ICOT Technical Memorandum: TM-1070

TM-1070

分子生物学のデータベース

田中　秀俊

July, 1991

ICOT
Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan
(03)3456-3191～5
Telex ICOT J32964

Institute for New Generation Computer Technology

分子生物学のデータベース

田中秀俊
(財) 新世代コンピュータ技術開発機構

分子生物学のデータと利用形態の考察から、要求されるデータモデル、特にデータ構造とデータへの操作とについて検討した。これは、分子生物情報の分野のデータベース群を共通の知識表現言語で記述することにより、この分野の統合知識環境を実現するという大目標の一環である。

分子生物学のデータのうち、ゲノム情報ではその特徴記述のためのデータ構造と柔軟に変更できるスキーマ、タンパク質情報では構造や機能のモチーフと化学反応式に関するハイパーグラフの表現が要求項目として挙げられる。これらについてICOTで設計開発中のDOOD言語 *Quixote* で簡単な記述実験を行った。

Molecular Biological Database

Hidetoshi Tanaka

Institute for New Generation Computer Technology (ICOT)

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

In order to build *molecular biological databases*, it is necessary to represent feature descriptions of genomes, motif patterns and reaction expressions of proteins: for example, represent regular expressions for genomic feature descriptions and protein motif patterns, and hypergraphs for reaction expressions.

I have described such data in a DOOD language *Quixote* as an experiment for molecular biological databases, and find it very effective. I will proceed with this work for the furthur evaluation of *Quixote* and the construction of an integrated molecular biological database in it.

1 はじめに

分子生物学という分野は今まさに発展の途上にあり、計算機パワーの有効利用によって研究の飛躍的な発展が期待できる状況にある。この分野の計算機利用の中心は二つある。膨大なデータを解析して知識を得る技術と、そのようなデータや知識を表現し格納する技術である。ここでは格納はむろんのこと、解析に関しても、あいまい検索、解析結果の知識ベース化による解析へのフィードバックといった面などで、データベース / 知識ベース技術が強く求められている。

ICOT では、並列処理と知識利用という観点からデータ解析、中でもタンパク質の配列解析に、知識表現という観点から分子生物学データの表現方法とデータベースの統合に、それぞれ関心を持っている。後者のデータ表現に関しては、非正規関係 DBMS Kappa 上の応用開発評価、および演繹オブジェクト指向データベース言語 *QUIXOTE* の適用実験を行っている。本稿ではこの *QUIXOTE* の適用実験という立場から検討した中間結果を報告する。

2 分子生物学のデータ

分子生物学のデータには大きくゲノム情報とタンパク質の情報の 2 種類がある。現在、ゲノム情報では DNA 配列と地図、タンパク質情報ではアミノ酸配列、機能、立体構造などがそれぞれ別個にデータベース化されている [1]。ゲノムとタンパク質とは相互依存的な関係にあって、例えばゲノム情報にはその生物の持つタンパク質の情報が書かれているが、そのゲノムの複製や、ゲノム情報を読み出す過程ではタンパク質が活躍している。

2.1 ゲノム情報

DNA の配列は本来染色体の単位で 1 本にまとまっているものである。よって染色体単位で配列の情報と遺伝情報の分布地図とが分かることがひとつの理想像である。しかし現在の技術では染色体は長過ぎて、そのままでは DNA の配列を読むことはできない。どうしても薬品や超音波などで細かく切断し、解読可能な長さにまで短くしなければならない。

切断して読むという方法をとる以上、その断片の情報をまた 1 本の染色体の情報としてまとめあげる技術が不可欠である。そのためには断片のもとの位置を知る必要がある。断片の位置を知るには、基本的にはその断片になんらかの印をつけてもとの染色体とまぜ、結合する個所を測定するという方法をとる。しかし、繰り返し配列、類似配列の識別などの問題があり、一意に決めることが難しい。

現在の所、断片の DNA 配列の解読に関しては比較的信頼性の高いデータが得られるようになっている。DNA 配列のデータベースも、この切断後の断片の単位でデータを収集している。位置の情報は別のデータベースや参考文献をあたることを前提にしている。大腸菌、線虫など、ゲノムのサイズの比較的小さな生物に関しては、位置情報のデータベースも充実してきている。米国のヒトゲノム計画では、この位置決定技術の研究をまず重視するという方式をとっている (トップダウン方式) [2]。

