後半で述べることにする。

第 3 章

ゲノムネットのデータベース

3.1 ゲノムネット

1980 年代終りに開始されたヒトゲノム計画は、分子生物学の技術革新の流れと医学研究への期待から始まったプロジェクトである。これに呼応して各種のモデル生物に関する配列決定プロジェクトが立ち上がり、多種多様な生物のゲノムの全塩基配列という大量データの出現に対処するため、当初から新しい情報処理技術と情報インフラストラクチャの整備に重点が置かれていた。また、計画の波及効果として生命科学全般の情報化が引き起こされ [16, 17]、それに伴い分子レベル、細胞レベル、個体レベル、生物種レベルでの生命現象に関する基礎データや、病気の診断・治療への可能性を示す様々なデータが、急速に蓄積されてきた。情報化には二つの意味がある。１つは情報処理技術が実験技術と同じように生物科学の研究を行う上で不可欠の要素になってきたこと。もう一つは、ゲノム計画がもたらす大量のデータの出現によりデータベースの重要性が高まってきたことである。

日本では、1991 年度より開始された文部省ヒトゲノムプロジェクトにより、京都大学化学研究所と東京大学医科学研究所ヒトゲノム解析センターが中心となって、ゲノムネット (GenomeNet) と名付けたコンピュータネットワークの構築、整備、運用を行ってきた。ゲノムネットは国内で他のネットワークと相互乗り入れしているだけなく、世界中のネットワークをつなぐインターネットの一部である。しかし、ゲノムネットは単なるネットワークの集合体ではない。生物学の分野では、文献情報の他に、ゲノムの地図、塩基配列、タンパク質のアミノ酸配列や立体構造、代謝系や制御系の分子ネットワーク、神経系や免疫系における細胞のネットワーク、そして発生・分化・老化や疾病に関する個体レベルのデータなど多種多様なデータが世界中でデータベース化され公開されている。これら

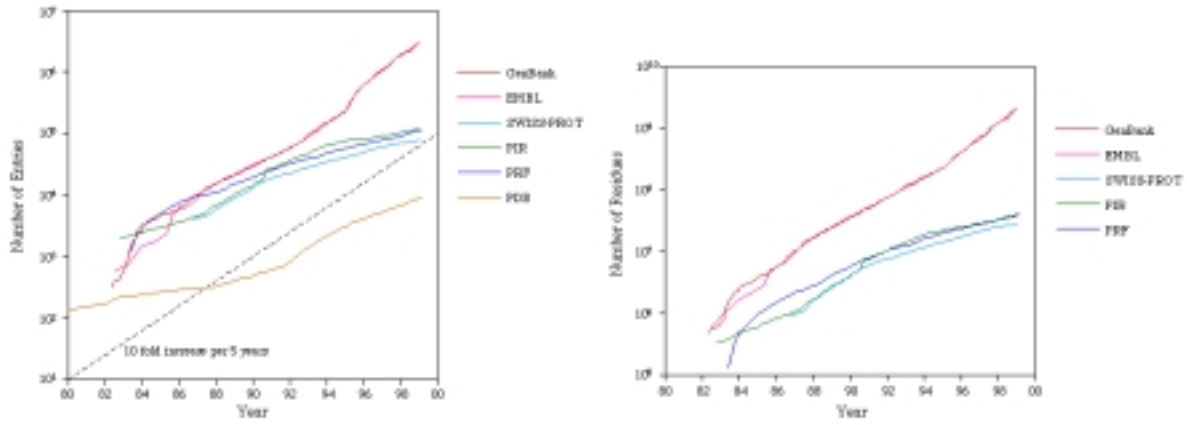

図 3.1: 主要データベースのデータ遷移図

は相互に深く関連しており、頻繁に更新されている。データの種類もテキスト形式だけでなく、イメージ画像やグラフィックスなどがある。ゲノムネットでは、既存の主要なデータベースを集積しネットワーク利用環境を構築すると同時に、バイオサイエンスに対応した新しいデータベースシステムの開発と実用化をめざしている。このようにゲノムネットは多様なデータベースをそれぞれ統合して利用できる情報サービス網である。1999 年からは京都大学化学研究所と東京大学医科学研究所ヒトゲノム解析センターに加え、北陸先端科学技術大学院大学でもゲノムネットの運用を行っている。

3.2 ゲノムデータベースの特徴と意義

ゲノム計画の開始とともに各研究プロジェクトを推進するためにゲノムデータベースが誕生した。ゲノム計画では、各種多量のデータが産出される。このことは、以前の生物学では考えられなかったことであり、とてもこれらのデータを人手で整理解析することはできない (図 3.1参照)。そのため、当初からゲノム計画の遂行には欠かせないものになっている。大げさに言うと、データベースがなければゲノム計画は進展しなかったとも言える。例えば、ゲノムの配列を手にいれてもデータベースがなければデータの意味を知ることができないからである。つまりデータベースは、重要な道具であると同時に研究の成果そのものである [18, 19, 20]。このような意味においてもデータベースはゲノム計画にとってかなり重要な意味がある。

分子生物学におけるデータベースの役割は大きく分けて２つある。ひとつは研究成果と

して公表されたデータを蓄積したデータベースで、不特定多数の利用者を対象としている。例えば、現在ではゲノム研究者以外の生命科学やバイオ産業に関わる人々にもゲノムデータベースが利用されるようになってきている。もうひとつは、研究プロジェクトを遂行するためのデータベースで、一般に特定のグループ内で利用されている。

ゲノム情報は、ゲノムの塩基配列データとそれに付随するデータ (各種アノテーション) からなる。そこでゲノム情報の付随データを質量ともにできるだけ充実させ、それを使っていろいろなデータの解釈をしたり、その中から生物学医学の知識を引き出すことが必要となる。逆に付随データがなければ、ゲノムの塩基配列の生物学的医学的な意味は明らかにならない。これによりデータベースはゲノム解析に欠かせないものになっている。

付随データのデータベース化の問題は、各種生物学関係のデータベースの統合化の問題に置き換えて考えることができる。ヒトゲノム計画が始まった 1980 年代の終り頃には、現在よく使われている核酸配列データベース、アミノ酸配列データベース、タンパク質データベースなど主要なデータベースはすでに確立され一般に普及していた。しかしながら、これらのデータベースは、その開発の歴史的な経緯や政治的な背景などによりばらばらに管理運営されてきた。このため、複数のデータベースにまたがるような検索や解析は困難であった。そこで上記の統合化の問題を解決すべく、ゲノムネットを始めとするデータベースの研究開発が行われてきた (表 3.1)。

3.2.1 基本的な検索システム

ゲノムネットの基本的なサービスは

http://www.genome.ad.jp(英語版)
http://www.genome.ad.jp/Japanese(日本語版)

のリンクからたどることができる。おもな検索システムとして、簡単なキーワード検索やエントリの取得およびクロスリファレンス情報の検索を目的とした DBGET/LinkDB、配列データベースに関する代表的なホモロジー検索システム[2]である BLAST と FASTA、そしてモチーフ検索[3]システム MOTIF などがある。BLAST、FASTA、MOTIF の検索結果にはすべて DBGET/LinkDB へのハイパーリンクがついているので、さまざまなデータベースを参照して検索結果をより深く解釈することができる。

[2]ホモロジー検索は相同性検索とも呼ばれ、対象となる核酸配列やアミノ酸配列など相同な配列がデータベースに存在するか検索する手法 [16]。

[3]モチーフ検索は与えられた配列から特定の機能や立体構造についての情報を得るための手法であり、配列残基の多様性を表現した個々のモチーフがその機能や構造の情報とともにモチーフライブラリと呼ばれるデータベースに登録されていることから、その配列に関する情報を得るものである [16]。

データの内容	データベース名	メディア
塩基配列	GenBank,EMBL	テキスト
アミノ酸配列	SWISS-PROT, PIR, PRF	テキスト
	PDBSTR	
立体構造	PDB	テキスト、三次元グラフィックス
配列モチーフ	EPD, TRANSFAC, PROSITE	テキスト、三次元グラフィックス
酵素反応	LIGAND/ENZYME	テキスト
代謝化合物	LIGAND/COMPOUND	テキスト, イメージ
		二次元グラフィックス
パスウェイマップ	KEGG/PATHWAY	テキスト, イメージ
ゲノムマップ	KEGG/GENOME	テキスト, イメージ
		Java アプレット
遺伝子カタログ	KEGG/GENES	テキスト, 階層型テキスト
発現マップ	KEGG/EXPRESSION	テキスト
遺伝病	OMIM	テキスト
文献 (タンパク質)	LITDB	テキスト
文献 (医学・生物学)	Medline	テキスト
リンク情報	LinkDB	リンク情報

表 3.1: ゲノムネット上のデータベース一覧

システム	内容	利用形態	作成者
DBGET/LinkDB	統合データベース検索	WWW,Client	京大化研
		E-mail	
KEGG	遺伝子・ゲノム百科事典	WWW,FTP,CD	京大化研
PATHWAY	パスウェイ検索	WWW,E-mail	京大化研
BLAST	ホモロジー検索	WWW,E-mail	NCBI
FASTA	ホモロジー検索	WWW,E-mail	W.Pearson
MOTIF	タンパク質モチーフ検索	WWW	京大化研, 基生研

表 3.2: 利用可能な検索システム

3.3 DBGET/LinkDB 統合データベースシステム

DBGET/LinkDB 統合データベースシステム (図 3.2) は、ゲノムネットサービスにおけるバックボーンをなす統合データベース検索システムである。ここで述べているデータベースは、ほとんどがフラットファイル[4]として提供されている。マップ、塩基配列、アミノ酸配列、立体構造、配列モチーフ、酵素反応、文献情報、代謝パスウェイ、遺伝子カタログ、変異タンパク質、アミノ酸指標、遺伝病などの既存の分子生物学のデータベースのほとんどすべてがフラットファイルとして閲覧が可能になっている。各エントリには、エントリ名としてデータベース内でユニークな名前が与えられている。例えば **DATABASE:ENTRY** といった形でデータベース名とエントリ名の組を与えると世界中に存在する数多くのデータベースを統合的に参照することができる。

