

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号
特開2005-190427
(P2005-190427A)

(43) 公開日 平成17年7月14日(2005.7.14)

(51) Int.Cl. ⁷	F I	テーマコード (参考)
G 0 6 F 19/00	G O 6 F 19/00	4 B O 2 4
C 1 2 Q 1/68	C 1 2 Q 1/68	4 B O 6 3
G 0 6 F 17/30	G O 6 F 17/30	5 B O 7 5
// C 1 2 N 15/09	C 1 2 N 15/00	A

審査請求 未請求 請求項の数 9 O L (全 21 頁)

(21) 出願番号	特願2003-434554 (P2003-434554)	(71) 出願人	000001007
(22) 出願日	平成15年12月26日 (2003.12.26)		キヤノン株式会社
			東京都大田区下丸子3丁目30番2号
		(74) 代理人	100090538
			弁理士 西山 恵三
		(74) 代理人	100096965
			弁理士 内尾 裕一
		(72) 発明者	塚田 護
			東京都大田区下丸子3丁目30番2号キヤ ノン株式会社内
		Fターム(参考)	4B024 AA11 AA20 CA06 HA20
			4B063 QA01 QA12 QA18 QA19 QQ41
			QR32 QR55
			5B075 ND20 UU18

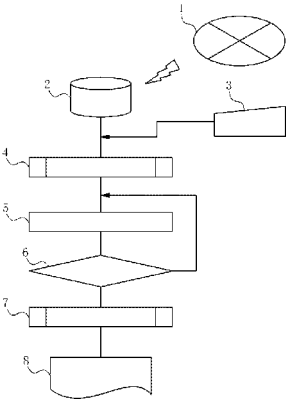
(54) 【発明の名称】 配列を同定するための変異要素のセット抽出方法

(57) 【要約】

【課題】 同定のための識別要素のセットの候補を、正確且つ効率的に取得する方法を提供する。

【解決手段】 複数の配列データにおいて特定配列を識別可能な要素（配列部位）に関する情報を取得するための情報処理方法であって、類似の配列を持つ共通領域においてアライメントされたデータを取得するステップと、前記アライメントされたデータそれぞれに同一の演算を行い、該演算結果に基づいて識別可能な要素の候補を取得するステップと、を有することを特徴とする情報処理方法。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

複数の配列データにおいて特定配列を識別可能な要素（配列部位）に関する情報を取得するための情報処理方法であって、

類似の配列を持つ共通領域においてアライメントされたデータを取得するステップと、
前記アライメントされたデータそれぞれに同一の演算を行い、該演算結果に基づいて識別可能な要素の候補を取得するステップと、
を有することを特徴とする情報処理方法。

【請求項 2】

前記演算が、アライメント位置におけるそれぞれの配列情報の要素が同じである場合に 10
、該要素を排除する演算である請求項 1 に記載の情報処理方法。

【請求項 3】

前記演算が、前記排除する演算処理がなされた後に、排除されなかった要素をもつアライメント位置の要素同士をさらに演算するものである請求項 2 に記載の情報処理方法。

【請求項 4】

前記演算が、少なくとも一つ以上の他の配列データとは異なる要素を有するアライメント位置のそれぞれの要素と、選択された要素とを対比し、同じまたは異なる要素を排除する演算と、

前記排除する演算により取得された第一のアライメント位置のそれぞれの配列要素と、
前記演算により取得された第二のアライメント位置のそれぞれの配列要素と、をさらに対 20
比する演算である請求項 1 に記載の情報処理方法。

【請求項 5】

前記得られた候補の評価を行うステップを更に有することを特徴とする請求項 1 ～ 4 のいずれかに記載の情報処理方法。

【請求項 6】

前記演算が、前記アライメントされた個々の配列情報（ b_m ）を行の情報として配置し、
前記アライメント位置情報を n 列の情報として配置した行列の集合情報 P として保持し、
該行列における任意に選ばれた二つの列の同一行の要素に対して行われる演算である請求項 1 ～ 5 のいずれかに記載の情報処理方法。

【請求項 7】

請求項 1 ～ 6 の方法を用いて導き出される変異要素を少なくとも 1 つ以上含むことを特徴とした、プローブまたはプローブセット。 30

【請求項 8】

請求項 1 ～ 6 のいずれかに記載の方法を実現するためのコンピュータに実行させるための制御プログラム。

【請求項 9】

請求項 1 ～ 6 のいずれかに記載の方法をコンピュータに実行させるための制御プログラムを格納した記憶媒体。

【発明の詳細な説明】

【技術分野】

40

【0001】

本発明は複数の配列情報の中から、特定の配列を特徴づけている要素（例えば特定箇所の文字や記号）を取得する情報処理方法に関するものであり、好適には塩基配列やアミノ酸配列などの配列情報を処理する技術に関する。

【背景技術】

【0002】

近年 SNPs、多型解析等、異なる核酸配列をハイブリダイゼーションや PCR 等の方法によって識別、同定したいというニーズがある。そのためにはまず識別、同定するための要素を抽出し、その要素を含んだプローブ（またはプライマー）を選定しなければならない。

