A-010

DNA 分子を利用したリレーショナルデータベースの開発
A Development of DNA Relational Database

北 豊† 柏村 聡† 亀田 充史‡ 山本 雅人†,‡ 大内 東†,‡
Yutaka Kita Satoshi Kashiwamura Atsushi Kameda Masahito Yamamoto Azuma Ohuchi

1. はじめに

近年，DNA の安定性，微細性を利用した DNA メモリに関する研究が行われている．しかし，DNA メモリの多くはストレージとしてのメモリであり，DNA 分子に格納されているデータの処理までを考慮したデータベースの観点では不十分な点が多い．DNA メモリがデータベースの機能を持つことができれば，ゲノムデータなどの遺伝情報を電子データに変換することなく DNA 分子中に保存でき，DNA 分子の利点である超並列操作を利用した操作ができる．本研究ではリレーショナルモデルを DNA 分子に実装し，各関係演算について化学実験を行った上で，提案したモデルの有効性を検証する．

2. DNA 分子を利用したデータベース

2.1 DNA 分子を利用する利点

DNA 分子にデータを保存するということはデータを塩基配列の並びで表現することであり，ゲノムデータはまさに塩基配列の並びであり DNA データベースで扱い易いデータである．Reif たちは DNA 計算を用い DNA 分子をそのままデータベース化し検索をする方法を考案している[1]．また，DNA 分子の並列性によりデータサイズが大きくなっても化学反応の時間はほぼ一定であり，スケーラビリティが高いと言える．さらに，類似したデータを取り出す必要がある場合，電子的なデータベースではすべてのデータに対して類似度をチェックしなければならないが，DNA 分子の場合には，類似度の高い分子は完全な相補配列でなくても結合する．これを利用すると同時並行的に類似したデータを取り出すことができる．この機能はホモロジー検索への応用可能性がある．

2.2 関連研究

DNA 分子によるリレーショナルデータベースについての研究の中で，Arita らは関係演算の実現可能性について実験的に検討している[2]，また， Katsányi は基本的な実験操作での関係演算の実現可能性を理論的に証明している[3]．しかし，これらのデータモデルでは１つの DNA 分子に複数のデータを保持しているためデータに順序が生まれ，関係演算を行う際に順序を解消する実験操作が必要になる場合が生じ，エラーの蓄積の原因となると考える．そこで，本研究では１つの DNA 分子には 1 データを保持するデータモデルを使用する．

3. モデル化

3.1 定義

あるリレーション R(n×m)を表す DNA 分子を含んだ試験管を U とする．

$R(A_1,...,A_n) = \{(v_1^{(1)},...v_n^{(1)}),...,(v_1^{(m)},...v_n^{(m)})\}$

A_i : **属性**(Attribute)

$v_i^{(j)}$: **関係の要素**

$t_j = (v_1^{(1)},...v_n^{(1)})$: **組**(tuple)

ID_j : ***j*を保持するキー**

i=1,…,nであり，j=1,…,m である．

各データは図 1 のように，属性の情報とタプルに関する情報とともに一本鎖 DNA によって表される．

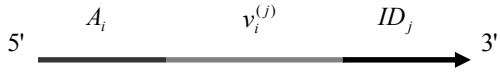

図 1：一本鎖 DNA によるデータの表現

3.1.1 基本命令(実験操作)

以下にデータベースを処理する際に必要となる基本的実験操作について定義する．

- *Merge*(U_x, U_y)：試験管 U_x，U_y 中にある DNA 分子を混合する
- *Amplify*(*U, FW, RE*)：試験管 *U* に含まれる DNA 分子に対してプライマー集合 FW,RE で増幅反応を行う
- *Append*(*U, S, E*)：試験管 *U* 中の DNA 分子で 3 ’末端に *E* を含む配列に対して DNA 分子に対して対応する配列集合 *S* を付加する

また，試験管 *U* 中の DNA 分子に対して片方の鎖の末端が biotin 化された二本鎖 DNA 分子において

- *Separate*$_+$*(U)*：biotin 化されていない方の一本鎖 DNA を抽出する
- *Separate*$_-$*(U)*：biotin 化されていない方の一本鎖 DNA を抽出する

試験管 *U* 中の DNA 分子に対して，

- *Get*(*U, +S*)：部分配列集合 *S* のうち何れか含んだ一本鎖 DNA を抽出する
- *Get*(*U, −S*)：何れの *S* も含まない一本鎖 DNA を抽出する

