数理モデル化と問題解決 20－9
（1998．7．24）

最大密度部分グラフ探索アルゴリズムを用いたアミノ酸配列集合からの共通保存領域の抽出手法

竹谷 英泰，　谷 秀城，　松田 秀雄，　橋本 昭洋
{taketani, h-tani, matsuda, hasimoto}@ics.es.osaka-u.ac.jp

〒560-8531 豊中市待兼山町 1-3
大阪大学 大学院基礎工学研究科 情報数理系専攻

本稿では，多数のタンパク質アミノ酸配列の集合から共通して保存されている領域を求める手法について述べている．この手法では，各アミノ酸配列から切り出した固定長ブロックを頂点とし，それらの間の類似度を辺の重みとするグラフから最大密度部分グラフを探索することにより，共通保存領域を見つける．このため，配列ごとに共通保存領域の順番が異なっているような複雑なタンパク質ファミリに対しても共通保存領域を求めることができる．実際に複雑な領域構成をしているいくつかのタンパク質ファミリに本手法を適用してその有効性を確認した．

A Method for Extracting Common Conserved Regions from Amino Acid Sequences using Maximum-Density Subgraph Search Algorithm

Hideyasu Taketani,　Hideki Tani,　Hideo Matsuda and Akihiro Hashimoto,
Department of Informatics and Mathematical Science,
Graduate School of Engineering Science, Osaka University
1–3 Machikaneyama, Toyonaka, 560–8531

In this paper, we propose a method for extracting common conserved regions from a set of amino acid sequences that belongs to a protein family. This method extracts such regions: first, generate blocks of a fixed length from the sequences then construct a graph whose nodes are the blocks and whose edges are links between any two of the blocks with similarity scores of the two blocks; second, explore common conserved regions by searching for the maximum density subgraph from the graph. The effectiveness of our method is demonstrated by the results applying the method to two protein families that have complex structure of their conserved regions.

1　はじめに

ゲノムプロジェクトが進展するにつれて，種々の生物の持つ遺伝子に関するデータが急激に蓄積しつつある．ゲノムプロジェクトから得られた遺伝子データは，その多くが仮説的なタンパク質 (hypothetical protein) としてアミノ酸の系列（以下，アミノ酸配列と呼ぶ）の形で与えられる．その中の一部は，タンパク質データベースに登録された既知のタンパク質と，アミノ酸配列の比較により対応付けることができる．つまり，既知のタンパク質とアミノ酸配列が十分に類似している仮説タンパク質については，既知のタンパク質と同じ機能を持つと推測できる．

また，データベース中に登録された既知のタンパク質と，アミノ酸配列全体に渡っては類似性を示さない仮説タンパク質であっても，局所的に類似したアミノ酸部分配列を持つ領域（以

```
カンジタ酵母     GGTGEFEAGISKDGQTREHALLAYTLGVKQLIVAVNKMDS--VKWDKNRFEEIIKETSNF
タマホコリカビ   SPTGEFEAGIAKNGQTREHALLAYTLGVKQMIVAINKMDEKSTNYSQARYDEIVKEVSSF
ミドリムシ       STTGGFEAGISKDGQTREHALLAYTLGVKQMIVATNKFDDKTVKYSQARYEEIKKEVSGY
赤痢アメーバ     AGTGEFEAGISKNGQTREHILLSYTLGVKQMIVGVNKMDA--IQYKQERYEEIKKEISAF
マラリア病原虫   ADVGGFDGAFSKEGQTKEHVLLAFTLGVKQIVVGVNKMDT--VKYSEDRYEEIKKEVKDY
硫黄依存好熱菌   AKKGEYEAGMSAEGQTREHIILSKTMGINQVIVAINKMDLADTPYDEKRFKEIVDTVSKF
パン酵母         GGVGEFEAGISKDGQTREHALLAFTLGVRQLIVAVNKMDS--VKWDESRFQEIVKETSNF
                    * ++++ ++ ***+** +*  *+*+ * +*  ** *          *  ** ++ +
```