ゲノム情報データベースのもう 1 つの大きな懸案事項はその量である。ヒトゲノムで DNA 約 30 億、個人差まで格納するようになるとさらに桁が上がることになる。

2.2 タンパク質情報

タンパク質の多くは、多数のアミノ酸が 1 列につながって複雑に折れたたまったものである。(中には複数本のアミノ酸配列が集まって 1 つのタンパク質を構成しているものもある。) タンパク質のデータには、構造に関する情報と機能に関する情報がある。

2.2.1 構造情報

タンパク質の構造情報に関しては極端な話、立体構造のデータさえあればよい。アミノ酸がどのような順番で並んでいるかという配列の情報や、配列的に離れたアミノ酸同士の結合や金属イオンとアミノ酸の結合なども、立体構造データに記述されるからである。またタンパク質合成をしようとする場合、タンパク質のどの機能がどの部分に由来するのかという、タンパク質の部分的な特徴のデータが重要である。タンパク質の機能にはその形状がかなりの割合で効いていると見られており、立体構造のデータが強く求められるひとつの理由ともなっている。

しかし、立体構造はX線による結晶解析やNMRなどといった手法で直接構造を解析することも可能だが、現段階では非常に手間のかかる作業であり、またあまり大きなタンパク質に適用することはまだできない上に、結晶化の難しいタンパク質はX線解析にかかりにくいし、適当な溶媒のないタンパク質はNMRにかかりにくい、などと問題は多い。

そこで、比較的容易に得られるアミノ酸配列から、配列の折れ曲がり方、できれば立体構造を予測したい。そのための方法は既にいくつか提案されており [3]、今後の研究が待たれる。

現在(1991年春)、構造情報でデータベース化されているのは、アミノ酸の配列が約2万、その折れ曲がり方による立体構造や3次元座標による立体構造については数百が登録されている。なお、地球上のタンパク質の種類は1000億という試算がある [4]。

2.2.2 機能情報

機能情報には、タンパク質としての機能と、その構成要素であるアミノ酸単位での機能とがある。前者は例えば化学反応に関与する1物質としてであったり、生体を構成する組織であったりする [4]。後者の例では、金属イオンとの結合やアミノ酸同士の結合などが挙げられる。

機能情報でデータベース化されているものは少ない。化学反応としては酵素のデータベースが代表的で、これと配列データベースの特徴記述の形で記されているアミノ酸単位の機能の2つが、主な機能記述データベースである。なお研究者の個人レベルでは、例えばある機能や構造を司るアミノ酸パターンをモチーフと呼んでデータベース化している例もある。

3 DOOD言語によるアプローチ

ICOTでは現在、演繹オブジェクト指向データベース(DOOD)言語 *QUIXOTE* を開発中である。分子生物学のデータはその有力な適用分野として考えている。現在の興味は大きく分けると次の2点になる。

(1) 分子生物学のデータをDOODの枠組みでモデル化すること

前章で挙げたようなデータがどの程度無理なくDOODの枠組みで表現できるか。既にいくつかの記述実験でその有効性は確認しているが、[5] さらに分子生物学の統合的なデータベースへ向けて、DOODそのものの評価も含めモデル化を検討したい。

(2) DOODで表現されたデータを知識として活用するような高度なデータ解析手法

分子生物学の知識は膨大な量のゲノムやタンパク質の情報を解析することで獲得される。より高度な解析を目指すには、データ解析、知識抽出、そしてそれをもとにした更なるデータ解析、というサイクルを実行していく必要がある。知識ベースがこのサイクルに貢献するには、知識を抽出し格納する部分と、抽出した知識を次のデータ解析に役立てる部分とにおいて、データ解析との関係を密接にするような枠組みが必要である。そのためには、前項で述べたモデル化の容易さの他に、データ解析への利用の面での有効性を検討したい。

本稿では前者の、モデル化の枠組みとしての評価、特にデータ構造とそれに対する操作への要求項目を中心に考察する。

3.1 ゲノム情報の要求

3.1.1 特定個体のゲノム情報

ゲノム情報の利用に関しては、現在はある個体に関する全体像という点に関心が絞られている。ここでの検索要求は、遺伝情報を初めとする様々な特徴の、染色体上の相対的位置関係と DNA 配列である。これには染色体の全 DNA 配列と、 DNA 単位の精度の位置情報を持つ特徴記述の 2 つの属性を持つデータベースがあればいい。