上述のエントリ取得機能およびキーワード検索機能は DBGET システムで実現されている。一方、エントリ内に埋め込まれているクロスリファレンス情報 (同じデータベースや他のデータベースにある、別のエントリへの参照情報) を使って、特定のエントリに関連するエントリ集合を高速に選び出す機能は、LinkDB によって実現されている。LinkDB では、リンクの正引きや逆引きに加えて個々のリンクをたどって得られる間接的なリンクについても予め全ての検索を行い、結果を保持しておくことで、関連エントリ集合の高速な取得が可能になっている。塩基配列データベース、アミノ酸配列データベースをはじめとする多くのデータベースは毎日更新されており、それにともなって、LinkDB も毎日更新されている [21]。

3.3.1 リンク情報の概念

本小節では、エントリ内に書かれている内部形式、LinkDB 内のリンク情報の概念、およびエントリの内部形式について述べる。図 3.3に示した例は、遺伝子配列データベース GenBank エントリの１つである [22]。GenBank のエントリは、１つのエントリが１つの遺伝子に相当することもあれば、遺伝子１つの中のエクソン[5]や、一部の断片であることもある。また逆に、ウイルスなどの場合はそのゲノム全体が１つのエントリというものもある。エントリの中身は、いくつかのフィールドに分かれている。LOCUS フィールドに書かれている HUMSRC04 は、このエントリの名前を表した識別子である。また、長

[4]データを単純なテキストファイルの形に編成したもの

[5]遺伝子 DNA のうち、一個のポリペプチド鎖 (または tRNA または rRNA) の情報を持つ一区画をいい、これは真核細胞遺伝子のある部分を占める [24]。

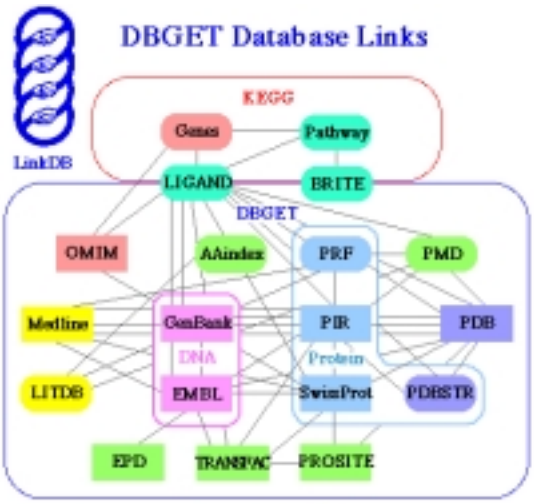

図 3.2: DBGET リンク・ダイアグラム

さ 115 の塩基対[6]であるといったことが書かれている。DEFINITION フィールドは、エントリの定義、タイトルである。ACCESSION フィールドには、LOCUS とは別の識別子があり、これはデータベースに登録したときに交付される受付番号である[7]。KEYWORDS フィールドにはエントリの内容に関連した語句が列挙してある。SOURCE フィールドにはこの配列をもつ生物種に関する情報、REFERENCE フィールドにはこの配列情報が掲載されている文献などの情報が書かれている。FEATURES フィールドは、配列情報の中で、どの場所が生物学的にどんな意味をもつかを表している。FEATURES フィールドには、配列に関する重要な情報が色々と書かれているが、現状では配列情報に比べてそれほど利用されていないことが多い。その理由のひとつには自然言語で書かれていていてパースしにくいということがある。BASE COUNT フィールドは A,T,G,C ごとの数の集計である。最後に、ORIGIN フィールドには、このデータベースの実体情報というべき塩基配列情報が a,t,g,c で書かれている。

前節で述べたように DBGET/LinkDB 統合データベースシステムが提供する各種ゲノムデータベースの間には、互いにリンクがはられている。一般に１つのエントリには、他のエントリへのクロスリファレンス (リンク) が埋め込まれている。データベース全体からこのような情報を抽出した結果を、ここではリンク情報と呼ぶ。リンク情報には、エン

[6]DNA は通常、2 本鎖の 2 重らせんをつくり、そのなかで A,T,G,C の塩基はそれぞれ A:T,G:C のように対をつくっているため、RNA の長さを数えるのに何塩基対ともいう。

[7]データベース登録を義務づけるために、これを提示しないと論文を受け付けてくれないジャーナルも存在する。

図 3.3: GenBank のエントリの例.

トリ名、リンク先へのリンク形式、リンク先への経路が書かれている。リンク情報を得るためには、2 種類の方法がある。１つは WWW 版であり、DBGET では、LinkDB を検索するための blink モードインターフェイスが用意されている (図 3.4)。blink を利用するためには、DBGET のリンク・ダイヤグラム (図 3.2) から LinkDB と書かれた部分をクリックすると blink モードへの検索画面へと移る。blink モードを使う場合は、データベース名とエントリ名の組を指定する。結果として、入力したエントリに関連するエントリのリストが表示される (図 3.5) 。

もう１つの検索方法は、コマンドによる検索である。コマンド実行時には、WWW 版と同様の書式で、blink *dbname:entryname* のように入力する。図 3.6は 1 エントリからのリンク情報の関係を表している。コマンド版 blink では、入力したエントリに対して出力される結果は、リンク先のデータベース、リンクのタイプ、リンク経路の情報を得ることができる。

これまで述べてきた LinkDB の各データベースのリンク数についてまとめる。表 3.3に書かれているリンク数は、各データベースにおける総リンク数を表している。このようなクロスリファレンス情報が LinkDB 全体で約 1200 万エントリ集積されている (2000 年 2 月 10 日現在)。

図 3.4: WWW 版 blink の入力画面

図 3.5: WWW 版 blink モードで PDB:101M からリンクを検索した結果

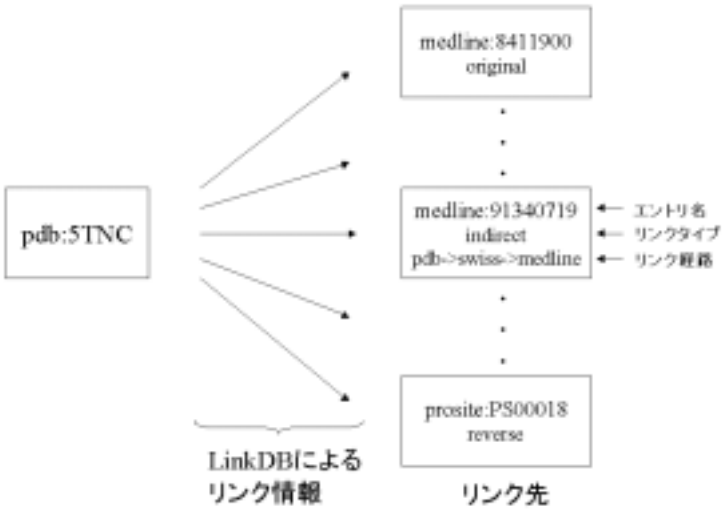

図 3.6: リンク情報の例

```
db2(137):% blink pdb:101M
medline:77144097        indirect        pdb->swiss->medline
medline:77144098        indirect        pdb->swiss->medline
medline:81119812        indirect        pdb->swiss->medline
medline:91132649        indirect        pdb->swiss->medline
pdbstr:101M     reverse
pir:A02506      indirect        pdb->swiss->pir
prosite:PS01033 reverse
swissprot:P02185        reverse
```

図 3.7: コマンド版 blink モードで PDB:101M からのリンクを検索した結果

データベース名	データの内容	リンク数
AAindex	アミノ酸指標	7026
COMPOUND	酵素反応に現れる化合物データ	44675
EMBL	核酸塩基配列	8073035
ENZYME	酵素反応データ	206285
EPD	真核生物プロモーター	21996
GenBank	核酸塩基配列 (DDBJ を含む)	11026703
Genes	KEGG 遺伝子カタログ	2277039
LITDB	タンパク質関連文献	300275
Medline	医学・生物学文献	3712844
OMIM	ヒト遺伝病	208537
Pathway	パスウェイマップ	153731
PDB	タンパク質立体構造	472151
PDBSTR	PDB アミノ酸配列	510671
PIR	タンパク質アミノ酸配列	1465640
PRF	タンパク質アミノ酸配列	451275
PROSITE	タンパク質配列モチーフ	473424
PMD	変異タンパク質	79647
SwissProt	タンパク質アミノ酸配列	980611
TRANFAC	転写制御因子	24841

表 3.3: LinkDB 内のリンク情報の数 (データベース別)

Link table for PROSITE

Links from PROSITE are computed according to the following paths.