図 1: アミノ酸配列の多重アライメントの例

下，共通保存領域と呼ぶ）が存在することがしばしばある．共通保存領域は，元は単一であった遺伝子が進化の過程で分岐していく際に，遺伝子の配列に対してランダムに生じる変異を受けていない領域（逆にいうとその領域に変異が起きた遺伝子を持つ生物は生存できない）と考えられることから，その遺伝子がタンパク質として働くときに特に重要な部位とみなすことができる．さらに，共通保存領域は，その多くがタンパク質の立体構造で局所的に類似した部分構造（構造ドメイン）を構成していると考えられる．

これまでに知られている共通保存領域は，その多くが多重アライメント (multiple alignment) と呼ばれる処理により見つけられてきた．図 1 に多重アライメントをかけた後のアミノ酸配列の例を示す．図 1はタンパク伸長因子 (elongation factor) EF-1αと呼ばれるタンパク質のアミノ酸配列（一部）を各生物から取り出して整列させたものであり，配列の左側の列は各生物の慣用名，配列中の-はギャップをそれぞれ表している．また，一番下の行に縦の列のアミノ酸が完全に一致したとき*, 1 個を除いて一致したとき+を表示している．この例では，配列の前半から中央にかけてよく保存された領域があることが示されている．

なお，実際のアミノ酸配列の比較では，単純な文字の一致でなく，化学的性質が類似したアミノ酸同士が高く，そうでないアミノ酸同士は低く設定したスコアを定義し，このスコアを元にアミノ酸配列の類似度が設定される．PAM や BLOSUM 等のスコアが広く使用されている．

図 1で示されるような共通保存領域は，それが各タンパク質のアミノ酸配列上で同じ位置にくるときは，多重アライメントにより求めることができる．しかし，4章で述べるようにいくつかのタンパク質では，同じ共通保存領域が，あるタンパク質では配列の前半に，別のタンパク質では配列の後半にくることがある．また，まれではあるが複数の共通保存領域を持つタンパク質において，タンパク質ごとに共通保存領域の順番が異なっているケースも報告されている（4章で例を示す）．このような場合は，多重アライメントにより共通保存領域を見つけるのは困難である．

配列によって順番が異なるような場合でも共通保存領域を探索できる方法として，これまでにいくつかの手法が提案されている．これらは大きく分けて，配列間に共通してできるだけ多く出現する部分文字列を探索する方法（例えば MOST [2]）と，2 つの配列間での局所アライメントを繰り返し行ない，そこから前述のスコアに基づく類似度である閾値以上のものを結合していく方法（例えば DOMAINER[1] や DIVCLUS [3]）がある．

前者は，与えられた配列集合中の配列すべてが持つ部分文字列を見つけるには適した方法であるが，4章で示す例のように一部の配列のみが持つ部分配列を見つけるのは容易ではない．後者ではこのような場合にも対応できるが，結果が閾値の取り方に大きく依存し，かつ共通保存領域によって適当な閾値の値が異なるので，閾値を設定するのが難しいという欠点がある．

そこで，本研究では，両者の方法を組み合わせ，簡略化された局所アライメントの結果をもとに，多くの配列にまたがって強く保存されている領域から順に共通保存領域を見つけていく方法を開発した．以下では，この手法を示すとともに，従来の手法では見つけるのが困難であった共通保存領域の例に対して本手法を適用した結果について報告する．

