情 報 学 基 礎 27－7
（1992. 9. 8）

オブジェクト指向データベースを用いた遺伝子データの管理

坂本憲広† 五斗進‡ 高木利久*

†九州大学医学部

‡九州大学工学部情報工学科

*東京大学医科学研究所ヒトゲノム解析センター

我々は、GenBankの塩基配列データを管理するためのオブジェクト指向データベースを開発した。このデータベースは商用のオブジェクト指向データベースであるVersantを用いており、GenBankのエントリは複合オブジェクトとしてデータベース中に格納されている。ユーザはルールの形で質問を与えることにより、データベース中のゲノム情報を検索することができる。このシステムを用いることにより、単純なキーワードの検索から塩基配列のホモロジー検索まで、様々な種類の検索を行うことができる。

Management of Genome Data by an Object-Oriented Database

Norihiro Sakamoto† Susumu Goto‡ Toshihisa Takagi*

†Kyushu University Medical School

‡Department of Computer Science and Communication Engineering,
Faculty of Engineering, Kyushu University

*Human Genome Center, Institute of Medical Science, University of Tokyo

We have developed an object-oriented database system for storing GenBank nucleotide sequence data. The database is constructed on a commercially available object-oriented database system Versant and contains GenBank entries as complex objects. Logic-based rules and queries are available for retrieving various genome information from it. This system can be used for simple keyword searches, homology searches for nucleotide sequences and their combinations.

1. はじめに

現在ヒト遺伝子のデータは約15000エントリ、塩基数にして約1700万塩基が報告されている（GenBank release 72. 1992年6月）。GenBankではこれまでこれらのデータをフラットファイルの形式で配布してきた。また、これらのデータを解析、利用するための応用プログラムも数多く開発されている。一方、より効率よくデータの管理、利用を行なうために、遺伝子データのデータベース化も進んでおり、最近ではGenBankも内部的には関係データベースシステムによって管理されている[1]。さらに配布データについてもフラットファイルに加えて、ASN.1フォーマットによる配布も開始された[2]。我々はこれまで、ヒト遺伝子データを関係データベース化し、さらに演繹エンジンを応用し、高度な質問処理機能を備えたシステムODSを開発してきた[3]。

関係データベースでは、データは正規化され、テーブルの形で管理される。しかし、遺伝子データは実験により得られた情報をできるだけ忠実に保存することが望ましく、そのため、定型化が困難な場合が多い。一方、オブジェクト指向データベースでは、必要に応じて様々なオブジェクトを定義することができ、遺伝子データの管理に適していると考えられる。

2. 関係データベースによる管理の問題点

図1は関係データベース化した遺伝子データの例である。例えば、遺伝子データの中心となる塩基配列情報は、エントリ間で数塩基から数十万塩基までと非常にバラツキが大きく、これをそのまま関係データベースに格納することは難い。ODSでは塩基配列情報を長さ8のOverlapping Oligonucleotideに変形することによって、これを解決している（図1：テーブルSequences）。仮に塩基配列情報を関係データベースにテキスト型として格納したとすれば、それらに対するSQLなどの組み込みのデータベース問い合わせ言語を用いた検索方法は非常に限られたものとなる。そのため、塩基配列を比較するための特別なプログラム（FASTA[4]など）との組合せが必要である。しかし、それを関係データベースシステムで実現しようとすると、問い合わせ言語とプログラミング言語の不整合（インピーダンス・ミスマッチ）という問題が生じる。

テーブルKeyword

Locus	Keyword
AGMPPINS	preproinsulin
⋮	⋮

テーブルFeatures

Locus	Start	End	Signal
AGMPPINS	426	463	exon
AGMPPINS	654	857	sig_peptide
AGMPPINS	671	742	exon
⋮	⋮	⋮	⋮