Factual Links

PROSITE (<>) SWISS-PROT -> EMBL -> **GenBank**

PROSITE (<>) SWISS-PROT -> **EMBL**

PROSITE (<>) **SWISS-PROT**

PROSITE (<>) SWISS-PROT -> **PIR**

PROSITE -> **PDB**

PROSITE <- **TPCLASS**

PROSITE <- **ENZYME**

PROSITE <- ENZYME -> **PATHWAY**

PROSITE (<>) SWISS-PROT -> **OMIM**

PROSITE (<>) SWISS-PROT -> **MEDLINE**
PROSITE (<>) SWISS-PROT -> EMBL -> GenBank -> **Medline**

PROSITE <- ENZYME <- **E.coli**
PROSITE <- ENZYME <- **H.influenzae**
PROSITE <- ENZYME <- **B.subtilis**
PROSITE <- ENZYME <- **M.genitalium**
PROSITE <- ENZYME <- **M.pneumoniae**
PROSITE <- ENZYME <- **M.jannaschii**
PROSITE <- ENZYME <- **Synechocystis**
PROSITE <- ENZYME <- **S.cerevisiae**
PROSITE <- ENZYME <- **H.sapiens**

Biological Links

PROSITE <- ENZYME <- (PATHWAY) -> ENZYME -> **PROSITE**

PROSITE <- ENZYME <- (PATHWAY) -> **ENZYME**

->: original link
<-: reverse link
<>: both original and reverse links

図 3.8: リンクテーブルの例 (PROSITE)

第 4 章

クロスリファレンス情報を用いたデータマイニング

本章では、前章まで述べてきたLinkDBのクロスリファレンス情報を用いたデータマイニングについて述べる。

4.1 関連研究

ゲノムデータを用いたデータマイニングについては、これまでにもいくつか研究されてきた。東京大学医科学研究所の佐藤らの研究グループは、ゲノムデータベースの一部に対して相関ルール発見手法を適用し、異種データベース間すなわちタンパク質の配列、構造、機能の３種類にまたがる相関ルールを発見することに成功した。結果として得られた相関ルールの中には、例えばカルシウム結合タンパク質群に共通かつ特有な部分配列が発見された [23]。大阪大学細胞生体工学センターの大久保らは、BODYMAP という遺伝子発現データベースを構築している。これは、ヒトおよびマウスの約 2 万個の遺伝子が生体内のどの組織で働いているかを網羅的に調べた結果をまとめたもので、世界的にもユニークなプロジェクトとして知られている。東京大学医科学研究所の森下らは、BODYMAP から得られる遺伝子発現量データに対しデータマイニング技術を適用し、遺伝子のクラスタリングや遺伝子同士の相関関係を抽出することを試みている。特に、後者については次世代のゲノム解析として研究が活発化されている遺伝子ネットワーク解析に直接貢献することから、今後の進展が期待されている。以上、２つの関連研究を事例として挙げたが、本研究のように大規模なゲノムデータベース全体を対象とした研究は過去にも例がない。

図 4.1: クロスリファレンス情報からの 2 値情報への変換

4.2 本研究のアプローチ

ゲノムデータベース全体を対象としたデータマイニングを可能にするため、本研究では LinkDB が提供するクロスリファレンス情報を用いてデータマイニングを行うことを考える。個々のクロスリファレンス情報は「あるエントリと別のエントリの間に参照関係がある」という真なる命題を表現している。そこで本研究では、クロスリファレンス情報を全て２進値に変換してデータマイニングを行った。すなわち、あるユーザが入力したエントリに対して、クロスリファレンス情報があるものを１、クロスリファレンス情報がないものを０として扱うことにした。図 4.1はコマンド版 blink で得られたクロスリファレンス情報が相関ルール発見のための 2 進値データ (ビットベクターテーブル）に変換可能であることを例示したものである。しかし、LinkDB には表 3.3で示したように約 1200 万のクロスリファレンス情報が収められているため、これら全てのデータを２進値テーブルに変換すると 10^{17} を越えるセルを持つ巨大なテーブルになる。よってこのままではデータマイニングを行うことは不可能である。しかし、実際に利用者は全てのデータをマイニングにかけて全ての相関ルールを得るよりも、むしろ自分が興味を持っている部分に関してのみ、高速に知識発見を行うことを望んでいる。そこで本研究では、次のようなアプローチでマイニング用のデータの加工を行うことを考える。

図 4.2: TARGET により限定された探索空間

- ユーザは、自分が興味を持っているデータ集合を表す「*TARGET*」という情報をシステムに与える。
- *TARGET* に関係のあるクロスリファレンス情報だけを切り出し、それを対象としてマイニングを行う。
- 同様にクロスリファレンスに書かれている Type および Link Path 情報に関してもユーザの指示を受け入れる。

図 4.2は *TARGET* を導入したことにより縮小したクロスリファレンス情報である。これにより、ユーザのほしい情報だけに探索空間を限定することができ、高速にデータマイニングを行えるようになる。

4.3 エントリ内のフィールド抽出

前節で述べたクロスリファレンス情報は、いわばエントリ間の情報であり、このままではエントリ単位でしかマイニングを行うことができない。エントリ内には図 3.3のように各種のフィールドがあり、フィールド内に埋まっている情報を使って詳細なレベルでマイニングを行いたい場合は、マイニングに先立ってこれも抽出しておくことが必要である。ここでは、研究室で別途製作された entry-splitter.pl という Perl プログラムを用いて、ま

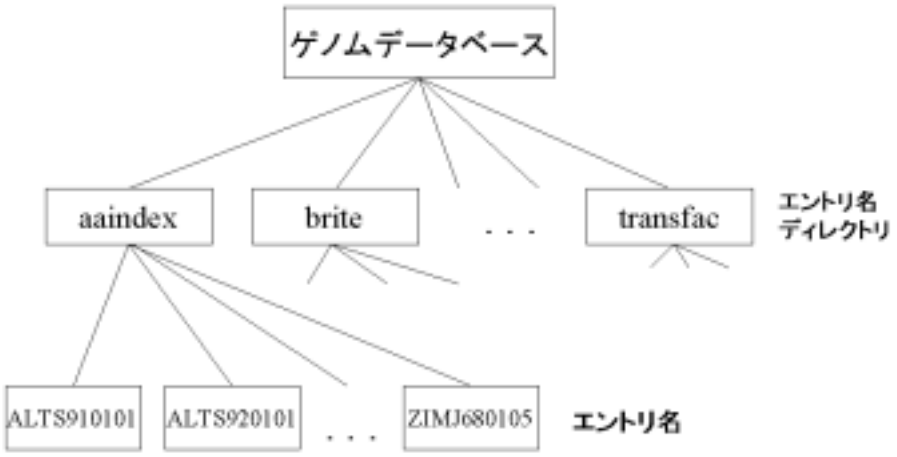

図 4.3: エントリごとに分割された階層構造

```
db2(167):% pwd
/home/db111/warehouse/entry/aaindex
db2(168):% ls
ALTS910101  FASG760102  KOSJ950115  OOBM850103  RADA880101
ANDN920101  FASG760103  KRIW710101  OOBM850104  RADA880102
ARGP820101  FASG760104  KRIW790101  OOBM850105  RADA880103
ARGP820102  FASG760105  KRIW790102  OVEJ920101  RADA880104
```

図 4.4: aaindex におけるエントリ単位の情報抽出の例

ずエントリをフィールド単位に分解することを考える。entry-splitter.pl は、大きく分けて二つの機能を持つ。

- ゲノムデータベースに収められているデータベース群をエントリ単位でファイルとして切り出す。具体的には図 4.3、4.4のようになる。
- ゲノムデータベースに収められているデータベース群をフィールド単位でファイルとして切り出す。各ファイルはフィールド名と同じディレクトリの下に置かれる。具体的には図 4.5、4.6のようになる。

本研究では、後者の機能を用いてフィールド抽出を行った (図 4.7)。同じ方法で、LinkDB に集積されているデータベースについてほぼすべてのエントリフィールドを抽出すること

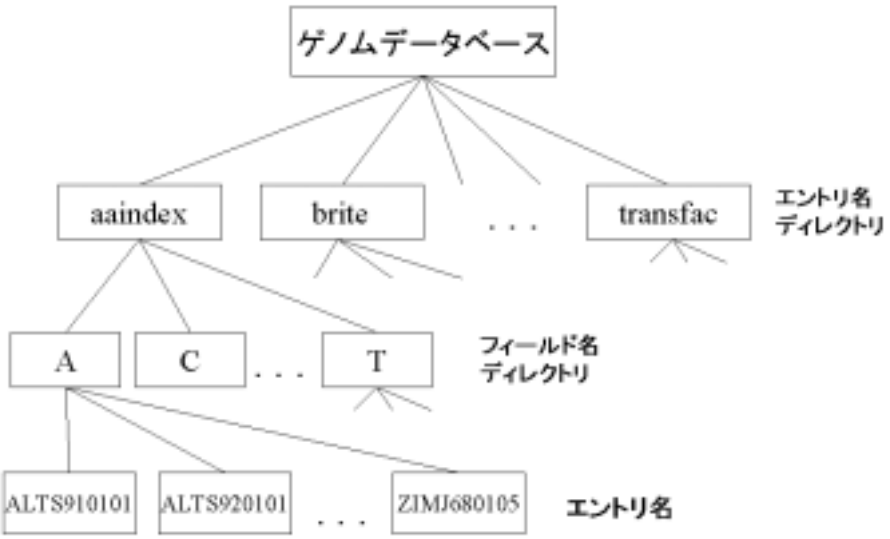

図 4.5: エントリからのフィールド抽出

```
db2(223):% pwd
/home/db110/warehouse/field/aaindex/A
db2(224):% ls
ALTS910101  FASG760102  KOSJ950115  OOBM850103  RADA880101
ANDN920101  FASG760103  KRIW710101  OOBM850104  RADA880102
ARGP820101  FASG760104  KRIW790101  OOBM850105  RADA880103
ARGP820102  FASG760105  KRIW790102  OVEJ920101  RADA880104
db2(225):% less ALTS910101
A Altschul, S.F.
```

図 4.6: aaindx をフィールド単位でファイルに切り出した例
(aaindex には A という名前のフィールドがありディレクトリ名 A はこれを表す)

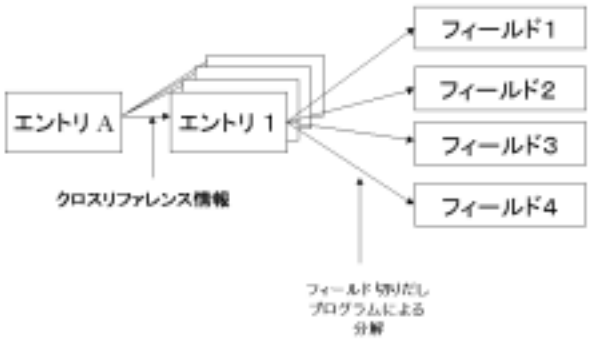

図 4.7: エントリからのフィールド抽出

ができる。これを使えばフィールド内に埋まっている詳細な情報の切り出しがやりやすくなる。しかし、フィールド内に書かれている情報をすべてデータマイニングのデータとして使用することはできない。その理由として次のようなことが挙げられる。

