



DNAコンピュータとは

新しい計算機の可能性に向けて

寄 書

Lila KARI 榊原 康文

Lila KARI: Department of Computer Science
University of Western Ontario
榊原康文: 東京電機大学理工学部情報科学科
E-mail yasuj@dendai.ac.jp

DNA Computers. By Lila KARI, Nonmember (Department of Computer Science, University of Western Ontario, Ontario, N6A5B7, Canada), and Yasubumi SAKAKIBARA, Nonmember (Faculty of Science and Engineering, Tokyo Denki University, Saitama-ken, 350-03 Japan).

Adleman の実験

Adleman⁽¹⁾ は、DNA 配列上に問題を符号化し、試験管内の DNA 配列に対して分子生物学の実験操作を行うことによって、頂点数 7 の有向ハミルトン経路問題を実際に解いてみせた。有向ハミルトン経路問題とは、有向グラフ G において、指定された始点 v_{in} から終点 v_{out} へ至るパスで他のどの頂点もちょうど一度だけ通過するようなものが存在するか否かを決定する問題である。この問題は、次の単純な（非決定性の）アルゴリズムによって解くことができる。

- ステップ 1: 有向グラフ G 上のいくつかのパスをランダムに生成する。
- ステップ 2: v_{in} で始まり v_{out} で終わるパスのみを選択する。
- ステップ 3: グラフ G の頂点数を n としたとき、含まれる頂点の数がちょうど n であるパスのみを選択する。
- ステップ 4: すべての頂点を少なくとも一度だけ通過するパスを選択する。
- ステップ 5: もしこの時点において残っているパスが存在すれば、それはこの問題の解であり、解が存在すると判定し、一つもパスが残っていないければ、解はないと判定する。

このアルゴリズムを、試験管の中の DNA 配列とそれに対する分子生物学の実験操作を用いることによって、実現しようというのが Adleman の試みである。彼は、以下のような方法を取った。

- (1) ステップ 1 を実現するために、グラフ上の各頂点を、A, C, G, T の 4 文字のアルファベットからなる長さ 20 のランダムな DNA 配列に符号化する。次にグラフ上の各有向辺に対して、その辺の始点の頂点を符号化している DNA 配列の後半の半分とその終点の頂点を符号化している DNA 配列の前半の半분을結合した DNA 配列を用意する。そしてこのように各辺を符号化した DNA 配列すべてと各頂点の DNA 配列の相補配列すべてを生成することにより、各 DNA 配列は試験管の中でその相補配列と結合（アニール）するため、辺で結ばれた頂点の間だけがつながっていく。リガーゼという酵素を用いた反応により、試験管中にランダムなパスを符号化する DNA 配列が次々に合成されていく（図 1 参照）。
- (2) ステップ 2 を実現するために、ステップ 1 で生成されたパスに対して、 v_{in} で始まり v_{out} で終わるパスを符号化している DNA 配列を PCR（ポリメラーゼ連鎖反応）を用いて大量に増幅する。