2 最大密度部分グラフの探索

まず，グラフに関する記法について定義する．

- $G = (\boldsymbol{V}, \boldsymbol{E})$ ： 頂点集合$\boldsymbol{V}$，辺集合 $\boldsymbol{E}$ のグラフ
- (p, q) ： 頂点 p から 頂点 q への辺，無向グラフにおいては，$(p, q) = (q, p)$
- $G' \subseteq G$ ： G' は G の部分グラフ，すなわち $G = (\boldsymbol{V}, \boldsymbol{E})$，$G' = (\boldsymbol{V}', \boldsymbol{E}')$ とすると，$\boldsymbol{V}' \subseteq \boldsymbol{V}$，かつ，$\boldsymbol{E}' \subseteq \boldsymbol{E}$

- $w(p,q)$: 辺 (p,q) の重み（weight）
- $c(p,q)$: 辺 (p,q) の容量（capacity）
- $f(p,q)$: 辺 (p,q) の流量（flow）

重み集合 $\boldsymbol{w}=\{w(e)|e\in\boldsymbol{E}\}$ をもつ，重みつき無向グラフ $G=(\boldsymbol{V},\boldsymbol{E})$ が与えられたとき，その密度（density）$d(G)$ は，頂点数を $|\boldsymbol{V}|$ とすると，次のように定義される．

$$d(G)=\frac{\sum_{e\in\boldsymbol{E}}w(e)}{|\boldsymbol{V}|}$$

このとき，$G'\subseteq G : d(G')=\max_{g\subseteq G}d(g)$ なる G' を求める問題，つまり，G の部分グラフの中で，密度が最大となるグラフを求める問題を**最大密度部分グラフ探索問題**という．この問題に対して，ネットワークにおける最大輸送量問題を解くアルゴリズムの応用として，多項式時間で解くアルゴリズムが示されている [4]．以下，最大密度部分グラフ探索問題をどのように最大輸送量問題として扱うかについて述べる．

ネットワーク $N=(G,\boldsymbol{c})$ ($G=(\boldsymbol{V},\boldsymbol{E})$，$s\in\boldsymbol{V}$：ソース，$t\in\boldsymbol{V}$：シンク，$\boldsymbol{c}=\{c(e)|e\in\boldsymbol{E}\}$：容量集合）に対して，次の条件を満たすように f の値を与える．

- $f(v,w)\le c(v,w),\ (v,w)\in\boldsymbol{E}$：容量制限
- $f(v,w)=-f(w,v),\ (v,w)\in\boldsymbol{E}$ ：非対称性
- $\sum_{u\in\boldsymbol{V}}f(u,v)=0,\ v\in\boldsymbol{V}-\{s,t\}$：保存則

このとき，$F=\sum_{v\in\boldsymbol{V}}f(v,t)$ をネットワーク N の**輸送量**という．この F の最大値を求める問題を**最大輸送量問題**という．

さらに，容量 c を次のように拡張したネットワーク（パラメータ λ 付きネットワーク）に対して，最大輸送量を求める問題を**パラメータ付き最大輸送量問題**という．

- $c_\lambda(s,v)$ は，λ の非減少関数（$v\neq t$）．
- $c_\lambda(v,t)$ は，λ の非増加関数（$v\neq s$）．
- $c_\lambda(v,w)$ は，定数（$v\neq s,w\neq t$）．

パラメータ付き最大輸送量問題は，パラメータ付き preflow アルゴリズムを用いることによって，頂点数 N，辺の数 M のグラフでパラメータの個数が K のとき，データ構造としてキューを用いれば，$O(N^3+KN^2)$ の計算量で，キューに加えて動的木構造を用いれば，$O\left((N+K)M\log N^2/M\right)$ の計算量で解くことが知られている [4]．