- omim(タンパク質関連文献データベース) のようにフィールド内に書かれている情報のほとんどが自然言語情報である場合、単純に属性と値のペアとしてデータを切り出すことができない。そのため、本研究では、自然言語が多く書かれているフィールドに関しては、データマイニングの対象から除外した。これには、著者情報やタイトル情報、ジャーナル情報なども含まれる。

- GenBank、EMBL(核酸配列データベース) などのデータベースには各エントリの最後に配列情報が記載されている。配列情報は a,t,g,c の文字が並んでいるだけなので、配列解析プログラムを適用するなりして何らかの意味を表現する値に変換しない限り、マイニング用のデータには成り得ない。この理由で、配列情報についても除外した。

- 他にもフィールド内情報を見てマイニングのデータとしてふさわしくないと思われるフィールドに関しては、マイニングの対象としなかった。

結局、本研究では、以下のデータベースのフィールドを用いた。

aaindex,brite,compound,enzyme,litdb,omim,prf,prosite,swissprot,tranfac

具体的なマイニングの例については 5.2.2 で述べる。他のデータベースのフィールドに関しては、1 エントリ内に含まれるフィールド情報の数が非常に多く、現在のところ未着手である。

第 5 章

ゲノムデータを用いたデータマイニングシステムの構築

これまでの章は、データマイニングシステムの構築に必要なアルゴリズムおよびシステムで使用するデータの作成法を中心に述べてきた。本章では、これらのアルゴリズムおよびデータを用いたデータマイニングシステムの構成および提供するサービスについて述べ、システムの利用の手順について説明する。

5.1 システム構成

図 5.1に示すように、本システムではユーザはすべてのデータ入力および結果出力を WWW 経由で行う。ユーザが自分の調べたい情報 (データ) をブラウザに入力すると WWW サーバ経由で CGI にデータが受け渡される。CGI は、ユーザの指示に従ってゲノムデータベースから必要な情報を参照し、マイニング用のデータに加工する。加工されたデータはデータマイニングプログラムに渡される。マイニングプログラムは、ユーザが指定した条件下で計算をおこない、指定された閾値を超えた相関ルールを再び WWW サーバ経由でユーザ側のブラウザに結果として表示する。

本システム環境は、Sun S-7/400Ui 上で Perl および CGI プログラミングを用いて開発を行った。ゲノムデータが収められているサーバは、Sun Microsystems 社 Enterprise 3000 である。サーバマシンは 2 台で構成されており、そのマシンスペックは 4 cpu ／メインメモリ 4.0GB および 4cpu ／メインメモリ 1.0GB である。

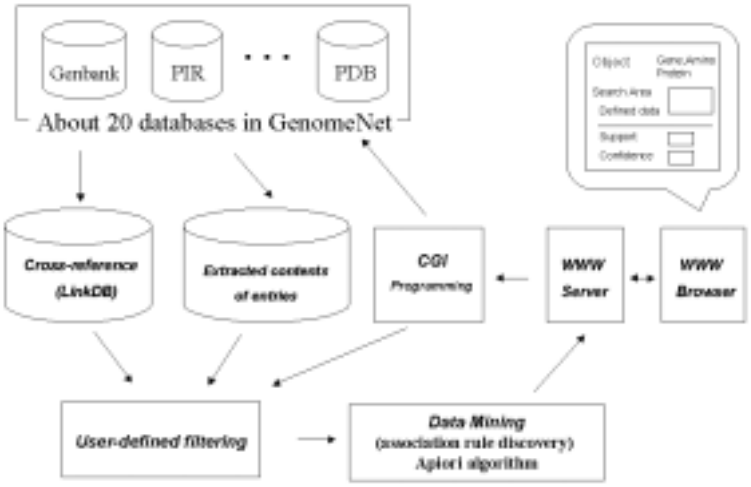

図 5.1: システム構成

5.2 異種データベース間のデータマイニング

5.2.1 エントリ単位のデータマイニング

LinkDB のクロスリファレンス情報を利用することで、エントリ単位のデータマイニングを異種データベースにまたがって行うことができる。本論文ではこれを **ENTRY-ENTRY Data Mining** と呼ぶ。図 5.2は、ENTRY-ENTRY Data Mining の入力画面である。まず、各入力ボタンおよび入力フォームについて説明する。

- **Database for mining:** データマイニングで調べたいエントリが収めらているデータベースを選択する[8]。
  - **Nucleic Acid Database** 核酸配列データベース群
  - **Amino Acid Database** アミノ酸データベース群
  - **Protein Database** タンパク質データベース群
  - **The Rest Databases** 上記以外のデータベース群
- **Filtering LinkDB:** 入力したエントリに対してのリンク先のデータの情報をより細かくするためのオプション。

[8]Database for mining に書かれているデータベースの詳細情報は、図 3.1を参照

図 5.2: エントリ単位データマイニングの入力画面 (ENTRY-ENTRY Data Mining)

  - **Destination** リンク先のデータベースを指定する。入力形式は *database:entry name* もしくは **database:∗**という書式で指定する。「∗」は指定した database 全体を表す。
  - **Type** リンク先のデータベースへのリンクの種類を限定する。
    - **original** 直接リンク
    - **indirect** 間接リンク
    - **reverse** 逆向きリンク
  - **Link Path** リンク先のデータベースへの経路を限定する。入力形式は、リンク元データベース → リンク先データベースという書式で指定する。ただし、エントリによっては、複数のデータベースを経由してリンク先のデータベースに到達するものもある。

- **Cutoff values:** データマイニングを実行する際の閾値。
  - **Support** 入力したエントリに対する最小支持度。
  - **Confidence** 入力したエントリ間の最小確信度。

- **List of target entries:** データマイニングを行いたいエントリデータを入力するためのフォーム。ただし入力するエントリはカンマで区切って入力する。
- **submit:** このボタンを押すとデータマイニングの計算を開始する。

実際の使用方法については以下で例を用いて説明する。タンパク質立体構造データベースから選択した 24 のエントリに関して ENTRY-ENTRY Data Mining を行う。データマイニングを実行する際の各オプションの条件は、つぎのように設定した。

- 入力エントリ (TARGET)

  ```
  1a29,1a75,1aui,1avs,1bf5,1cdl,1cdp,1omd,1pal,1pon,
  1rec,1rro,1rtp,1tcf,1tco,1tn4,1tnq,2scp,2tn4,3ctn,
  3pat,5cpv,5pal,5tnc
  ```

- Database for mining = pdb(リンク元のデータベース名)
- Filtering on LinkDB
  - Destination = prosite:* (リンク先は prosite 全体)
  - Type = reverse (逆向きのリンク)
  - Link Path (リンクパスの指定なし)
- Cutoff values
  - Support = 18 entries (全入力エントリに対する支持度 66.6 %)
  - Confidence = 80 %

以上の条件下でデータマイニングを行った結果を図 5.3に示す。この例では、データマイニングの結果得られた情報はユーザが指定した 24 のエントリ集合に関して、支持度 23 entries、確信度 95.8 %で、prosite のエントリ PS00018 への相関ルールがあったことがわかる。

図 5.4は相関ルールに埋め込んだハイパーリンクをたどって PS00018 のエントリを表示した所である。エントリの表示には第 3章で説明した DBGET システムを用いている。

図 5.3: エントリ間データマイニングの計算結果 (ENTRY-ENTRY Data Mining)

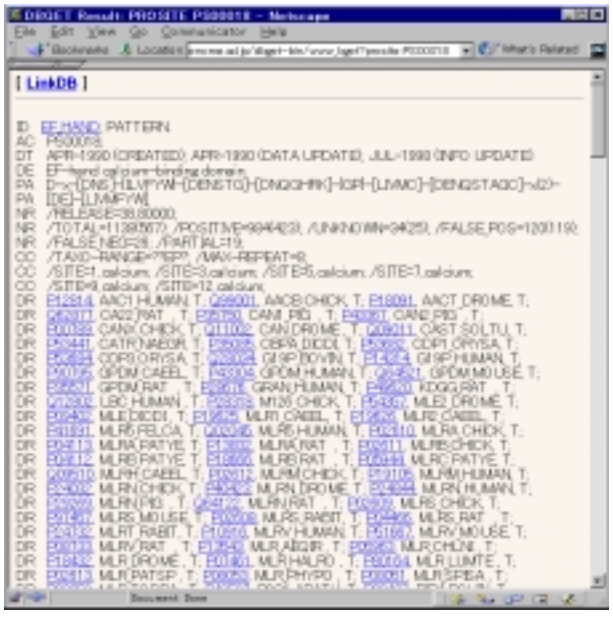

図 5.4: bget による詳細情報

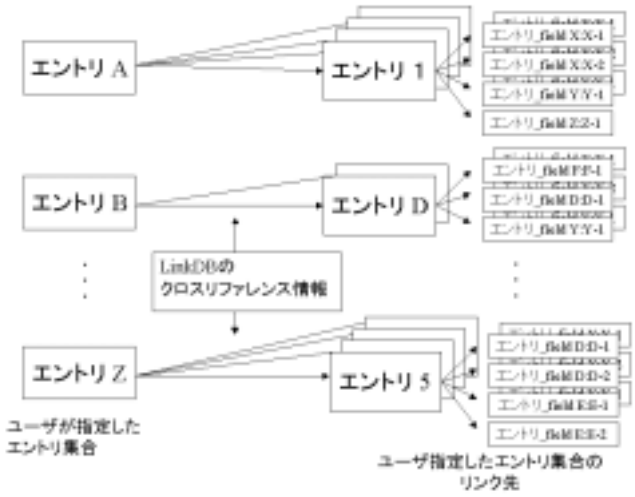

図 5.5: エントリ、フィールド間データ概念図

5.2.2 フィールド内の情報を単位とするデータマイニング

前節で説明を述べたエントリ単位のデータマイニング (ENTRY-ENTRY Data Mining) は LinkDB が提供するエントリ間の関係に基づいていた。ここではさらに詳細なデータすなわちフィールド内のデータを用いたデータマイニング (4.3 節参照) について述べる。フィールドから抽出したデータをクロスリファレンス情報と結びつけることにより、詳細なデータマイニングを行うことが可能になる。本研究ではこのような処理機能として以下の 2 種類を作成した。