このようなデータベースに関して考えなければいけない点は次の 2 点である。

(1) 長い配列に対するパターンサーチ

DNA のあるパターンが特定の特徴を持つ、例えば遺伝子の発現の速度などを表現していると仮定されているため、そのパターンを高速に検索する必要がある。このための技術としては並列処理、情報検索のテキストサーチ技術、専門知識を用いた事前解析などが挙げられるが、データモデルの話から外れるのでここでは省く。

(2) 複雑なパターンの記述

遺伝子発現の制御に使われるパターンはかなり複雑なものになることが既に知られている。例えば SV40 というウィルスにおける転写制御領域には DNA72 塩基からなるエンハンサー領域が 2 つあり、どちらか一方あれば転写が促進される。さらにこのエンハンサー領域中には 8 塩基ほどの、コア配列と見られるパターンが確認されている [6]。このように特徴記述が入れ子式になっていく場合、フラットな関係モデルでは記述が難しい。現在の配列データベース GenBank ではこのような入れ子は 2 階層までしか許していないし、それぞれの階層でパターンの種類も限定されている [7]。パターンの記述に関しては論理プログラミングの性質が有効であると考えられる。現在のところ DCG を用いる方法や時間論理を応用する方法などが提案されている [8][9]。

3.1.2 ゲノムデータベース構築用

現実には、まだ利用よりは情報の格納と修正に関する特性を考慮する必要がある。

配列の解読も検証も断片単位で行われる以上、断片単位のデータを基礎とするスキーマが必要である。断片の位置は、断片同士の相対的な位置関係として記述されるものと、染色体の中での絶対的な位置で記述されるものと両方が混在したものを扱う必要がある。

また、相対位置関係の記述の尺度や基準は多彩で、生物学の実験から判明してくるさまざまな関係を格納するために、データベース側はそれにあわせてスキーマを柔軟に変更することが要求される。これに関しては、ヒト染色体 21 番の地図を Prolog で記述しようという試みがなされている例がある [10]。そこでは属性と値の組を全てリスト形式で 1 つの引数にまとめることにより、柔軟なスキーマを実現している。

さらに、利用のためには可能な限り絶対的な位置で表示されることが望ましいので、相対的な位置関係のうち、絶対位置に対応がつくものはそのような規則を定義しておいて変換表示させたい。これはビューもしくは演繹機能で実現できる。

3.1.3 個体差の情報

ゲノム情報は将来的には、個体差を考慮した種の全体像に関心が移るだろう。

現在でも個体差の情報は、他の特徴記述と混ざって配列データベースにごく少数ではあるが記載されている。

個体差の情報を、格納効率を重視して現在のように差分でもつべきか、個体毎のコピーで持つべきか、それとも種としてのゲノムのデータモデルとして一から考え直すべきかは、生物学が進歩してデータが集まり、個体差がDNAにどのように表現されてくるのかある程度判明するまで、その検討を保留せざるをえないだろう。

3.2 タンパク質情報の要求

3.2.1 構造情報

立体構造データではタンパク質が単位であり、それがタンパク質の情報単位として望ましいわけだが、現在の配列データの単位はアミノ酸配列1本で、複数本からなるタンパク質は複数のデータ単位からデータが構成される形を取っている。将来的には配列単位のデータはタンパク質単位のデータの一部として扱われることになるだろう。

立体構造のデータへの利用に関しては、3D画像の形で見たいという要求、部分的な形状を検索するという要求などが考えられる。全体の画像を見せるだけなら、例えば専用の属性を設ければ十分だろう。後者は難しい問題を含んでいるが、現在のアプローチとしては、アミノ酸配列のパターンに帰着する方法がとられている。すなわち、ある部分形状をなすアミノ酸配列はどういう共通パターン(モチーフ)をとるか、という問題と、そのモチーフで別のタンパク質を検索するという問題の2つに分割して考えるわけである。