テーブルSequences

Locus	Sequence
AGMPPINS	GGGCCATC
AGMPPINS	GGCCATCC
⋮	⋮

図1：関係データベース化した遺伝子データの例

また、遺伝子の生物学的特徴を記述したFeature Tableも構造が複雑で、正規化は容易ではない。ODSではこのFeature Tableを一つの関係テーブルに格納し、属性Locusをキーとして遺伝子データの他の部分との関連を実現している（図1：テーブルFeatures）。しかし、Feature Tableの情報はフォーマットが複雑なためいくつかの情報は関係テーブルから削除している。

遺伝子データの研究では、関係データベースでは扱いにくいこれらの塩基配列情報とFeature Tableを非常によく参照利用する。しかし、ディスクスペースを効率よく使うためには、これらを別々の関係テーブルに格納しなければならないため、実際の利用時に頻繁に結合操作を強いられ、検索時間がかかってしまう。

オブジェクト指向データベースでは、一つのオブジェクトからそれに関連したオブジェクトの参照は非常に容易であり、あるエントリの塩基配列

の一部分とそれの持つ生物学的特徴の対応が自然にかつ迅速に行なえる。さらに、遺伝子データそのものの性質により、ある遺伝子エントリには必ずそれと関連したエントリが存在する。つまりゲノム構造上では、ある遺伝子エントリにはその上流のエントリと下流のエントリが物理的かつ意味的に強く関連している。また、遺伝情報の観点からは、あるDNAのエントリにはそれに由来するmRNAのエントリが存在することがあり、さらに対応するタンパク質のデータが存在することになる。そのため、遺伝子データベースではある遺伝子エントリの情報の1つとして別の遺伝子エントリを含むことのできるような構造が望まれる。

3. 遺伝子データのオブジェクト化

オブジェクト指向データベースは、いくつかのオブジェクトとそれらを結び付ける関連とによって構成され、オブジェクトは属性とメソッドよりなる。オブジェクトの属性は関係データベースにおけるカラムに対応して考えることができる。図2から図4は遺伝子データに含まれるオブジェクトとその関係を示したものであり、長方形はクラスを、矢印は属性とクラスの関連を、実線はクラスの階層関係を表している。各クラスの属性はイタリック体で表している。オブジェクト指向データベースでは、属性値に加えてこれらの関係も格納することができるため、データ構造をうまく設計すれば、検索時の結合操作を行なわずに、必要な関連データを引き出すことができる。

3.1 オブジェクト指向データベースにおける遺伝子データのクラス例

遺伝子データの中心となるオブジェクトは、クラスGenBankEntryのインスタンスである。GenBankEntryは実際の遺伝子データの1つの単位、すなわちGenBank遺伝子データの1エントリに対応するものであり、GenBank遺伝子データに記載されている全てのデータフィールドを属性として含んでいる。GenBank遺伝子データではデータフィールドの値は、数値もしくは文字列の

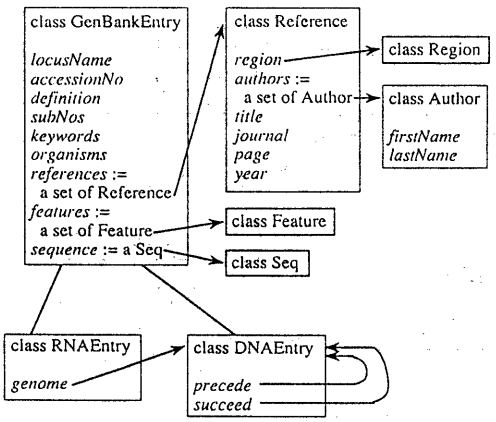

図2：クラスGenBankEntryと関連クラス

いずれかであるが、GenBankEntryではそれらの属性のうちいくつかは、*references*や*features*のように複合オブジェクトとして定義している（図2）。