3.1.2 データベースの基本操作

データベースにおけるデータの挿入，削除，更新，問い合わせの基本操作は前節の基本命令を用いて以下のように記述できる．

データの挿入 : $Insert(U,s) := Merge(U,s)$

データの削除 : $Delete(U,s) := Get(U,-s)$

データの更新 : $Update(U,s_{old},s_{new})$ *do* :

$$Delete(U,s_{old})$$
$$Insert(U,s_{new})$$

データの問い合わせ : $Select(U,i,j) := Amplify(U,A_i,\overline{ID_j})$

$\overline{ID_j}$はID_jの相補鎖を表す

†北海道大学大学院情報科学研究科複合情報学専攻
‡CREST, Japan Science and Technology Corporation

3.2 関係演算の実装方法

独立な 5 つの関係演算(和，差，射影，選択，直積)について化学実験での実装方法を基本命令を用いて記述する．

3.2.1 和(Union)

2 つの union compatible な R_x，R_y に対して，和集合をとる．

$R_z := R_x \cup R_y \; do:$

$U_z = Merge(U_x, U_y)$

3.2.2 差(Difference)

2 つの union compatible な R_x，R_y に対して，差集合をとる．

$R_z := R_x - R_y \; do:$

$S_1 := Amplify\,(U_y, 0, RE)$

$RE = \{\overline{ID}_j^{\,y} \mid j = (1,..., m), 3'\,end\ biotinylat\ ed\}$

$S_2 := Separrate\ _{-}(S_1)$

$T_z := Get\,(U_{x,} - S_2)$

Separate_(S1)，Get(Ux－S)の操作にはアフィニティ分離を用いる．

3.2.3 射影(Projection)

R において，X を{A1,…,An}の任意の部分集合とするとき，X で指定された属性を取り出し，新しい関係を生成する．

$R := \Pi_X R \; do:$

$S_1 := Amplify\,(U, X, RE)$

$X = \{A_i^x \mid i = \{1',..., k'\} \subseteq \{1,..., n\}\}$

$RE = \{\overline{ID}_j \mid j = (1,...m), 3'\,end\ biotinylat\ ed\}$

$U = Separate\ _{+}(S_1)$

3.2.4 選択(Selection)

選択条件 F を用いて，条件を満足する組を取り出す．

$R := \sigma_F R \; do:$

$S_1 := Amplify(U, FW, F)$

$FW = \{A_i \mid i = (1,..., n)\}$

$F = \{\overline{ID}_j \mid P_F(t_j), 3'\,end\,biotinylated\}$

$U := Separate_{+}(S_1)$

$P_F(t_j)$は，tj が F を満足する時に真となる述語である．

3.2.5 直積(Cartesian product)

2 つの $R_x(A_1,...,A_n)$，$R_y(A_1',...,A_n')$の直積をとる

$R_z := R_x \times R_y \; do:$

$S_1 := Append(U_x, S, E_x)$

$S_2 := Append(U_y, S, E_y)$

$E_x = \{ID_j \mid j = (1,..., m)\}$

$E_y = \{ID_{j'} \mid j' = (1',..., m')\}$

$S = \{ID_{f(j,j')} \mid t_j \in R_x, t'_{j'} \in R_y\}$

$U_z := Merge(S_1, S_2)$

$f(j, j')$は$t_j, t'_{j'}$の連結でできるタプル$(t_j, t'_{j'})$を区別するためのj, j'によって一意に決まる値

Append(U, S, E) の操作には State Transition PCR(ST-PCR)[4]を用い完全な二本鎖の形成を防ぐ．

4. 実験

化学実験で関係演算が適切に実行できるかどうかを検証するために 2 つのリレーション R(3×3)，R'(2×2)を設定し，実験を行う．

4.1 配列設計

本研究で用いる塩基配列の長さは，16 塩基を 1 ユニットとし，このユニットを属性 A，タプルの情報 ID，データの要素 v にそれぞれ割り当て，このユニット対しての配列設計を行う．上記の 2 つのリレーションに対して全てのデータが異なる場合，設計する配列ユニット数は，リレーション R，R'についてそれぞれ 15，8 ユニット，R，R'の直積に必要な 6 ユニット，合計 29 ユニットが必要になる．各配列ユニットは正規直交性を満たしている必要があり，さらに PCR などの実験操作を実現するための制約が必要になる．