- LinkDB で得られたリンク先のエントリをフィールド単位 (コンテント情報) まで分析したデータマイニング。本論文では **ENTRY-CONTENT Data Mining** と呼ぶ。図 5.5は ENTRY-CONTENT Data Mining で使用するデータの概念図である。

- LinkDB で得られたリンク先のエントリをフィールド単位 (コンテント情報) まで分析する。さらにユーザが入力したエントリ集合もフィールド単位まで処理を行う。本論文ではこれを **CONTENT-CONTENT Data Mining** と呼ぶ。図 5.6は CONTENT-CONTENT Data Mining で使用するデータの概念図である。

次に、ENTRY-CONTENT Data Mining の利用方法について説明する。図 5.7は、ENTRY-CONTENT Data Mining の入力画面である。ENTRY-ENTRY Data Mining の各入力ボタンおよび入力フォームと同様のフォームを使用しているので、詳細については第 5.2.1 節の Entry-Entry Data Mining の各入力ボタンおよび入力フォームを参照されたい。以下

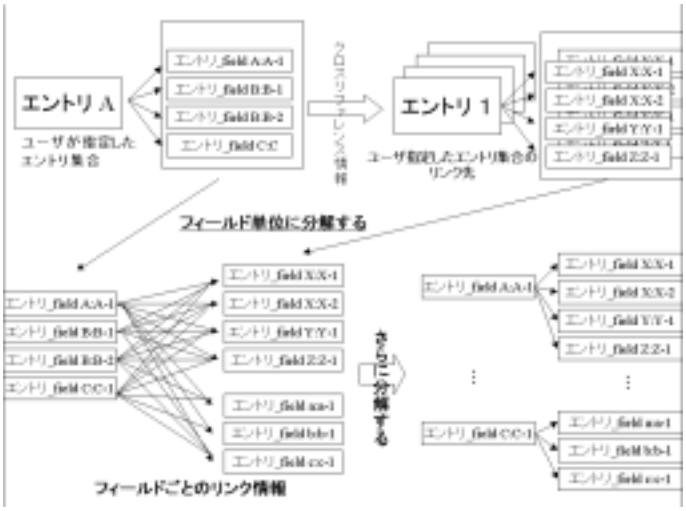

図 5.6: フィールド、フィールド間データ概念図

図 5.7: エントリとフィールド内の情報によるデータマイニング入力画面

図 5.8: エントリとフィールド内の情報によるデータマイニングの計算結果

では実際にゲノムデータを用いた例を使って説明する。この例では酵素反応に関するデータベース enzyme のエントリ５つに関してデータマイニングを行う。データマイニングを実行する際の各オプションの条件は、以下の通りである。

- 入力エントリ (TARGET)

  `1.1.1.1,1.1.1.10,1.1.1.100,1.1.1.101,1.1.1.102`

- Database for mining = enzyme(リンク元のデータベース名)
- Filtering on LinkDB
  - Destination = compound:* (リンク先は compound 全体)
  - Type = original (逆向きのリンク)
  - Link Path (リンクパスの指定なし)
- Cutoff values
  - Support = 4 entries (全入力エントリに対する支持度 66.6 %)
  - Confidence = 80 %

この条件下でデータマイニングを行った結果を図 5.8に示す。以下では得られたデータマイニングの計算結果についていくつか説明する。

(Sup: 4 entries,Conf:80.0 %)

TARGET_ENTRIES → compound_formula:C21H30N7O17P3

このルールは入力したエントリ集合について支持度 4 entries、確信度 80 %で compound(代謝化合物データベース) の formula フィールドの化学式 C21H30N7O17P3 に関係があることを示している。

(Sup: 4 entries,Conf:100.0 %)

compound_formula:C21H30N7O17P3,TARGET → compound_formula:C21H28N7O17P3

このルールは入力したエントリ集合のうち化学式 C21H30N7O17P3 を持つものに関して、4 entries、確信度 100 %で化学式 C21H30N7O17P3 に関係があることを示している。この他にも 12 の相関ルールが結果として出力されている。

次に、フィールド内の情報同士に関するデータマイニング (CONTENT-CONTENT Data Mining) について説明する。図 5.9 は、CONTENT-CONTENT Data Mining の入力画面である。CONTENT-CONTENT Data Mining の各入力ボタンおよび入力フォームについては 5.2節の Entry-Entry Data Mining を参照されたい。ただし、Cutoff values の Support(最小支持度) に関しては、入力フォームを entries から%に変更している。なぜなら図 5.6で述べているように LinkDB の情報をもとに ENTRY-ENTRY Data Mining 用のデータを加工した場合、入力したエントリ集合もリンク先のエントリ群も両方フィールド単位に加工してしまうので、トランザクション ID がエントリ数からフィールド数に変化する。そのため入力したエントリ数に対しての最小支持度を入力してもデータ数が変化してしまうので効果が得られない。これを回避するために最小支持度を%にした。

以下では具体例を用いて説明する。ここでは酵素反応に関するデータベース enzyme のエントリ５つに関してデータマイニングを行うとする。データマイニングを実行する際の各オプションの条件は以下の通りである。

- 入力エントリ (TARGET)

  `1.1.1.1,1.1.1.10,1.1.1.100,1.1.1.101,1.1.1.102`

- Database for mining = enzyme(リンク元のデータベース名)
- Filtering on LinkDB
  - Destination = compound:* (リンク先は compound 全体)

図 5.9: フィールド内の情報同士に関するデータマイニングの入力画面

  - Type = original (逆向きのリンク)
  - Link Path (リンクパスの指定なし)

- Cutoff values
  - Support = 80 %
  - Confidence = 80 %

以上の条件下でデータマイニングを行った結果を図 5.10に示す。以下では得られたデータマイニングの計算結果についていくつか説明する。

(Sup: 95.5 % ,Conf:100.0 %)

compound_formula:C21H28N7O17P3 → compound_formula:C21H30N7O17P3

このルールは入力したエントリ集合５つをフィールド単位に分解した場合、分解した全フィールドの 95.5 %について compound_formula:C21H28N7O17P3 および compound_formula:C21H30N7O17P3 が含まれており、確信度 100 %でこのルールが成立すること、すなわち compound_formula:C21H28N7O17P3 が必ず compound_formula:C21H30N7O17P3 を包含することを表している。他にも 3 つの相関ルールが結果として出力されている。

図 5.10: フィールド内の情報同士に関するデータマイニングの計算結果

5.3 縮合型データマイニング

前節までに構築したシステムのパフォーマンスを調べた結果、従来ビジネスデータを対象としたデータマイニングでは考えられなかったような新しい問題、すなわち「同じビットパターンを持つアイテムが多数存在する場合に相関ルール生成処理が爆発する」という問題が見つかった。これは、通常のビジネスデータに見られるように、相関ルール発見に使用するビットテーブルが一般に縦長（つまりアイテム数に比べて顧客の数が圧倒的に多い）という状況では殆んど生じない。しかし、本研究のように、利用者が興味を持っている一部のデータだけを切り出してマイニングを行なうという枠組では、ゲノムデータベースの偏り、すなわち同じようなデータが多数登録されていることも手伝って、切り出したデータの中に同じビットパターンを持つ似たようなアイテムが多数存在することが多い。

この問題を解決するために、ユーザが入力したデータを加工する段階で同じようなビットパターンを持つアイテム集合を１つのマクロアイテム (チャンク) として扱い、計算を行うことを考えた。このようにアイテム集合をあたかも１つのアイテムのように縮合してデータマイニングを行うことを本論文では**縮合型データマイニング**と呼ぶことにする。

次に、縮合型データマイニングの有意性について述べる。各トランザクションは図 5.1 のようなアイテムを持つとする。ただし、各トランザクションがアイテムでは 2 進値で表し、アイテムを持っている場合は 1、そうでない場合は 0 とする。縮合によりアイテム A

縮合前のトランザクションデータベース

Transaction ID	アイテム 1	アイテム 2	アイテム 3	アイテム 4	アイテム 5
100	1	1	1	1	1
200	1	1	1	0	0
300	1	1	1	1	1
400	0	0	0	1	1

⇓

縮合後のトランザクションデータベース

Transaction ID	アイテム A	アイテム B
100	1	1
200	1	0
300	1	1
400	0	1

表 5.1: アイテム縮合の例

は { アイテム 1, アイテム 2, アイテム 3}、アイテム B は { アイテム 4, アイテム 5} となる。これにより 5 個あったアイテムは 2 個のマクロアイテムに縮合される。次にアイテムを縮合していない場合と縮合を行った場合の相関ルールの生成数について述べる。最小支持度 50 %、最小確信度を 75 %と設定した場合、以下のように生成される相関ルールの点で違いが出てくる。

- **縮合前**：
  閾値を満たすラージアイテム集合は {1}、{2}、{3}、{4}、{5}、{1,2}、{1,3}、{1,4}、{1,5}、{2,3}、{2,4}、{2,5}、{3,4}、{3,5}、{1,2,3}、{1,2,4}、{1,2,5}、{1,3,4}、{1,3,5}、{1,4,5}、{2,3,4}、{2,3,5}、{2,4,5}、{3,4,5}、{1,2,3,4}、{1,2,3,5}、{1,3,4,5}、{2,3,4,5} となる。これらのラージアイテムについてさらに最小確信度の閾値を満たしている組み合わせは $\{1\} \Rightarrow \{2\}$、$\{2\} \Rightarrow \{1\}$、$\{1\} \Rightarrow \{3\}$、$\{3\} \Rightarrow \{1\}$、$\{1\} \Rightarrow \{4\}$、$\{4\} \Rightarrow \{1\}$、$\{1\} \Rightarrow \{5\}$、$\{5\} \Rightarrow \{1\}$、$\cdots$、$\{2,3\} \Rightarrow \{4,5\}$、$\{2,4\} \Rightarrow \{3,5\}$、$\{2,5\} \Rightarrow \{3,4\}$、$\{2,3,4\} \Rightarrow \{5\}$、$\{2,3,5\} \Rightarrow \{4\}$ の合計 124 となる。