モチーフの抽出法は、マルチプルアラインメントという手法が一般的である [11]。得られたモチーフのデータは文字パターンの正規表現に準ずる方法で表されることが多い。得られたモチーフからアミノ酸配列を高速に検索するには、配列の方も事前に解析しておいてそれを索引として使う方法が有効と考えられる。

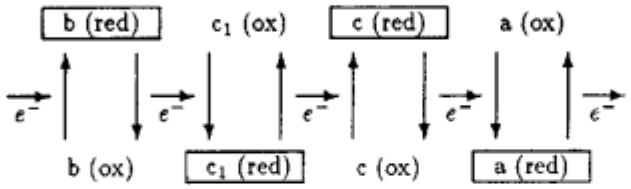
3.2.2 機能情報

タンパク質の機能をおおまかに分けると、酵素、輸送、栄養、収縮、組織、防御、制御などとなる [4]。それぞれの機能を表現するために必要な属性を考えると、酵素以外のタンパク質は活動場所や活動対象などの名称を挙げることで表現できると考えられるのに対し、酵素に関してはその活動対象もしくは場所として化学反応式を扱わなくてはならない。(ただしこの分類において、酵素以外の詳しい機能は未整理であることは確かで、今後の課題と言える。)

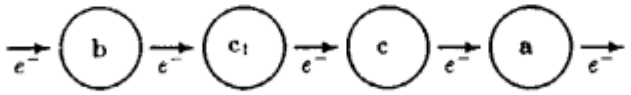
酵素機能を表すための化学反応式は、酵素名、物質の集合、環境の3つの要因と、そこから結果として得られる物質の集合、環境、この5要素が記述されればよい。記述の単位はその酵素の働いている範囲である。

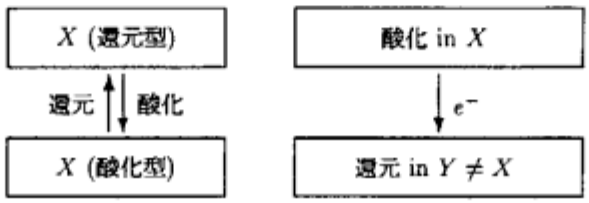
しかし実際は、さまざまな観点から記述された化学反応式の中から酵素の関わる反応式を検索してくる形になる。そのためにはまず反応式をつなぐという演繹的な操作が要求される。また化学反応式には、物質の表現には集合、反応の表現にはおおまかなレベルから中間的に生産される物質の記述を含んだ細かいレベルまでの記述、というハイパーグラフ的な表現が要求される。

さらに、反応間でエネルギーの交換が発生するなど、反応間関係も記述する必要がある。以下の図はシトクロムにおける電子輸送という、反応間関係を要する例である。



(1) 総括的記述





(2) 段階的記述

演繹的な質問処理を想定すれば、シトクロム間の電子の流れと酸化還元による電子輸送の規則で段階的に表現した下段 (2) の方式で上段 (1) のような表現を構築することができる。(2) の書き方もまた、ノードを詳しく見るとアークとノードから成っているという、ハイパーグラフによる表現である。

4 *Quixote* による記述実験

前章の要求をまとめると、特徴記述とモチーフで文字列パターン、酵素機能でハイパーグラフの表現と利用が問題であると言える。ここではこれを *Quixote* [12][13][14] による記述実験の対象として選んでみた。

4.1 パターンの記述例

3.1.1 のエンハンサーの例は以下のように記述できる。

```
m_enhancer :: {{                                                    %% 1
  enhancer_core.                                                    %% 2
  enhancer_region / [l_length = 72].                                %% 3
  enhancer1 =< enhancer_region.                                     %% 4
  enhancer2 =< enhancer_region.                                     %% 5
  enhancer_features / [l_include+ <- {enhancer_core} ].             %% 6
  enhancer_features =< enhancer1.                                   %% 7
  enhancer_features =< enhancer2.                                   %% 8

  enhance_effect <= control_region /
                    [l_include+ -> {enhancer1, enhancer2}].         %% 9
}}
```

1 はモジュール名の宣言。モジュールはデータとルールの集まりとして定義でき、特定分野の知識を他と独立に記述するための仕組みである。これにより、例えばエンハンサーの上位概念である遺伝子発現制御領域などといったものの記述の際には、このモジュールのデータやルールを継承する形にでき、記述の見通しの良さや全体の記述量の削減に貢献できる。