本研究では Template Method の GC-Template[5]によって設計される配列集合を使用した．GC-Template の性質として全てのユニットの塩基 G，C の含有量(GC-content)が揃っているだけでなく，GC の位置(GC-position)が揃っている．このため配列の Tm(融解温度)の分布が揃っているユニットを使用でき，配列ユニットの違いによる反応速度の違いが生じにくくなることが期待される．GC-Template では Hamming-distance についても連結部分を含めて保障されているので mis-hybridization が起こりにくくなっている．また，PCR の primer はミスマッチによるアニールを防ぐために 3'末端の塩基は T を避けること，primer-dimer などが生じないように 3'末端に G or C が 3 塩基以上連続しないことを考慮して設計を行うが，GC-Template では Template を決める段階でこの制約を満たせる．

しかし，Template Method ではユニット，ユニットを連結した一本鎖 DNA が分子内で結合する 2 次構造について考慮されておらず，意図した反応を妨げる要因となりかねない．そこで一本鎖 DNA の二次構造を予測するプログラム Mfold[6]を用いて⊿G (自由エネルギー)と 2 次構造を求めて配列の選定を行った．配列の選定方法は GC-Template で生成される 16 塩基の配列 64 ユニットとその相補配列 64 ユニットの計 128 ユニットからランダムな 3 ユニットを組み合わせて 48 塩基の配列を 20000 セット作り，Mfold で⊿G を算出し，値が上位の配列から選択した．Transition-molecule については残った配列ユニットからすべての組み合わせについて Mfold を使用し，R，R'の Transition-molecule が同時に安定な二次構造を形成しない組み合わせを選んだ．

4.2 化学実験

データの問い合わせと関係演算の実行について化学実験を行って確かめた．和の操作は DNA 分子を混合するだけなので省略することにする．また，射影と選択の操作は同じ実験操作で実現できるのためここでは選択の実験を行った．直積の操作については，一本鎖 DNA を伸長する ST-PCR によって最も重要な操作の *Append(U, S, E)*が実現できるかを確かめた．ここで，実験で用いる A_i，ID_j を含む一本鎖 DNA を data(i, j)，A_i，ID_j に対するプライマーセットを p(i, j)，と定義する．

4.2.1 データの問い合わせ

Select(U, 1, 1*)*を行った．すなわちタプル t_1 の属性 A_1 を問い合わせる操作である．9 種類の DNA 分子を含む R に対して，p(1, 1)を用いて PCR を行う．反応後の溶液では A_1, ID_1 の何れかを含む配列が線形的に増加することになるが，A_1，ID_1 を含む data(1, 1)が指数的に増幅され，大部分が data(1, 1)となる．確認のために PCR 後の溶液を希釈したものを p(1, 1)，p(2, 1)，p(2, 2)を用いて PCR を行った．反応後のゲル電気泳動の結果が図 2 である．p(1, 1)で PCR したものが目的の位置にバンドが現れ，残りのプライマーセットではほとんどバンドが現れていなかった．よって Select 後の溶液の大部分が data(1, 1)であり問い合わせの反応は実現できる．

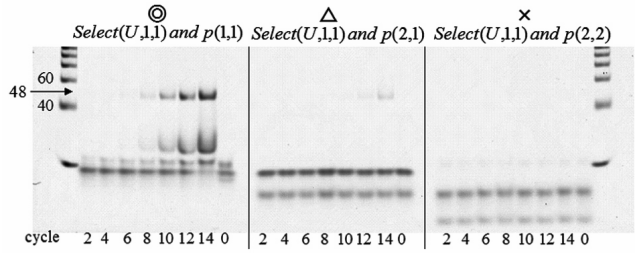

図 2：Select の確認実験

4.2.2 *Append(U,S,E)*の実験

48 塩基の data(1, 1)，data(2, 1)，data(3, 1)に対して Transition-molecule を用いて ST-PCR で $ID_{f(1,\,1')}$16 塩基を付加した．適切に動作した反応後の data 配列が 64 塩基の長さであればよいが，実際に 64 塩基となっていることを確認した．反応後の data(1, f(1, 1'))，data(2, f(1, 1'))，data(3, f(1, 1'))は差の実験を長さによって確認するために用いた．