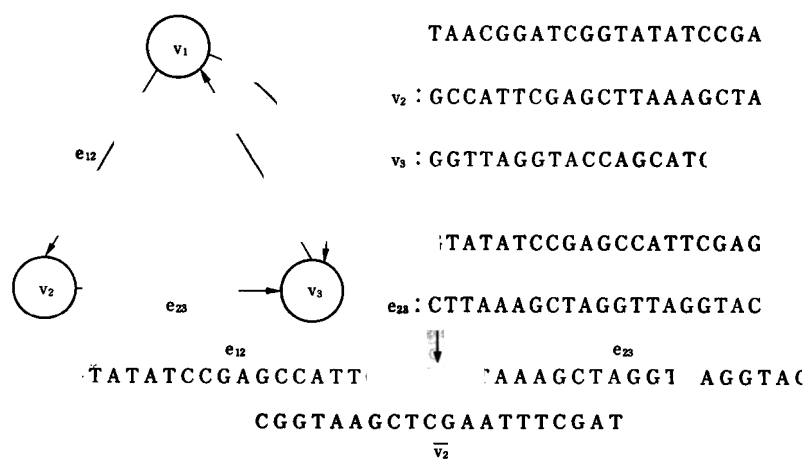


図1 有向グラフの DNA 配列上への符号化 3 頂点の有向グラフと、三つの各頂点を符号化する長さ 20 の DNA 配列と頂点 1 と頂点 2 を結ぶ辺と頂点 2 と頂点 3 を結ぶ辺を符号化する長さ 20 の DNA 配列。A と T, G と C の相補対の結合により、頂点 2 の DNA 配列の相補配列を経由して二つの辺が結ばれる。

- (3) ステップ 3 を実現するために、ゲル電気泳動とよばれる方法を用いて、長さによって DNA 配列を分離する。
- (4) ステップ 4 は、ある部分配列 (ある頂点を符号化している配列) を含んでいる DNA 配列だけを取り出す実験操作によって実現される。この操作を n 個の各頂点に対して繰り返し行う。
- (5) ステップ 5 の実現は、以上の操作が行われた試験管中に一つでも DNA 配列が存在しているか否かを確認し、残っていればその配列を読みとり、解を与える。

日本でも有田ら⁽²⁾により追試の報告がなされているので、実験に関する分子生物学的側面の詳細については、そちらを参照されたい。

さて、この実験において注目すべき最大のポイントは、現在の決定性のコンピュータでは効率良く解くことが難しいと考えられている NP 完全問題 (決定性の多項式時間アルゴリズムが現在まで知られていない、しかし非決定性のアルゴリズムで多項式時間で解くことのできる問題で最も難しいもの) の一つである有向ハミルトン経路問題を、試験管内で発生する DNA 分子の超並列的な相互作用を用いて現実的時間内

で解くことの可能性を示したという点である。ここに Adleman の実験の意味があり、DNA コンピュータを考える意義がある。またこのことを可能にしているのは、PCR とよばれる DNA 配列を大量に増幅させることのできる分子生物学の実験手法である。

2 DNA コンピュータ

Adleman の実験は、ある特定の問題を DNA を用いた分子生物学的実験的手法により解くことによって、DNA コンピュータの可能性を示したわけであるが、このような DNA コンピュータの優位性と実現性について考えてみることにする。

DNA コンピュータを考える最大の理由は、分子生物学の実験で用いられる操作が、小さな試験管内での超並列的な探索を可能とすることである。そして、DNA コンピュータは、スピード、エネルギー効率、情報格納量の点から非常に大きな優位性をもっている^{(1),(3)}。例えば、Adleman のモデルでは、1 秒間に実行されるオペレーションの数は、約 1.2×10^{18} ぐらいに及ぶと見積もられている。これは、現在の最速のスーパーコンピュータより、大体 120 万倍速いこ

ととなる。また現在のスーパーコンピュータは、1ジュール当り 10^9 のオペレーションを実行するが、DNA コンピュータは1ジュール当り 2×10^9 を実行する。すなわち、DNA コンピュータの方が、10億倍以上のエネルギー効率が良いことになる。更に、DNA 分子における情報格納量は、 1 nm^3 (立方ナノメートル (nanometer)) 当り1ビットの情報量を許すが、現在の情報記憶媒体では、 10^{12} nm^3 当り1ビットの情報を格納する。DNA メモリは、かつてない超高密度の記憶媒体を提供する。

さて、DNA コンピュータの実現性については、Adleman はその実験において、一つの具体的な問題がDNA コンピュータによって解けることを示したにすぎない。そこで、DNA コンピュータを実現するためには、