GenBankEntryの下にはDNAEntryとRNAEntryの2つのサブクラスが定義されており、GenBankEntryに含まれる属性とメソッドはこれらのサブクラスに継承される。継承はオブジェクト指向の重要な特長の1つであり、クラス階層において、あるクラスはそのスーパークラスと同じ性質（属性及びメソッド）を持つことができる。さらにスーパークラスにない特殊化された性質を含むことも可能になる。GenBankEntryでは全ての遺伝子データが共通に持つ属性である、アクセッション番号*accessionNo*、エントリ名*locusName*、エントリの定義*definition*などを定義している。しかし、実際の遺伝子データのエントリはクラスGenBankEntryのインスタンスではなく、その遺伝子データがゲノムDNA由来かmRNA由来かにより、サブクラスDNAEntryのインスタンスかRNAEntryのインスタンスのどちらかに分類する。

DNAEntry、RNAEntryはGenBankEntryのサブクラスとして定義するだけで、GenBankEntryで定義した全ての性質を継承することができる。そこに、それぞれに固有の性質を付け加える

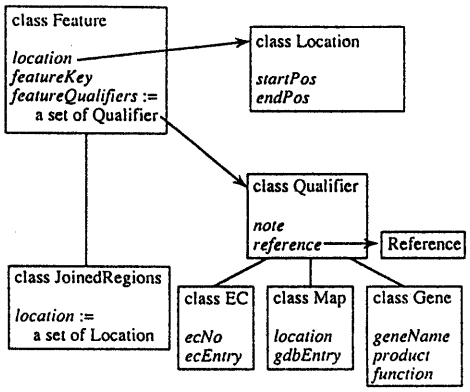

図3：クラスFeatureと関連クラス

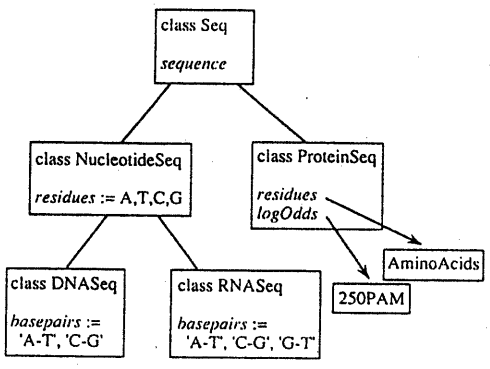

図4：クラスSeqと関連クラス

ことにより、不要な属性を含まず冗長性が除かれるとともに、データの定義域を明確に示すことができる。例えば、DNAEntryであれば、その上流及び下流に存在する（接する）他のDNAEntryを格納する属性として、それぞれ*precede*及び*succeed*が必要であるが、RNAEntryにはそのようなエントリは存在せず、それらの属性は不要であり、意味をなさない。逆に、RNAEntryには、転写元のDNAEntryを示す*genome*という属性がある。

GenBankEntryの属性の*features*はその値にクラスFeatureのインスタンスの集合として定義される。*features*はエントリによっては空集合のこともある。Featureは遺伝子データのうち、非常に重要でかつ複雑な構造を持つ生物学的特徴に関する情報に対応するためのクラスである。Featureは属性として、*location*、*featureKey*、*featureQualifiers*を持つ（図3）。*featureKey*には、文字列型の値が入れられる。*featureQualifiers*は複雑なオブジェクトであるクラスQualifierのインスタンスの集合である。*location*は生物学的特徴の塩基配列上の始点及び終点のデータを保持するクラスLocationのインスタンスを値として持つが、Featureのサブクラスであるクラス JoinedRegionsではLocationのインスタンスの集合を値として持つように再定義されている。

GenBankEntryの*sequence*の値は、クラスNucleotideSeqのインスタンスであるが、その属性は文字列型の*sequence*のみであり、他のデータは格納されていない（図4）。しかし、クラスNucleotideSeqには塩基配列の長さを返すメソッドsizeや塩基配列の構成数(A:C:G:Tの数)を求めるcompositionが定義されており、これらのメソッドを用いて、sequenceのデータから必要な時に計算により値を求めることができる。これらの値はGenBank遺伝子データでは実際の数値データとして記述されているが、クラスNucleotideSeqにはそのメソッドしか定義されていないため格納のためのスペースが節約できる。またオブジェクトに対しては、計算によって求められた値をその属性に納められている実データと同様の手続きで問い合わせることができるため、ユーザーはこの2つを区別する必要はない。