このパラメータ付き preflow アルゴリズムを使えば，前述の最大密度部分グラフ探索問題を解くことができる．具体的には，重み付き無向連結グラフ $G=(\boldsymbol{V},\boldsymbol{E})$ に対して，$\boldsymbol{x}\in\{0,1\}^{|V|}$ とすると，G の部分グラフ G'は $\boldsymbol{x}$ の値で表せる．つまり，対応する値が 1 の頂点は G'に含まれるが，0 なら含まれないことになる．この $\boldsymbol{x}$ に対して，関数 $g(\boldsymbol{x})$ を $\boldsymbol{x}$ に対応する G'の辺の重みの総和，関数 $h(\boldsymbol{x})$ を G'の頂点数とすると，サブグラフの密度は，$g(\boldsymbol{x})/h(\boldsymbol{x})$ となる．従って，最大密度部分グラフ探索問題を，

$$\lambda(\boldsymbol{x}^*)\ =\ \max_{\boldsymbol{x}}\left\{\lambda(\boldsymbol{x})=\frac{g(\boldsymbol{x})}{h(\boldsymbol{x})}\right\}\qquad(1)$$

を満たす $\boldsymbol{x}^*$ を求める問題（分数計画問題の一種）に変換することができる [4]．

これを解くには，

$$z(\boldsymbol{x}^*,\lambda)\ =\ \max_{\boldsymbol{x}\in\boldsymbol{S}}\{z(\boldsymbol{x},\lambda)=g(\boldsymbol{x})-\lambda h(\boldsymbol{x})\}\qquad(2)$$

を満たす $\boldsymbol{x}^*$ を求める問題が使われる．

この 2 つの問題は，$\boldsymbol{x}^*$ が式（1）を満たすことと，$\lambda=\lambda^*=\lambda(\boldsymbol{x}^*)$ に対して $(\boldsymbol{x}^*,\lambda^*)$ が式（2）を満たしかつ $z(\boldsymbol{x}^*,\lambda^*)=0$ であることが，同値であることを利用して，パラメータ付き preflow アルゴリズムで解くことができる．

具体的には，G に対して，ソース s とシンク t の 2 頂点を加え，頂点 $v\in\boldsymbol{V}$との間に (s,v) と (v,t) という有向辺を引く．(s,v) と (v,t) の容量は$\delta_v-\lambda$（δ_vは vに接続されているすべての辺の重みの合計を 2 で割ったもの）とすれば，このネットワークの輸送量は G の部分グラフの頂点からの容量$\delta_v-\lambda$の辺における輸送量の合計，すなわち $g(\boldsymbol{x})-\lambda h(\boldsymbol{x})$ となり，ネットワークの最大輸送量を求めれば，最終的に部分グラフの最大密度 $g(\boldsymbol{x})/h(\boldsymbol{x})$ を求めることができる．

なお，文献 [4] では，この問題に関しては，パラメータλ の個数 Kがグラフの頂点数で抑えられる（$K=O(N)$）ことが示されている．従って，最大密度部分グラフ探索問題（頂点数 N，辺の数 M）は，動的木を使わないアルゴリズムでは $O(N^3)$，動的木を使うアルゴリズムでは $O(NM\log N^2/M)$ で解くことができる．

3 共通保存領域抽出アルゴリズム

本稿で提案する共通保存領域抽出アルゴリズムは，以下の方針に基づいている．まず，最終的に求める領域の長さは，領域の種類に依存し一定していない．また，共通保存領域中には数は少ないがギャップを含む可能性がある．しか

し，初めから，可変かつギャップを含む領域を考えることは困難なので，共通保存領域を十分短い固定長領域（これを**ブロック**と呼ぶ）の結合で近似することにした．ブロックの長さが十分短ければその中にギャップが入る可能性を少なくでき，ギャップはブロック間にのみ存在するように設定できると考えられる．

これにより，共通保存領域は，与えられたアミノ酸配列集合から切り出したブロックを頂点とし，任意の 2 個のブロックの間に辺を引き，辺の重みとして類似スコア（2 個のブロックで同じ位置のアミノ酸同士で 1章で述べたスコアを求め，それらを合計したもの）をとったグラフの中で，最大密度部分グラフを求めることで抽出できる．