(1) どのような種類の問題がDNA コンピュータによって解くことができるのか？

(2) プログラミング可能なDNA コンピュータを設計することができるのか？

という問題に答える必要がある^{(4),(5)}。特に(2)の問題、すなわち、任意のプログラムとその入力となるデータを受け取り、そのプログラムを実行することにより関数を計算する装置を作ること、汎用のDNA コンピュータを実現する上で重要な課題である。このような汎用のDNA コンピュータを作る上で、現在利用可能な分子生物学的実験操作は次のものがある^{(1),(3),(5)-(7)}。

- 合成：DNA 合成機による、長さが100程度までの任意の配列の合成。
- 分離：ゲル電気泳動後のゲルを切り出すことによる特定長の配列の分離。
- マージ：二つの試験管の中身を一つに混合。
- 抽出：あるパターンを部分列として含んでいる配列のピオチン アピジンを用いた抽出。
- アニールリング/メルティング：相補的な二つの配列の2本鎖への結合、2本鎖から1本鎖への分離。

- 増幅：PCR (ポリメラーゼ連鎖反応) を用いたDNA 配列の大量のコピーの作成。
- 切断：制限酵素を用いたDNA 配列の切断。
- ライゲーション：リガーゼを用いて相補的な断面をもつDNA 配列の連結。
- 検出：試験管の中に一つでもDNA 配列が含まれているか、または何も含まれていないかを検出する。

DNA コンピュータにおけるプログラムは、これらの操作を組み合わせて書かれると考えられる。そしてこのようなDNA コンピュータのプログラムは、入力としてDNA 配列を含んだ試験管を受け取り、出力としてイエス、ノー、または試験管の集合 (これは計算結果を表している) を返すこととなる。DNA 配列を含んだ試験管に上の一つの操作を適用することが、DNA コンピュータにおける計算の1ステップと考える。従って計算の全体は、試験管の列からなることになる。

3 DNA コンピュータの理論モデル：スプライシングシステム

DNA コンピュータを考えるもう一つのモデルが、Head⁽⁶⁾ により提案されている。このモデルは、Adleman の実験的モデルとは対照的に数学的モデルであり、DNA コンピュータの基礎理論を与えるものと位置付けられる。Head のモデルは、スプライシングシステムとよばれ、その基本操作は、酵素 (制限酵素とリガーゼ) によるDNA 配列の切断・結合のみである。しかも、計算のすべてが始めから終わりまで一つの試験管内で行われるモデルである。このモデルの利点は、すべての議論が厳密な数学的裏付けをもって行われる点であり、逆に欠点は、スプライシングシステムは現在の分子生物学の実験方法では、まだ実現されていないという点である。

スプライシングシステムは、制限酵素とよば

れるある種類の酵素により、ある特定の部分配列を認識してその位置でDNA配列を切断する操作と、リガーゼとよばれる酵素により、切断された相補的な断面をもつDNA配列同士を連結する操作を基本とする計算方式である。Headは、計算機科学の形式言語理論の枠組みを用いて、この方式を数学的に定式化した。今、記号列(配列)が定義されるあるアルファベットを Σ とし、 Σ 上の二つの記号列を x, y とする。スプライシング規則 r による x と y のスプライシングとは、

- (1) スプライシング規則 r により決定される位置で x と y をそれぞれ切断する。
- (2) x の接頭語と y の接尾語、 x の接尾語と y の接頭語をそれぞれ結合する。

これにより新しく二つの記号列が生成される。 Σ 上のスプライシング規則 r は、 $\alpha_1 \# \beta_1 \$ \alpha_2 \# \beta_2$ と表される。ここで、 $\alpha_1, \beta_1, \alpha_2, \beta_2$ はそれぞれ Σ 上の記号列、 $\#, \$$ は Σ に属さない特殊記号である。これにより、スプライシング規則 $r = \alpha_1 \# \beta_1 \$ \alpha_2 \# \beta_2$ による x と y のスプライシングは、形式的に次のように書かれる：

$$(x, y) \rightarrow_r (z, w) \iff \begin{aligned} x &= x_1 \alpha_1 \beta_1 x'_1, \\ y &= y_2 \alpha_2 \beta_2 y'_2, \\ \text{かつ} \\ z &= x_1 \alpha_1 \beta_2 y'_2, \\ w &= y_2 \alpha_2 \beta_1 x'_1 \end{aligned}$$