さらにクラスNucleotideSeqには塩基配列同士のホモロジーを検索するためのメソッドhomologyを定義する。これにより、関係データベースではインピーダンス・ミスマッチのため実現困難な検索である、FASTAなどのホモロジー検索プログラムと同等の機能をデータベース自体に持たせることができる。このメソッドによりアクセッション番号やキーワードによる検索と塩基配列のホモロジー検索を組み合わせた質問処理が容易にできる。クラスProteinSeqでは、クラスRNAEntryが持つmRNAが翻訳されてできるタンパク質の

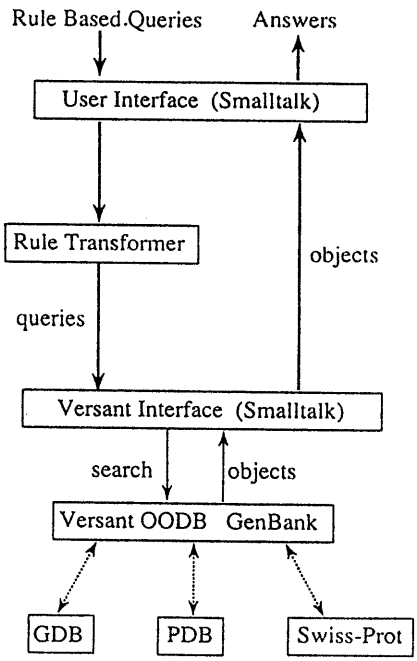

図5：システムの概要

情報を格納する属性を定義する。

3.2 システム

我々が開発しているシステムの概要を図5に示す。これは、商用のVersantオブジェクト指向データベースシステム(version 1.7)及びそのSmalltalkインタフェースを用いている。遺伝子データとしてはGenBank release 72の霊長類データファイルgbpri.seq（約48MB）を用い、データベースへの変換プログラムはObjective Works\Smalltalk R4.0を用いて開発した。データベースの大きさはデータファイル、ログファイル合わせて約130MBであり、元のフラットファイルの約2.7倍のディスク容量を占める。検索の速度は、Featureのオブジェクトを*featureKey*をもとに検索しさらにそのオブジェクトを含むGenBankEntryを検索するのに約15秒要する。

3.3 質問処理

オブジェクト指向データベースではいまだに関係データベースにおけるSQLのような標準的な問い合わせ言語は確立してはいないが、VersantではSQLに似た機能が実現されている。しかし、SQLのような手続き的な記述に比べ、ルールを用いた宣言的な記述の方が、ユーザーの負担が小さく、また記述能力の点でも柔軟性に富み優れていると思われる。特に研究を主眼とした遺伝子データベースでは、仮説の構築、検証が容易で、かつ核酸やタンパク質の高次構造を表現できることが重要である。

ルールを用いた質問を行なう場合、データベースをファクトの集合とみなして操作が行なわれる。関係データベースで管理されたデータでは、テーブルの各行はタプルの形をしており、それをファクトに変換するのは容易であった。一方、オブジェクト指向データベースではデータは配列、リスト、集合などの様々な型を持つため、ファクトの形で管理するのは困難である。またクラス階層と継承によって生じる問題もある。すなわち、あるクラスの属性とそのサブクラスの同じ属性とで、データの型が異なることがある。例えば、Featureの*location*にはLocationのインスタンスが格納されるが、Featureのサブクラスである JoinedRegionsではインスタンスの集合が格納される。このようなオブジェクト指向データベースにおけるデータ構造の多様性は遺伝子データの柔軟な格納には非常に有利であるが、それらのデータに対して問い合わせを行なう場合には大きな障害となり得る。