- **縮合後**：
  閾値を満たすラージアイテム集合は {A}、{B}、{A,B} となる。これらラージア

イテム集合についてさらに最小確信度の条件を満たしている組合わせは $\{A\} \Rightarrow \{B\}$, $\{B\} \Rightarrow \{A\}$ となる。ここで縮合前のアイテム集合に変換すると A=$\{1,2,3\}$、B=$\{4,5\}$、{A,B}=$\{\{1,2,3\},\{4,5\}\}$ となる。つまり最終的に $\{1,2,3\} \Rightarrow \{4,5\}$ および $\{4,5\} \Rightarrow \{1,2,3\}$ の 2 つになる。

アイテム縮合により $\{1,2\} \Rightarrow \{3,4,5\}$、$\{1,2,3,4\} \Rightarrow \{5\}$ などの冗長な相関ルールは全て $\{1,2,3\} \Rightarrow \{4,5\}$ に統合された。ここでそれぞれのルールが意味的に同じであることを示す。

- $\{1,2\} \Rightarrow \{3,4,5\}$ と $\{1,2,3\} \Rightarrow \{4,5\}$ の場合、統合ルール内のアイテム３がヘッド側のマクロアイテム $\{3,4,5\}$ に移動している。$\{1,2\}$ と $\{1,2,3\}$ は同じビットベクターをもつがゆえに支持度が等しく、ヘッドだけが変化する。与えられたアイテムのサポートを S(X) とすると、$\{1,2\} \wedge \{3,4,5\}$ の支持度は $S(\{1,2\} \wedge \{3,4,5\})$ となり $\{1,2,3\} \wedge \{4,5\}$ の支持度は $S(\{1,2,3\} \wedge \{4,5\})$ となる。$S(\{1,2\} \wedge \{3,4,5\}) = S(\{1,2,3\} \wedge \{4,5\})$ なので、結局２つのルールは支持度および確信度の点で全く同じであると言える。

- $\{1,2\} \Rightarrow \{3,4,5\}$ と $\{1\} \Rightarrow \{2,3,4,5\}$ の場合も同様にヘッドとボディが両方成立する範囲は同一である。しかし、この場合は統合されるルールのヘッドにあったアイテム 4 がボディのマクロアイテム $\{1,2,3,4\}$ に移動したことにより、２つのルールに関してボディの支持度が異なる。そのため、二つのルールの支持度は等しくならない。だが、４と５が同じビットベクターを持つことを考えると、二つのルールの間に意味的な差はないと言える。

次に実際のゲノムデータを用いたエントリ間データマイニングに関して、冗長性を考慮した縮合型データマイニングの計算時間を比較し、縮合型データマイニングの有用性について示す。実験に使用したデータは pdb:101M,pdb:102L,pdb:102M,pdb:103M, pdb:103L,pdb:10MH の６つのエントリを対象とした。実験条件は以下の通りである。

- リンク先のデータは制限しない。
- 最小支持度 (Support) は 2,3,4 entries
- 最小確信度 (Confidence) は 30 - 70 %の間を 10 %間隔で設定する。

実験環境は、Sun Microsystem 社の Ultra Enterprise 10000, 64cpu, Main memory 16.0GB 上で行った。ルール生成数は、実験条件として決めた最小支持度および最小確信度の閾値を越えた相関ルールの数の総数。計測時間は、閾値を越えている相関ルールをすべて計算するのに要した時間で実験回数は 4 回行い 2 回目から 4 回目までの行った計算時間の平均

を表している。図 5.11は、最小支持度 2、最小確信度 30 %から 70 %までのルール生成を表している。冗長性を考慮しない場合では、最小支持度 30 %ではルールの生成数が大量であったために、実験に使用した計算機ではメモリ不足になり結果を計算することができなかった。40 %および 50 %に関しては、それぞれ約 33,000 の相関ルールが抽出された。60 %および 70 %に関しては、閾値を越えたルールが存在しなかった。冗長性を考慮した場合は、いずれの場合もルールは 1 つに縮合された。図 5.12は、最小支持度 2、最小確信度 30 %から 70 %までの計算時間を表している。冗長性を考慮しない場合では、最小確信度 30 %の時は計算不能であった。40 %および 50 %に関しては、それぞれ約 380 秒程度を計算に要した。60 %および 70 %に関しては、閾値を越えたルールが存在しなかった。冗長性を考慮した場合は、いずれの場合もルール生成に要した時間は 1 秒以下であった。図 5.13は、最小支持度 3、最小確信度 30 %および 70 %までのルール生成を表している。冗長性を考慮しない場合では、40 %および 50 %に関しては、それぞれ約 17,000 の相関ルールが抽出された。60 %および 70 %に関しては、閾値を越えたルールが存在しなかった。冗長性を考慮した場合は、いずれの場合もルールは 1 つに縮合された。図 5.14は、最小支持度 3、最小確信度 30 %および 70 %までのルール生成数および計算時間である。40 %および 50 %に関しては、それぞれ 380 秒程度を計算に要した。60 %から 70 %に関しては、閾値を越えたルールが存在しなかった。冗長性を考慮した場合は、いずれの場合もルール生成に要した時間は 1 秒以下であった。

計算結果の考察について考察した結果を以下に示す。

- **縮合を行わない場合：** 最小支持度 2、最小確信度 40 %および 50 %の場合、ラージアイテム 2 を生成した時に閾値を越えたアイテムが 6 存在した。このアイテムは、最初に指定したエントリ 5 つのうち 3 つのエントリ集合に存在した。そのためラージアイテム集合のサイズが大きくなっても、全ての閾値を越えてしまいルールが生成された。同様のことが最小支持度 3、最小確信度 30,40,50 %の時にも言える。

- **縮合を行う場合：** 最小支持度 2、最小確信度 40 %および 50 %の場合、ラージアイテムを計算する前に、各エントリで同じアイテムを持つものを１つのアイテムとして縮合を行った。その結果、ラージアイテム 2 を生成する段階で１つのアイテムとして計算するので、ラージアイテム 3 以降計算が行われない。同様な事が、最小支持度 3、最小確信度 30,40,50 %の時にも言える。