$\alpha_1 \beta_1, \alpha_2 \beta_2$ の部分は、スプライシングにおけるサイトとよばれる。また、 x の中に $\alpha_1 \beta_1$ が1か所以上、 y の中に $\alpha_2 \beta_2$ が1か所以上現れる場合には、どのサイトを選ぶかは非決定的であり、結果として、 x と y のスプライシングは、一つ以上のペア (z, w) を含む集合となる。

形式的にスプライシングシステムとは⁽⁹⁾、記

号列(公理とよばれる)の集合 A とスプライシング規則の集合 R からなり、次のように言語(記号列の集合)を生成する。まず初めに、公理の集合の中の任意の二つの記号列に R 中の一つのスプライシング規則を適用して、新しく二つの記号列を生成する。次に、このようにして生成されたすべての新しい記号列の集合と公理の集合に対してスプライシング規則を適用して、また新たに記号列の集合を生成する。この操作を繰り返すことにより得られるすべての有限の長さの記号列の集合が、スプライシングシステムにより生成される言語となる。スプライシングシステムは、形式的に四つ組み $\gamma = (\Sigma, T, A, R)$ で表される。ここで、 T は終端記号とよばれる記号からなる Σ の部分集合である。この基本的スプライシングシステムにいくつかの条件(上のような Σ 中の記号の二つの種類への区別やマルチセット性など)を加えることにより、スプライシングシステムは、有限個の公理の集合と有限個のスプライシング規則の集合から、すべての帰納的加算な言語を生成することができると証明されている^{(10), (11)}。すなわちスプライシングシステムは、現在の計算機が解くことができる計算問題はすべて計算できると証明されたわけであり、(理論計算機モデルである)チューリングマシンと同じ計算能力をもつわけである。更に万能のスプライシングシステムを構成することにより、プログラミング可能なDNAコンピュータの存在を理論的に示している⁽¹⁰⁾。万能スプライシングシステムは、万能計算(計算機におけるすべての計算命令)のための公理の集合をもち、実行されるプログラムを符号化した一つの記号列(配列)

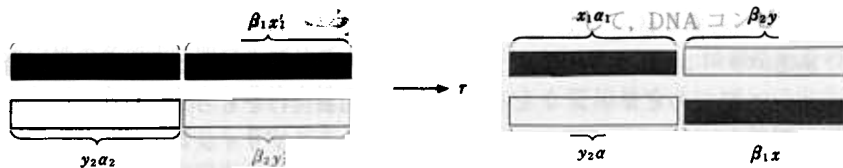


図2 スプライシング スプライシング規則 $r = \alpha_1 \# \beta_1 \$ \alpha_2 \# \beta_2$ を用いた $x = x_1 \alpha_1 \beta_1 x'_1$ と $y = y_2 \alpha_2 \beta_2 y'_2$ のスプライシング。

が入力される。公理の集合中の配列はそのコピーがいくらかでも供給され利用可能であると仮定し（これがDNA コンピュータにおける電力ならぬ消費エネルギーと考えられる）、行われる分子生物学的操作は、スプライシング（切断とリガーゼ）と配列（計算結果を表す記号列）の抽出だけとなる。