なお，ブロックを連結して共通保存領域とするには，理想的には密度最大の部分グラフ以外に，密度が 2 番目，3 番目，... の部分グラフを次々に求める必要がある．しかし，実際にはこれらを効率良く求めるのは容易ではないと考えられるので，ブロック間の類似スコアに閾値を設定し，閾値以下の重みを持つ辺をグラフから取り除くことにより，元のグラフを複数個の連結な部分グラフに分割することにした．

このようにして得られた連結な部分グラフの集合があれば，それらのグラフの頂点であるブロックを，各アミノ酸配列上で結合していけば，基本的に 1 章で述べた局所アライメントに基づく共通保存領域ができることになる．しかし，単純に低い閾値でブロックを結合していくと，見かけ上弱い類似度を示すものの実際には共通保存領域ではないような領域まで拾ってしまう可能性がある．

例として図 2のようなグラフを考える．この例でグラフ G の右端の頂点は一本しか辺が引かれていない．共通保存領域は，生物学的には1章で述べたように，元は同じ遺伝子であった領域が連続的に変異して分岐していった結果生じたものであるはずなので，このように特定の 2 個のブロックのみで低い類似度しか示さない場合は偶然生じた配列類似度である可能性が高い．このような場合も，本手法では最大密度部分グラフを求めることにより，G'のように相互に類似したブロックを抽出できる．

以下，アルゴリズムを示す．

入力

n 個のアミノ酸配列の集合，ブロック長 L，スコアの閾値 $cutoff$

アルゴリズム

Step 1　入力文字列集合から，長さ L のブロックをすべて切り出す．

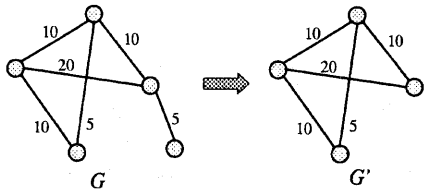

図 2: 最大密度部分グラフの例

Step 2　Step 1 で切り出したブロックのすべてのペアの類似スコアを計算し，$cutoff$ 以上となるペアをグラフ化する．

Step 3　Step 2 において出来たすべてのグラフの最大密度部分グラフを求める．

Step 4　Step 3 の結果，得られたグラフを密度の高いものから連結する．

Step 3 までを行った結果，得られたグラフの含んでいる頂点集合が，それぞれ共通保存領域を表していると考えられる．しかし，Step 3 までの段階では，ほとんどの場合，グラフの表す共通保存領域には重なりが生じている．明らかな例として，存在する共通保存領域の長さが L を超えている場合があげられる．このとき，文字が 1 つずつずれた領域を表すグラフが多数でてくる．このように，Step 3 までで得られた個々のグラフの表す領域を重なりを考慮して結合するのが Step 4 である．

Step 4 の基本的な考えは，Step 3 で得られた連結グラフを密度の高いものから順に，アミノ酸配列上の領域に戻していくことである．密度の高い順に考えるのは，密度の高いグラフの方が，低いものよりも，より確実性の高い共通保存領域を表していると考えられるからである．

以上のアルゴリズムの計算量について考察する．

入力として，長さ $O(l)$ のタンパク質アミノ酸配列が n 個与えられたとする．まず，アルゴリズムの Step 1 について．個々のアミノ酸配列から切り出されるブロックの個数は，$O(l)$ である．したがって，Step 1 の計算量は，$O(nl)$ である．

Step 2 について．Step 1 で切り出されたブロックのすべてのペアについてスコアを計算しグラフ化を行う．従って，Step 2 は，$O(n^2l^2)$ の計算量となる．

Step 3 について．2章で述べたように，頂点数 N，辺の数 M のグラフに対して，最大密度部

分グラフを求めるのに要する計算量は，動的木を使わない場合は $O(N^3)$，使う場合は $O(NM \log N^2/M)$ のアルゴリズムが知られている．本アルゴリズムでは，Step 2 で得られるグラフを調べた結果，グラフの大半が密な（すなわち $M = O(N^2)$ である）グラフであった．従って，この場合には両者のアルゴリズムの計算量は等価であるので，動的木を生成する必要のない前者のアルゴリズムを採用した．実際の Step 3 での計算量は，最悪，Step 2 で作られたグラフの $O(nl)$ 個の頂点が完全グラフを構成しているときなので，$O(n^3l^3)$ となる．