4.2.3 差の実験

$R_x - R_y = \{($ data(1, f(1, 1')), data(2, f(1, 1')), data(3, f(1, 1'))), (data(1, 2)，data(2, 2)，data(3, 2))$\} - \{$(data(1, 2)，data(2, 2)，data(3, 2))，(data(1, 3)，data(2, 3)，data(3, 3))$\}$を行った．今回の実験では完全相補鎖の代わりに ID 部分の相補配列の 5'末端を biotin 化したものを用いた．図 3 にゲル電気泳動の結果を部分的に示す．R_x と R_y を混合した時点で biotin 化した配列と data(*, 2)の配列が結合していることがわかる．そして，アフィニティ分離で biotin 化されていない配列を取り出すことによって，64 塩基以外のバンドは取り除くことができており，差が実験的に実現できることを確認した．

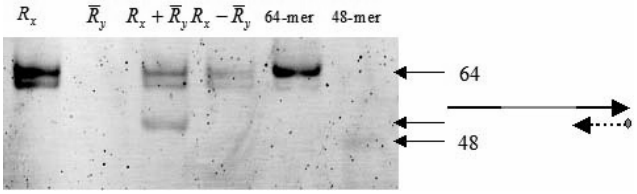

図 3：差の実験結果

4.2.4 選択の実験

R(3×3)からタプル t_1 の要素を取り出す実験を行った．p(1, 1)，p(2, 1)，p(3, 1)を用いて PCR を行う．このとき ID 部分のプライマーは差の実験と同様に 5'末端が biotin 化してある．PCR 後に Select の確認実験と同様に反応溶液を希釈し，すべてのプライマーの組み合わせについてそれぞれ別々に PCR を行った．図 4 が確認実験後の溶液をゲル電気泳動した結果である．P(*, 1)で PCR を行ったものは目的の位置にバンド(48 塩基)が出てきているが，それ以外のプライマーの組み合わせの場合は増幅が見られない．選択の実験が実現できているので同じ実験操作の射影についても同様に実現可能であると考える．選択，射影の実験後の data 配列は二本鎖になっているので，この処理の後，別なデータ操作を行う場合は，アルカリ変性などで一本鎖の状態にする必要がある．

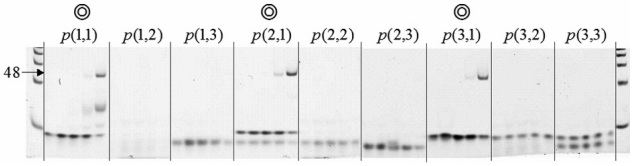

図 4：選択の確認実験の結果

5. まとめ

本研究では，DNA 分子によるリレーショナルデータベースの実現方法として，従来よりも化学実験の操作が簡単なモデルを考案した．また，化学実験によりいくつかの演算について実行可能であることを示した．今後の課題としては，全ての関係演算の化学実験での確認，連続したデータ処理を行う場合の動作精度に関する分析が必要となるであろう．また，どれだけ大規模なデータベースを構築可能であるかというスケールアップの課題があるであろう．

参考文献

[1] J. H. Reif, T. H. LaBean, M. Pirrug, V. S. Rana, B.Guo, C. Kingsford, and G. S. Wickham, "Experimental Construction of Large Scale DNA Databases with Associative Search Capability", 7th Internatinal Meeting on DNA-Based Computers (DNA7), in Lecture Notes in Computer Science, pp. 231-247,2002.

[2] M. Arita, M. Hagiya, and A. Suyama, "Joining and Rotating Data with Molecules", Proc. of IEEE 4th International Conference on Evolutional Computation (ICEC'97), pp.243-248, in IEEE press, 1997.

[3] I. Katsányi, "On Implementing Relational Database on DNA Strands", Acta Cybernetica, vol. 16(2), pp. 259-270, 2003.

[4] K. Hashimoto, A. Kameda, M. Yamamoto and A. Ohuchi, "State Transition Model Based On DNA Polymerization", Proc. of the International Technical Conference on Circuit/Systems, Computers and Communications (ITC-CSCC'03), pp.1889-1892, 2003.

[5] M. Arita, and S. Kobayashi, "DNA Sequence Design Using Templates", New Generation Computing, vol. 20, pp. 263-277, 2002.

[6] M. Zuker, "Mfold web server for nucleic acid folding and hybridization prediction", Nucleic Acids Research, vol. 31, pp. 3406-15, 2003.