2 は enhancer_core というオブジェクトの存在の宣言。"enhancer_core" はオブジェクト識別子である。これは属性を記述しない例。

3 は enhancer_region というオブジェクトの存在宣言と同時に、その内容を属性と値のペアで説明している。6 も同様で、enhancer_features というオブジェクトの内容記述である。属性は単値も集合もとることができる。属性とその値の間の包摂関係を記述することもできるようになっていて、この例では enhancer_features の集合属性 l_include+ は { enhancer_core } も含む、という意味を <- が示している。

4,5,7,8 はオブジェクト間の包摂関係を示している。enhancer_region にはエンハンサーというものの定義を書き、enhancer_features にはエンハンサーに見られる特徴を列挙していくという位置付けにしていることを記述している。enhancer1 と enhancer2 は enhancer_features の属性値で包摂関係が <- ,= であるものと、enhancer_region の持つ定義情報 (正確には -> ,= である属性値) とを継承する。継承した分も含めて書くと、

```
enhancer1 / [l_include+ <- {enhancer_core}, l_length -> 72 ].
enhancer2 / [l_include+ <- {enhancer_core}, l_length -> 72 ].
```

という形になる。

9 はルールである。control_region に enhancer_region のどちらかがあれば、enhance_effect があると言っている。

4.2 ハイパーグラフの記述例

3.2.2 のシトクロムの例の「段階的表現」の記述例は以下の通り [15]。

```
m_outline :: {{                                                    %%  1
  el_transfer [l_donor= cyto_b, l_acceptor= cyto_c1].              %%  2
  el_transfer [l_donor= cyto_c1,l_acceptor= cyto_c ].              %%  3
  el_transfer [l_donor= cyto_c, l_acceptor= cyto_a ].              %%  4

  cyto_b  =< cytochrome.                                           %%  5
  cyto_c1 =< cytochrome.                                           %%  6
  cyto_c  =< cytochrome.                                           %%  7
  cyto_a  =< cytochrome.                                           %%  8
  cytochrome.                                                      %%  9
}}
m_detail >- m_outline.                                             %% 10
m_detail :: {{                                                     %% 11
  oxidation [l_object = X]
     <=el_transfer [l_donor = X, l_acceptor = Y].                  %% 12
  reduction [l_object = X]
     <=el_transfer [l_acceptor = X, l_donor = Y].                  %% 13

  oxidation =< reaction.                                           %% 14
  reduction =< reaction.                                           %% 15
  reaction.                                                        %% 16

  oxidation [l_object = X] /
       [l_source+ <- Y, l_product+ <- Z] <= ||
           {Y =< X, Y!l_type = reduced,
            Z =< X, Z!l_type = oxidized}.                          %% 17
  reduction [l_object = X] /
       [l_source+ <- Y, l_product+ <- Z] <= ||
       {Y =< X, Y!l_type = oxidized,
        Z =< X, Z!l_type = reduced}.                               %% 18

  cytochrome [l_type = reduced ].                                  %% 19
  cytochrome [l_type = oxidized].                                  %% 20
}}
```

1 から 9 までが電子の流れを示すモジュール (m_outline)。2 から 4 が流れを示し、5 から 9 は各種のシトクロムを定義している。

11 から 20 までは酸化還元機構と電子輸送の関係を記述したモジュール (m_detail)。10 はモジュール間の関係を定義している。これで m_outline モジュールの記述は全て m_detail に継承される。12 と 13 は電子の輸送による酸化還元の定義。14 から 16 は酸化も還元も反応の一種と宣言。17 と 18 は酸化型と還元型の間で電子のやりとりが行われ、それをそれぞれ酸化反応、還元反応と呼ぶ、という記述。19 と 20 はシトクロムに酸化型と還元型が存在するという宣言をしている。

2 から 4 で記述したように、属性がオブジェクトの識別に必要な場合は / なしでその属性を書く。オブジェクト識別とは無関係な属性をそこに記述する際には、例えば 17 や 18 のように / の前にオブジェクト識別用属性、後ろに非識別用属性を書く。この 17, 18 では || の後ろに変数の制約条件を記述しており、それがこのルールのボディに相当している。A!B は、A / [B = X] であるような X を表している。