我々は、ルールを用いた質問の利点をオブジェクト指向データベースにも取り入れるため、ユーザーの質問をVersantの質問に変換するルール変換部を開発中である。このルール変換部が処理するルールは、インスタンスの集合をとる変数を持ち、その集合中の各インスタンスに対して適用する操作を定義することができる。遺伝子データベースはデータ構造の変更を必要とする更新はほとんどないので、現在の固定されたクラス階層の中で再帰を含まないルールを用いた質問の処理を行なう機能を実現している。

```
Seq-entry ::= seq {
  id    {giim    {id 62810},
          genbank {name      "AGMA13GT",
                   accession "M73307"}},

  descr {title   "Cercopithecus aethiops alpha-1,3GT gene, 3' flank.",
          genbank {source    "Cercopithecus aethiops DNA.",
                   keywords {"alpha-1,3-galactosyltransferase"},
                   date      "06-SEP-1991",
                   div       "PRI",
                   taxonomy "Eukaryota; ... Cercopithecinae."},
          org     {taxname "Cercopithecus aethiops"},
          pub     {pub {muid 91334473,
                        gen {serial-number 1},
                        article {title   {name "Gene sequences ... from monkeys"},
                                 authors {names str {"Galili,U.","Swanson,K.W."}},
                                 from    journal {title {iso-jta "Pro.N.A.S."},
                                                  imp {date std {year 1991},
                                                       volume "88",
                                                       pages "7401-7404"}}}}}},

  inst {repr      raw,
        mol       dna,
        length    371,
        strand    ds,
        seq-data iupacna "TTTGAGGTCAAGCCAGAGAA...CTGGAAAAGAAAAATGACAT"},

  annot {{data ftable {{data     gene {locus "alpha-1.3GT"},
                        location int  {from 0, to 370, id giim {id 62810}}},
                       {data     imp  {key "3'UTR"},
                        location int  {from 0,to 370, id giim {id 62810}},
                       qual {{qual "product",
                              val "alpha-1,3-galactosyltransferase"},
                             {qual "gene", val "alpha-1.3GT"}}}}}}
```

図6：ASN.1フォーマットの遺伝子データ例

4. まとめ

遺伝子データベースは実験により導き出されたデータを扱う。それらは、計算機にとって望ましい形態を取っているとは限らない。そのため、多様なデータ型を扱え、また複雑なデータ構造を表現できるオブジェクト指向データベースは遺伝子データの管理に向いていると考えられる。そこで我々はオブジェクト指向データベースVersantを用いた遺伝子データベースを開発した。一方、遺伝子データベースはそれ自体生物学の研究手段でありデータベースに対する高度な質問が容易かつ柔軟に行なえることが望ましい。そのためにはルールを用いた質問処理機能を備える必要があり、我々はそのためのルール変換部を開発中である。

現在の我々のシステムはGenBankのフラットファイルからデータを抽出しているが、今後遺伝子データのASN.1フォーマット（図6）による配布も開始されるため、それを利用して、よりよいデータ構造による遺伝子データの管理が可能になると思われる。

参考文献

[1] Burks,C., Cassidy,M., Cinkosky,M.J., Cumella,K.E., Gilna,P., Hayden,J.E.-D., Keen,G.M., Kelley,T.A., Kelly,M., Kristofferson,D. and Ryals,J.: Genbank. *Nucleic Acids Res.*, **19**,2221–2225, 1991.

[2] National Center for Biotechnology Information: NCBI Software Development Kit Programmer's Reference. 1992.

[3] 坂本憲広、高木利久、佐藤賢二、榊佳之: Development of Overlapping Oligonucleotide Database and its Application to Searching for Signal Sequences over the Human Genome. 1992年情報学シンポジウム, pp.37-46, 1992.

[4] Pearson W.R. and Lipman D.J.: Improved tools for biological sequence comparison. *Proc. Natl. Acad. Sci. U.S.A.*, **85**, 2444–2448, 1988.