Step 4 の計算量は，Step 2 で作られるグラフの個数の 2 乗である．つまり，Step 2 で作られるグラフの最悪の個数の 2 乗 $O(n^2l^2)$ に等しい．

以上のことから，本アルゴリズムの計算量は，$O(n^3l^3)$ となる．

4 アルゴリズムの評価と考察

3章で提案したアルゴリズムの実装を行う．C 言語で記述し，NWS-5000X（MIPS R4400SC, 200MHz），NEWS-OS 4.2.1R 上で gcc-2.7.2.2 によるコンパイルを行ったものを用いて評価を行なった．また，今回の実験では，スコアマトリックスとして blosum50 を用い，固定長配列（ブロック）の長さは 20，カットオフスコアは 50 とした．アミノ酸配列は，以下の 3 つのタンパク質ファミリを選んだ．それぞれ，論文による結果（実験的な結果を含んでいる）を基準に，提案した手法（本手法とよぶ）による結果と，1章で述べた DOMAINER による結果との比較を行なった．

データ 1：Two-compenent ファミリ

まず，Two-component と呼ばれる，61 個のタンパク質ファミリに対して実験を行った．このファミリには，主に，Reciever，Transmitter，HPt と呼ばれる 3 つの共通保存領域が存在することがわかっている [5]．Reciever の長さは，アミノ酸の長さで 120 前後，Transmitter は，240 前後である．61 個のタンパク質は，この 3 つの共通保存領域のうち，どれがどの順番で存在するかによって，6 つのタイプに分けられる（図 3）．

それぞれのタイプの（）内の数字が，その個数を示している．この Two-component ファミリに対して共通保存領域抽出を行った結果が表 1である．

本手法では，Transmitter は 4 つの共通保存領域として抽出された．そのため，Transmitter

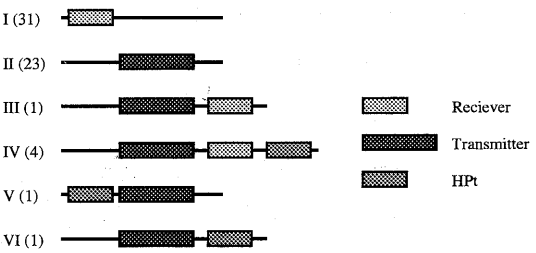

図 3: Two-component ファミリの共通保存領域

表 1: Two-component ファミリに対する実行結果

	Receiver	Transmitter	HPt
文献	36	30	6
本手法	32	20 (+9)	4
DOMAINER	30	20	0

の欄は，4 つとも抽出された配列は，20 であったが，4 つのうち 1 つでも抽出されたものを含めると 29 であったことを示している．Transmitter のように アミノ酸の長さ 240 もの長い共通保存領域になると，全体に渡って保存されていることはまれで，その共通保存領域の中に特に保存されている領域がいくつかあることが多い．この実験から，全体では共通保存領域として抽出できなくても，その特に保存されている領域の抽出が本手法でうまくできているといえる．Hpt は，文献 [5] によると，非常に弱く類似した共通保存領域であるが，その抽出も本手法ではある程度できている．実行にかかった時間は，Step 1, 2 の合計が 867.9 秒，Step 3 が 1.7 秒，Step 4 が 1.1 秒であった．