4 おわりに

DNA コンピュータの研究は、まだ始まったばかりである。世界的には、DNA コンピュータに関する国際会議 DIMACS Workshop on DNA Based Computers が過去 3 回アメリカで開催されるなど、アメリカでの研究が活発である。国内でも、平成 8 年から Molecular Computer Project が日本学術振興会の未来開拓学術研究推進事業の一つとしてスタートするなど盛んになりつつある（更に興味のある読者は、横森らによる解説⁽¹²⁾ やいくつかの web サイト (<http://www.cs.princeton.edu/~dabo>, <http://dope.caltech.edu/winfree/DNA.html>) を参照されたい)。もちろん、DNA コンピュータの前途は容易なものではない。DNA コンピュータの実現性に関して、いくつかの問題点も指摘されている。例えば、Hartmanis⁽¹³⁾ は、Adleman のモデルで現実的な大きさの問題（例えば、頂点数 200 のハミルトン経路問題）を解こうとすると、地球の重さ以上の重量の化学材料が必要になるだろうと計算している。また有田ら⁽²⁾ は、Adleman の方法による実験は、誤りを多く含みやすく、実用的な問題にまでスケールアップするのは非常に困難であると結論を出している。

いずれにしても、DNA コンピュータは「計算(コンピューティング)」というものに対して、全く新しい考え方を提案している。もしかしたら、自然はこのように「数学」を行ってきたのかもしれない。それは、足し算や引き算ではなく、切断や結合、挿入や削除といった操作を基本とする数学である。そして自然が行ってきた数学に基づく新しいコンピュータ、その一つが

DNA コンピュータなのかもしれない

文 献

- L. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol.266, pp.1021-1024, Nov. 1994.
- 有田正規, 渡辺真理, "DNA コンピューティング," 情報学プログラミング研究会研報, May 1996.
- (3) L. Adleman, "On constructing a molecular computer," (<ftp://ftp/pub/csinfo/papers/adleman/molecular-computer.ps>)
- (4) D.K. Gifford, "On the path to computation with DNA," *Science*, vol.266, pp.993-994, Nov. 1994.
- R. Lipton, "Speeding up computations via molecular biology," in *DNA Based Computers*, Baum and Lipton eds., vol.27 of DIMACS, American Mathematical Society, 1996.
- D. Beaver, "Molecular computing," (<http://www.transarc.com/~beaver/research/alternative/molecule/~molec.html>)
- D. Boneh, R. Lipton, C. Dunworth, and J. Sgall, "On the computational power of DNA," (<http://www.cs.princeton.edu/~dabo>)
- T. Head, "Formal language theory and DNA: an analysis of the generative capacity of recombinant behaviors," *Bull. Math. Biol.*, vol.49, pp.737-759, 1987.
- G. Paun, "On the splicing operation," *Discrete Appl. Math.*, vol.70, pp.57-79, 1996.
- R. Freund, L. Kari, and G. Paun, "DNA computing based on splicing: the existence of universal computers," T. Report 185-2/FR-2/95, TU Wien, Institute for Computer Languages, 1995, also (<http://www.csd.uwo.ca/~lkari>)
- (11) G. Paun, "On the power of the splicing operation," *Int. J. Comput. Math.*, vol.59, pp.27-35, 1995.
- (12) 横森 貴, 小林 聡, "計算論的言語理論と DNA 計算," *情報処理*, vol.37, no.10, pp.929-934, Oct. 1996.
- (13) J. Hartmanis, "On the weight of computations," *EATCS Bulletin*, vol.55, pp.136-138, Feb. 1995.



Lila KARI

1987 ルーマニアブカレスト大・数学・計算機科学専攻修士課程了。1991 フィンランドトゥルク大・数学・計算機科学専攻博士課程了。Ph.D. 現在、カナダウェスタンオンタリオ大計算機科学科準教授。計算の理論、計算機科学に関する離散数学と DNA コンピューティングの研究に従事。



原 康文

昭 58 東工大・理・情報科学卒。昭 60 同大学院修士課程了。昭 60-平 8 (株) 富士通研究所勤務。現在、東京電機大情報科学科助教授。博士(理学)。人工知能(特に、機械学習)、理論計算機科学、遺伝子情報処理。DNA コンピュータの研究に従事。