尚、詳しい計算結果は付録として添付した。

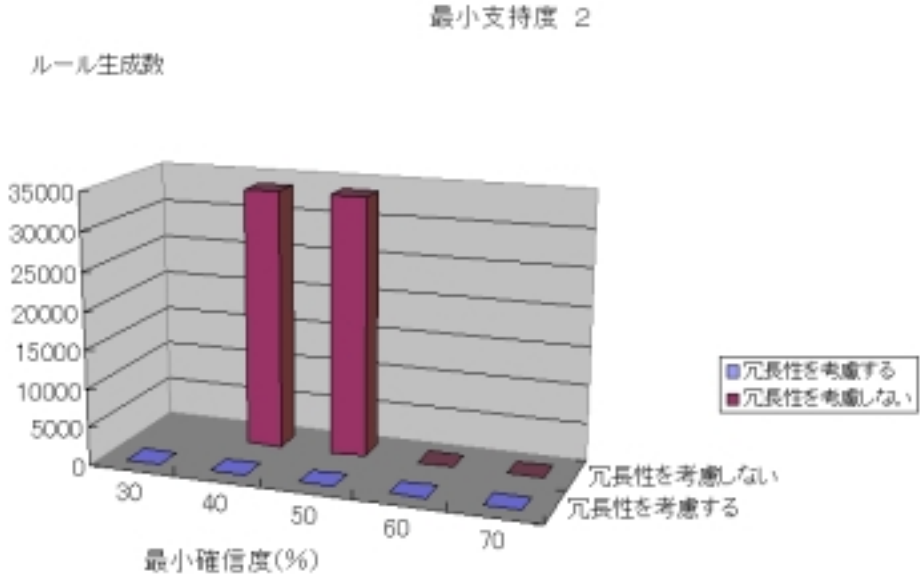

図 5.11: 最小支持度 2 におけるルール生成数比較

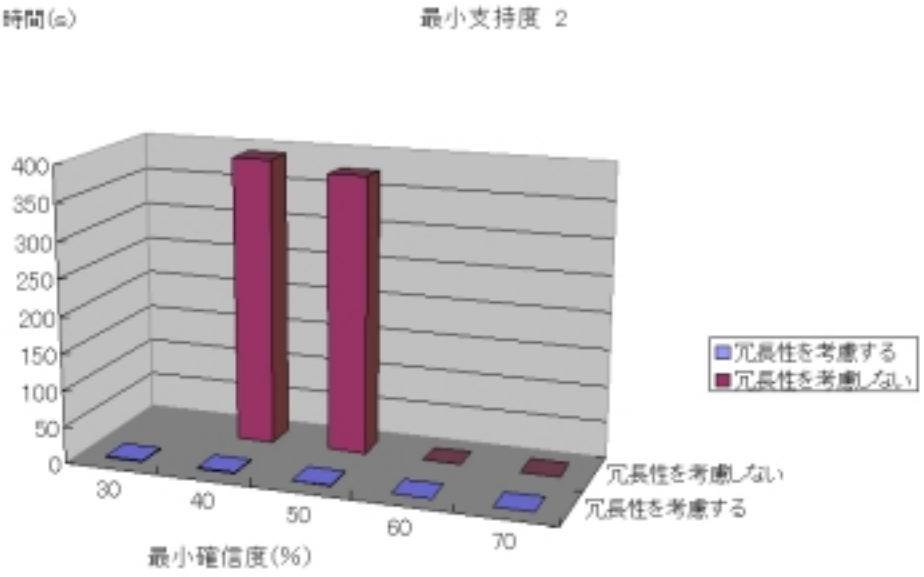

図 5.12: 最小支持度 2 における計測時間比較

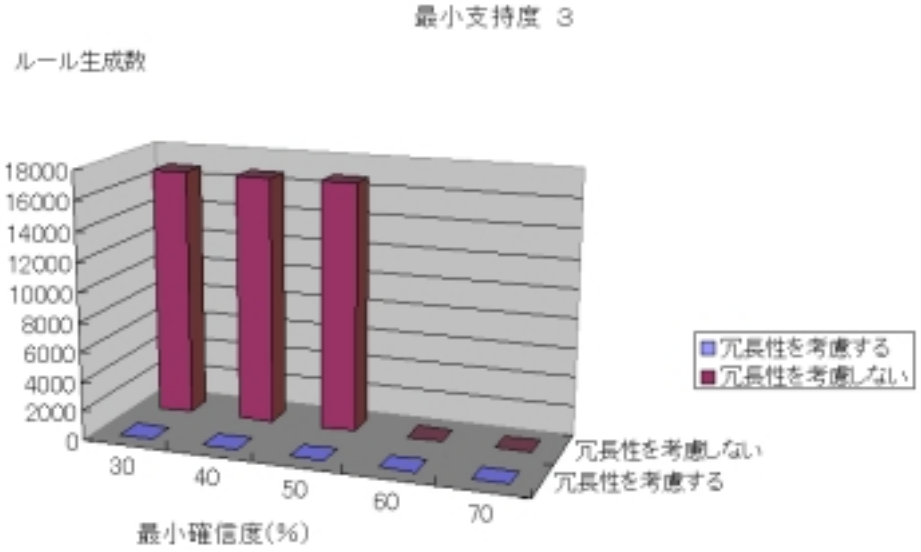

図 5.13: 最小支持度 3 におけるルール生成数比較

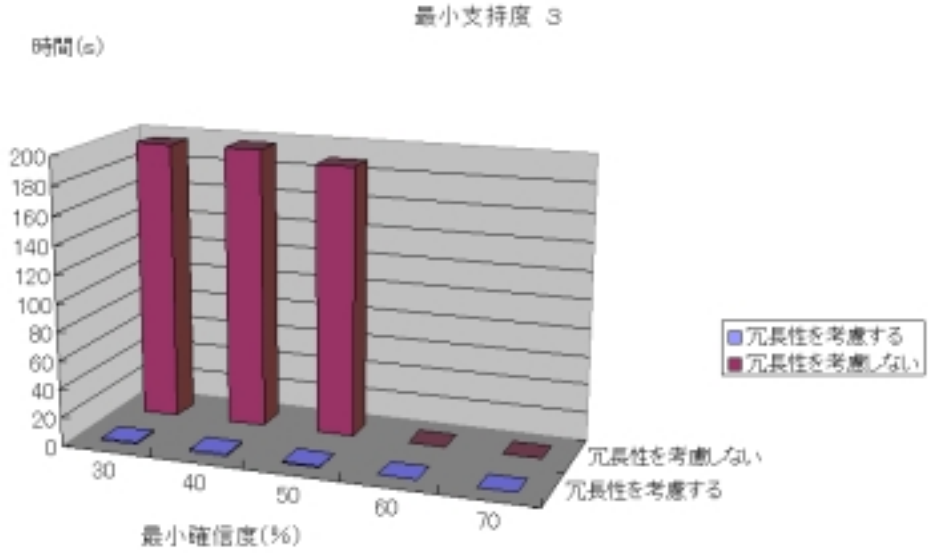

図 5.14: 最小支持度 3 における計測時間比較

第 6 章

おわりに

6.1 まとめ

本研究では、科学技術データベースの一種であるゲノムデータベースから柔軟にデータ加工および知識発見を行なうための枠組について検討を行ない、システム構築を行なった。構築したシステムはゲノムネット上で更新されているゲノムデータベース全体を対象としており、その間の参照関係である最新のクロスリファレンス情報（LinkDB）を基本的な情報源として相関ルール発見を行なう、という新しい枠組を導入している。ゲノム情報処理の分野では従来、固定的な少数のデータに対し、計算機科学者主導でその都度データマイニングを行なっていたが、本システムを利用することにより、計算機科学に詳しくない生物学や医学の研究者でも、特別な知識なしに、最新のゲノムデータベース全体を対象とした知識発見を Web ブラウザ上で行なうことが可能になった。また、データベース選択やマイニングを行なうレベル（エントリレベルかコンテントレベルか）に関して利用者の興味や指示を柔軟に反映し、マイニング用のデータ空間を動的に絞り込むことによって、1000 万を超えるデータ要素から高速にマイニングを行なうことができた。

さらに、新しい枠組を導入した事により、従来ビジネスデータを対象としたデータマイニングでは考えられなかったような新しい問題、すなわち「同じビットパターンを持つアイテムが多数存在する場合に相関ルール生成処理が爆発する」という問題が見つかった。これは、通常のビジネスデータに見られるように、相関ルール発見に使用するビットテーブルが一般に縦長（つまりアイテム数に比べて顧客の数が圧倒的に多い）という状況では殆んど生じない。しかし、本研究のように、利用者が興味を持っている一部のデータだけを切り出してマイニングを行なうという枠組では、ゲノムデータベースの偏り、すなわち同じようなデータが多数登録されていることも手伝って、切り出したデータの中に同じ

ビットパターンを持つ似たようなアイテムが多数存在することが多い。これを解決するために、同じビットパターンを持つアイテムをひとまとめにしてから処理を行なう縮合型データマイニングを考案した。検討の結果、縮合により冗長なルールだけが抑制され、かつ、計算の爆発が回避できることが分かった。縮合の効果として、相関ルール生成にかかる処理時間が劇的に削減され、いくつかの例では 1 秒もしくはそれ以下の時間で必要な相関ルール発見を行なうことができた。これは Web 経由でも十分サービスできる程の高速な応答であり、生物学者や医学者に対しブラウザ経由でマイニングサービスを提供するという本研究の目的にかなう結果が得られた。

6.2 今後の展望

本研究ではゲノムデータベース全体からデータマイニングを行なうシステムについて基礎的な部分を構築したが、検討はしたものの時間の関係で実装に到らなかった部分もいくつかある。

- 本研究で導入した枠組の一部、すなわち、エントリレベルのマイニングで言えば「利用者が興味を持つエントリ集合をシステムに与え、それに従って関連する知識だけを高速に発見する」という仮定は、例えば「ホモロジー検索でヒットした上位 20 個のエントリに共通かつ特異的な事実を調べたい」というような実用的な状況設定を元に考案した。しかし、現在の本システムはホモロジー検索など外部の解析サービスと連動するまでには到っておらず、利便性の点で問題がある。これについては今後、ゲノムネットの各種解析サービスと連動していく予定である。

- コンテントレベルのデータマイニングに関しては、本研究ではよく使われると思われる一部のコンテントのみ切り出して実験を行なった。各データベースのエントリに含まれる情報の詳細な調査と分類を行ない、利用価値の高いコンテントに関しては将来的には全て切り出しを行なう必要がある。

- 利用者の指示に従って最新のデータ空間を動的に絞り込み、高速なデータマイニングを行なうことは実現できたが、LinkDB のクロスリファレンス情報にしろ、詳細なレベルのデータマイニングに使用するコンテント情報（エントリから切り出した情報）にしろ、予めシステム上に存在するデータであり、絞り込みを行なうということを除けば、更新こそされるものの固定的なデータであることには変わりない。一方、ゲノムデータに対するマイニングの大きな需要の一つに、「利用者が用意した

か、もしくは質問時に動的に利用者が合成したデータを用いて、マイニングを行ないたい」というものがある。例えば利用者の実験室で新たに得られた結果を使いたいとか、ホモロジー検索やモチーフ検索などの解析結果を使ったデータマイニングを行ないたいというのが典型的なケースとして存在する。このような場合にはシステム側で予めデータ空間を用意する事が不可能であり、むしろ一定の形式で記述された利用者側のデータを受け入れる仕組みや、利用者の指示に従って質問時にシステム側で解析プログラムを走らせ、その結果をマイニング用のデータ空間として加工する仕組みが必要になる。後者については、例えば広く用いられている配列解析パッケージである GCG と本システムと連動させるなどの方法が考えられる。これにより、本当の意味で動的かつ合成的なデータマイニング、すなわち Dynamic and Synthetic Data Mining が可能になる。

今後は、本研究でのシステム構築から得られた知見を元に上記の課題を解決し、データの加工や縮合および動的な合成に関する理論的検討 [25] を行なう事により、大規模な科学データベースからの知識発見手法を確立する事を目指していきたい。

謝辞

本研究を進めるにあたり、適切な御指導、御助言を頂きました佐藤 賢二助教授には深く感謝いたします。

遺伝子システム論講座 小長谷教授には、本研究について多大な御指導と御助言をして頂いて、私を導いて下さいました。ここに感謝の意を表し、心より御礼を申し上げます。

また、東京大学医科学研究所 ヒトゲノム解析センター計算機室には、こころよく計算機の利用をお許しいただきました。そして、遺伝子システム論座の同輩、後輩諸氏には良き相談相手となり励ましいただいたことには感謝いたします。

参考文献

[1] 小長谷 明彦: 遺伝子とコンピュータ, 共立出版, 2000.

[2] 松原 謙一, 中村 桂子: ゲノムを読む, 紀伊国屋書店, 1996.

[3] 金久 實 編: ヒューマンゲノム計画, 共立出版, 1997.

[4] 阿久津 達也, 麻生川 稔, 小長谷 明彦: 分子生物情報学の現状と動向, 人工知能学会, Vol. 15, No. 1, pp.3-10, 2000.

[5] 西尾 章治郎: 大規模データベースにおける知識獲得, 情報処理学会, Vol. 34, No. 3, pp.343-350, 1993.