データ 2：転写制御タンパク質ファミリ

次に，32 個のタンパク質からなる，転写制御タンパク質ファミリに対して実験を行った．このファミリには，主に，4 つの共通保存領域があることがわかっている（図 4）．アミノ酸の長さ 80 程度の MGMT（以下 M），長さ 50 程度の HTH_ ARAC（以下 A），長さ 40 程度の Sugar Binding（以下 S），そして，長さ 20 に満たない HTH_ LACI （以下 L）の 4 つである．それぞれのタイプの（）内の数字が，その個数を示している．この 転写制御タンパク質ファミリに対して共通保存領域抽出を行った結果が表 2である．

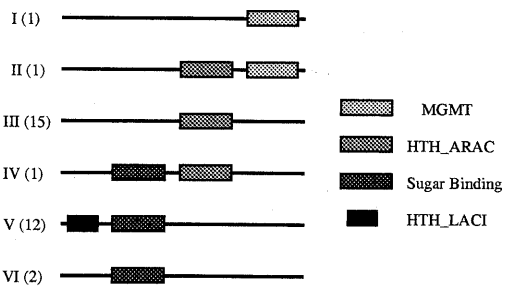

図 4: 転写制御タンパク質ファミリの共通保存領域

表 2: 転写制御タンパク質ファミリに対する実行結果

	M	A	S	L
文献	2	17	15	12
本手法	2	15	14	12
DOMAINER	2	16	14	

DOMAINER の結果では，S と L の共通保存領域領域が 1 つになってしまい，正しい結果が得られなかった．これは配列によって含まれる共通保存領域が異なっているため，このような結果になったと考えられる．一方，本手法では，S と L を別の共通保存領域としてほぼ抽出できている．実行にかかった時間は，Step 1, 2 の合計が 100.6 秒，Step 3 が 0.4 秒，Step 4 が 0.3 秒であった．

提案したアルゴリズムは，理論的な計算量では，Step 3 が最も時間がかかる（3章参照）．これは，Step 2 においてすべてのブロックが 1 つの完全グラフにまとまる時である．しかし，実際のデータにおける計算では，そういうことはありえず，これまで述べてきたように，Step 1, 2 が全実行時間のほとんどを占める．したがって，実行時間の改善は，Step 1，2 の改善にかかっているといえる．

5 おわりに

本稿では，文字列集合で共通に保存されているような領域を抽出するアルゴリズムを提案し，実装したうえで，実際のデータを適用し，評価を行った．

従来の手法では，困難であった，一部に保存されている領域や短い領域についても抽出することができることがわかった．また，グラフの密度を基準に選ぶことでより確実な領域を抽出することができるようになった．

以上より，この手法は，これからますます必要とされるアミノ酸配列集合からの共通保存領域の自動抽出に有用であると考える．

謝辞

本研究は一部，文部省科学研究費補助金重点領域研究「ゲノムサイエンス」（課題番号 08283103）によっている．

参考文献

[1] Sonnhammer, E. L. L. and Kahn, D.: Modular Arrangement of Proteins as Inferred from Analysis of Homology, *Protein Science*, Vol. 3, pp. 482–492, (1994).

[2] Tatusov, R. L., Altschul, S. F. and Koonin, E. V.: Detection of Conserved Segments in Proteins: Iterative Scanning of Sequence Databases with Alignment Blocks, *Proc. Natl. Acad. Sci. USA*, Vol. 91, pp. 12091–12095 (1994).

[3] Park, J. and Teichmann, S. A.: DIVCLUS: an Automatic Method in the GEANFAMMER package that finds homologous domains in single- and multi-domain proteins, *Bioinformatics*, Vol. 14, No. 2, pp. 144–150 (1998).

[4] Gallo, G., Grigoriadis, M. D. and Tarjan, R. E.: A Fast Parametric Maximum Flow Algorithm and Applications, *SIAM J. Comput.*, Vol. 18, No. 1, pp. 30–55 (1989).

[5] Mizuno, T.: Compilation of All Genes Encoding Two-Component Phosphotransfer Signal Transducers on the Genome of *Escherichia coli* , *DNA Research*, Vol. 4, pp. 161–168 (1997).