4.3 演繹データベースと *Quixote*

以上の例から Prolog や演繹データベースと DOOD 言語 *Quixote* の比較をすると、次のような点が *Quixote* の利点として挙げられる。

(1) モジュールとモジュール間の継承機能

知識の分類と独立な記述が可能となる。

(2) オブジェクト識別子

識別用属性(オブジェクト項)と非識別用属性(属性項)の区別により、参照時の記述が大幅に簡潔化する。順序で意味の決まるPrologや演繹DBの記述よりも、属性名を必ず付ける分記述は長くなっているが、これにより総体的にはむしろ短くなると考えている。

(3) 集合の取り扱い

(4) 属性と値の間の包摂関係記述による制約表現

ここで属性の型や、属性値における and, or の記述ができる。

(5) オブジェクト間の上下双方向の継承機能

5 今後の展望と課題

分子生物学のデータベースを構築するための、データモデルへの要求項目を考察した。また、それを *Quixote* によって記述し *Quixote* の利点と可能性を示した。

今回は量に関する問題への解決策は提示できなかった。格納方式にせよ高速化にせよ、分散化や並列化が鍵となるであろうことは容易に予想できる。ICOTでは並列化による効果に今後重点をおいて注目していくことになる。

記述実験例として挙げたものに関しては、実システムでの実行が目下の課題である。*Quixote* はこの研究会発表の頃には第1版が、並列実験マシンであるマルチPSIの上で動く予定になっている。

謝辞

最後に本研究に関して多くの示唆を頂いた横田一正氏(ICOT)と、Quixote会議、遺伝子情報WG、分子生物情報メーリングリストの方々に感謝致します。

参考文献

[1] Lesk, A.M. (editor) : *Computational Molecular Biology, Sources and Methods for Sequence Analysis*, Oxford Univ. Press (1988).

[2] 米国政府機関報告書: "Mapping Our Genes, The Genome Projects: How Big, How Fast", Congress of the United States, Office of Technology Assessment, Johns Hopkins University Press (1988) (伊藤 訳:「Newton Special Issue ヒトゲノム解析計画」教育社)

[3] 西川:「タンパク質の二次構造予測」, 情報処理, Vol. 31, No. 7, pp.887-896 (1990).

[4] Lehninger, A.L.: *Principles of Biochemistry*, Worth Publishers Inc. (1982).

[5] 田中:「代謝反応データベース」, 情報処理学会研究報告 90-DBS-78-14, (1990).

[6] 松原 編:「遺伝子と遺伝の情報 I」, 岩波書店, (1989).

[7] Cinkosky, M.J. et al.: "GenBank/HGIR Technical Manual", *LA-UR* 88-3038 (LANL) (1988).

[8] Searls, D.B.: "Investigating the Linguistics of DNA with Definite Clause Grammars", *NACLP 89*, pp.189-208 (1989).

[9] Overton, G.C., Koile, K. and Pastor, J.A.: "GeneSys: A Knowledge Management System for Molecular Biology", *Computers and DNA, SFI Studies in the Science of Complexity*, vol.VII, Addison-Wesley, pp.213-239 (1990).

[10] Yoshida, K., Overbeek, R., Zawada, D., Cantor, C.R. and Smith, C.L.: "Prototyping a Mapping Database of Chromosome 21", *Proceedings of Genome Mapping & Sequencing Meeting*, Cold Spring Harbor Laboratory, (1991).

[11] 五條堀 他: 「大量DNAデータを対象とした遺伝情報のコンピュータ解析」, 情報処理, Vol. 31, No. 7, pp.878-886 (1990).

[12] Yasukawa, H. and Yokota, K.: "The Overview of a Knowledge Representation Language *QUIXOTE*" *Draft*, (1990).

[13] 森田, 羽生田, 横田: 「*QUIXOTE* のオブジェクト識別性」, 情報処理学会研究報告 90-DBS-80-12, (1990).

[14] 安川, 横田: 「ラベルつきグラフに基づくオブジェクトの意味論」, 情報処理学会研究報告 90-DBS-80-13, (1990).

[15] Tanaka, H.: "Protein Function Database as a Deductive and Object-Oriented Database", *DEXA 91*, (1991).