[6] 河野 浩之, 西尾 章治郎, Jiawei Han: データベースからの知識獲得技術, 人工知能学会, Vol. 10, No. 1, pp.38-44, 1994.

[7] Pieter Adriaans, Dolf Zaninge, 山本 英子, 梅村 恭司 訳: データマイニング, 共立出版, 1998.

[8] 福田 剛志, 森本 康彦, 森下真一, 徳山 豪: 特別論説 情報処理最前線 データマイニングの最新動向 -巨大データからの知識発見術 -, 情報処理学会, Vol. 37, No. 7, pp.597-603, July 1996.

[9] データ・ウェアハウス最前線 仮説・検証から発見へ, Sun World, Oct, 1999.

[10] マイケル J.A. ベリー, ゴードン・リノフ: SAS インスティチュート ジャパン/江原 淳, 佐藤 栄作 共訳:データマイニング手法 海文堂, 1999.

[11] Agrawal, R., Imielinski, T. and Swami, A.:Database Mining: A Performance Perspective, IEEE Trans.on Knowledge and Data Engineering, Vol.5, No.6, pp.914-925, 1993.

[12] 森下 真一 データマイニングシステムの概念・理論・応用, 第 15 回大会併設チュートリアル データマイニングの実装と応用, 日本ソフトウェア科学会, 1998.

[13] Agrawal, R. and Srikant, R.: Fast Algorithms for Mininig Association Rules, Proc. of VLDB, pp.487-499, 1994.

[14] 喜連川 優: データマイニングにおける相関ルール抽出技法, 人工知能学会誌, pp.513-520, 1997.

[15] Agrawal, R., Imielinski, T. and Swami, A.: Mining Association Rules between Sets of Items in Large Databases, Proc. of ACM SIGMOD, pp.207-216, 1993.

[16] 金久 實: ゲノム情報への招待, 共立出版, 1996.

[17] 金久 實: ゲノム情報学, 情報処理学会, Vol. 37, No. 10, pp.908-913, 1996.

[18] 高木 利久: ゲノムデータベース -意義、歴史、課題 -, 数理科学, No. 432, pp.19-25, 1999.

[19] 連載 ゲノムデータベース, コンピュータサイエンス誌 bit,Vol. 31,No. 8, AUG.pp.28-33, 1999.

[20] 連載 ゲノムデータベース, コンピュータサイエンス誌 bit,Vol. 31,No. 10, OCT.pp.76-83, 1999.

[21] 高木 利久・金久 實 編: ゲノムネットのデータベース利用法 [第 2 版], 共立出版, 1998.

[22] 星田 昌紀編著: 遺伝子情報処理への挑戦, 共立出版, 1994.

[23] Satou,K., Shibayama,G., Ono,T., Yamamura,Y., Furuichi,E., Kuhara,S., and Takagi,T.: Finding Association Rules on Heterogeneous Genome Data,Proc. of the Pacific Symposium on Biocomputing '97 (PSB'97), pp.397-408, Jan. 1997

[24] Eleanor Lawrence 編, 荒木 忠雄, 清水 碩, 藤森 嶺 監訳: ヘンダーソン生物学用語事典, オーム社, 1996.

[25] Liu, H and Motoda, H eds: FEATURE EXTRACTION, CONSTRUCTION AND SELECTION: A Data Mining Perspective, Kluwer Academic Publishers, 1998.

研究業績

Yoshiki Fuseda and Kenji Satou: Toward a Data Mining Service from Large and Heterogeneous Genome Databases in GenomeNet, Genome Informatics 1999, UNIVERSAL ACADEMY PRESS,INC. TOKYO, JAPAN.

第 A 章

付録

5.3 節で比較実験を行ったデータの詳細を以下に示す。ただし表中の略記は、次のような意味である。

- LI 理論値は、初期アイテム数に対する理論上のラージアイテム生成数
- LI 実測値は、初期アイテム数に対する実際に生成されたラージアイテム数
- RU 理論値は、初期アイテム数に対する理論上の相関ルール生成数
- RU 実測値は、初期アイテム数に対する実際に生成されたルール数
- DM TIME はラージアイテム生成に要した時間およびルール生成に要した時間を表す。ただし、表中で計測時間が、1s と書いてあるのは計算時間が、1s 以下であることを表し、1s 以上は 4 回計測したうちの 2 回目から 4 回目までの平均時間を表している。
- また、計測不能と書いてあるのは、今回実験で使用したマシンがメモリ不足となり計算することができなかったことを表す。

	初期アイテム数	LI 理論値/LI 実測値	RU 理論値/LI 実測値	DM TIME
冗長性を考慮する	9	511 / 2	986409 / 2	1s
冗長性を考慮しない	56	$72 * 10^{15}$/計測不能	$19 * 10^{75}$/計測不能	計測不能

表 A.1: 設定：最小確信度 2、最小支持度 30 %

	初期アイテム数	LI 理論値/LI 実測値	RU 理論値/LI 実測値	DM TIME
冗長性を考慮する	9	511 / 2	986409 / 2	1s
冗長性を考慮しない	56	$72 * 10^{15}$ / 127	$19 * 10^{75}$ / 33590	391s

表 A.2: 設定：最小確信度 2、最小支持度 40 %

	初期アイテム数	LI 理論値/LI 実測値	RU 理論値/LI 実測値	DM TIME
冗長性を考慮する	9	511 / 2	986409 / 2	1s
冗長性を考慮しない	56	$72 * 10^{15}$ / 127	$19 * 10^{75}$ / 33590	377s

表 A.3: 設定：最小確信度 2、最小支持度 50 %

	初期アイテム数	LI 理論値/LI 実測値	RU 理論値/LI 実測値	DM TIME
冗長性を考慮する	9	511 / 0	986409 / 0	1s
冗長性を考慮しない	56	$72 * 10^{15}$ / 0	$19 * 10^{75}$ / 0	1s

表 A.4: 設定：最小確信度 2、最小支持度 60 %

	初期アイテム数	LI 理論値/LI 実測値	RU 理論値/LI 実測値	DM TIME
冗長性を考慮する	9	511 / 0	986409 / 0	1s
冗長性を考慮しない	56	$72 * 10^{15}$ / 0	$19 * 10^{75}$ / 0	1s

表 A.5: 設定：最小確信度 2、最小支持度 70 %

	初期アイテム数	LI 理論値/LI 実測値	RU 理論値/LI 実測値	DM TIME
冗長性を考慮する	9	511 /1	986409 /1	1s
冗長性を考慮しない	56	$72 * 10^{15}$ /127	$19 * 10^{75}$ /16795	196s

表 A.6: 設定：最小確信度 3、最小支持度 30 %

	初期アイテム数	LI 理論値/LI 実測値	RU 理論値/LI 実測値	DM TIME
冗長性を考慮する	9	511 / 2	986409 / 2	1s
冗長性を考慮しない	56	$72 * 10^{15}$ / 127	$19 * 10^{75}$ / 16795	197s

表 A.7: 設定：最小確信度 3、最小支持度 40 %

	初期アイテム数	LI 理論値/LI 実測値	RU 理論値/LI 実測値	DM TIME
冗長性を考慮する	9	511 / 1	986409 /1	1s
冗長性を考慮しない	56	$72 * 10^{15}$ / 127	$19 * 10^{75}$ / 16795	195s

表 A.8: 設定：最小確信度 3、最小支持度 50 %

	初期アイテム数	LI 理論値/LI 実測値	RU 理論値/LI 実測値	DM TIME
冗長性を考慮する	9	511 / 0	986409 / 0	1s
冗長性を考慮しない	56	$72 * 10^{15}$ / 0	$19 * 10^{75}$ / 0	1s

表 A.9: 設定：最小確信度 3、最小支持度 60 %

	初期アイテム数	LI 理論値/LI 実測値	RU 理論値/LI 実測値	DM TIME
冗長性を考慮する	9	511 / 0	986409 / 0	1s
冗長性を考慮しない	56	$72 * 10^{15}$ / 0	$19 * 10^{75}$ / 0	1s

表 A.10: 設定：最小確信度 3、最小支持度 70 %

	初期アイテム数	LI 理論値/LI 実測値	RU 理論値/LI 実測値	DM TIME
冗長性を考慮する	9	511 / 0	986409 / 0	1s
冗長性を考慮しない	56	$72 * 10^{15}$ / 0	$19 * 10^{75}$ / 0	1s

表 A.11: 設定：最小確信度 4、最小支持度 30 %

	初期アイテム数	LI 理論値/LI 実測値	RU 理論値/LI 実測値	DM TIME
冗長性を考慮する	9	511 / 0	986409 / 0	1s
冗長性を考慮しない	56	$72 * 10^{15}$ / 0	$19 * 10^{75}$ / 0	1s

表 A.12: 設定：最小確信度 4、最小支持度 40 %

	初期アイテム数	LI 理論値/LI 実測値	RU 理論値/LI 実測値	DM TIME
冗長性を考慮する	9	511 / 0	986409 / 0	1s
冗長性を考慮しない	56	$72 * 10^{15}$ / 0	$19 * 10^{75}$ / 0	1s

表 A.13: 設定：最小確信度 4、最小支持度 50 %

	初期アイテム数	LI 理論値/LI 実測値	RU 理論値/LI 実測値	DM TIME
冗長性を考慮する	9	511 / 0	986409 / 0	1s
冗長性を考慮しない	56	$72 * 10^{15}$ / 0	$19 * 10^{75}$ / 0	1s

表 A.14: 設定：最小確信度 4、最小支持度 60 %

	初期アイテム数	LI 理論値/LI 実測値	RU 理論値/LI 実測値	DM TIME
冗長性を考慮する	9	511 / 0	986409 / 0	1s
冗長性を考慮しない	56	$72 * 10^{15}$ / 0	$19 * 10^{75}$ / 0	1s

表 A.15: 設定：最小確信度 4、最小支持度 70 %