【0003】

従来こうした場合、例えばまったく違う種類のたんぱく質であるところのアミノ酸配列や塩基配列を比較したいなどといった場合には、核酸配列やアミノ酸配列の数が増えようともさほど問題とはならなかった。つまり個々の配列はお互いを十分に識別するだけの違いを保有していたからである。したがってこれらは一般的なアライメントツールやblastツールを使用することで、手作業でプローブ選定を遂行できた。しかし異種生物間で同じたんぱく質の識別をしたいとか、菌体の属、種、株の識別、人ではHLA領域の多型など、似たもの同士をうまく識別しようとした場合、こうしたツールを使ったとしても作業は次第に鬱陶しく、煩雑なものとなる。

【0004】

さらに対象となる複数の配列が非常に似た配列同士の場合、たいてい一箇所の変異のみでその配列を同定することは不可能で、アライメント位置の異なる、数箇所にわたる変異のセットをうまく抽出して初めてその配列の同定が可能となることが多い。

【0005】

しかも対象となる配列が格納されたデータベースにおけるデータ量は年々増大しつつ、従来の方法では上記の変異セットの抽出はますます困難になってきた。

【0006】

また、たとえば特許文献1のように、既知の生体高分子の塩基配列を、生体高分子の種類によらず同一の配列をとる共通領域と変異を含む変異領域とに分け、決定された共通領域と変異領域に対して別々に補助プローブを設計するアルゴリズムは公開されているが、該プローブ設計のアルゴリズムは、未知の遺伝子（またはDNA断片）を捕獲するための複数の補助プローブを設計するものであり、類似の配列を全て識別しうるプローブセットをできる限り少数の変異箇所の同定によって抽出するものではなかった。

【特許文献1】特開2003-038160号公報

【発明の開示】

【発明が解決しようとする課題】

【0007】

本発明は、この様な背景技術に鑑みてなされたものであり、同定のための識別要素のセットの候補を正確かつ効率的に取得する方法を提供する。

【課題を解決するための手段】

【0008】

すなわち、本発明に係る情報処理方法は、複数の配列データにおいて特定配列を識別可能な要素（配列部位）に関する情報を取得するための情報処理方法であって、

類似の配列を持つ共通領域においてアライメントされたデータを取得するステップと、前記アライメントされたデータそれぞれに同一の演算を行い、該演算結果に基づいて識別可能な要素の候補を取得するステップと、を有することを特徴とする。

【0009】

また、前記演算が、アライメント位置におけるそれぞれの配列情報の要素が同じである場合に、該要素を排除する演算であるとよい。

【0010】

また、前記演算が、前記排除する演算処理がなされた後に、排除されなかった要素をもつアライメント位置の要素同士をさらに演算するものであるとよい。

【0011】

また、前記演算が、少なくとも一つ以上の他の配列データとは異なる要素を有するアライメント位置のそれぞれの要素と、選択された要素とを対比し、同じまたは異なる要素を排除する演算と、前記排除する演算により取得された第一のアライメント位置のそれぞれの配列要素と、前記演算により取得された第二のアライメント位置のそれぞれの配列要素と、をさらに対比する演算であるとよい。

【0012】

10

20

30

40

50

また、前記演算が、前記アライメントされた個々の配列情報 (b_m) を行の情報として配置し、前記アライメント位置情報を n 列の情報として配置した行列の集合情報 P として保持し、該行列における任意に選ばれた二つの列の同一行の要素に対して行われる演算であるとよい。

【0013】

また上記の方法を用いて導き出される変異要素を少なくとも1つ以上含むプローブまたはプローブセット、方法を実現するためのコンピュータに実行させるための制御プログラム、上記方法をコンピュータに実行させるための制御プログラムを格納した記憶媒体、をも本発明は包含する。

【発明の効果】

10

【0014】

本発明によって、類似する複数の配列における特異部分の取得を例えばコンピューター処理により行なうことが可能となるため、従来は不可能であった多数の類似した配列から特定の配列を同定するための識別要素のセットを正確且つ効率的に抽出することができる。

【0015】

さらにアライメント処理がなされた複数の配列データに対して演算処理が行なわれるために、同一要素の部位の排除処理が容易であり、処理にかかる時間を短縮することが可能となる。

【0016】

20

また、得られた候補の評価を更に行なうことで、より少ない数のセットの抽出や、より正確な同定を可能とするプローブの選定を容易に行なうことが可能となる。

【発明を実施するための最良の形態】

【0017】

以下、本発明を詳細に説明する。

【0018】

本発明に係る変異要素のセット抽出方法の1実施形態として、以下のステップを有する。

(1) 共通領域においてアライメントされたデータを取得するステップ。

(2) 前記アライメントされたデータに所定の演算を行い、該演算結果に基づいて識別可能な要素の候補を取得するステップ。

30

【0019】

本発明において、共通領域においてアライメントされたデータとは、後述のアライメント処理を行なった上で得られるデータに基づいて取得されるものである。共通領域とは、対象となる複数配列の所定の領域において、たとえば塩基配列において一般的に用いられる共通領域のような、同じまたは非常に似た配列（相同性の高い配列）を有する領域であるが、異なる部位（要素）をそれぞれが互いに含む領域であることが好ましい。

【0020】

図2は本発明の一実施形態にかかる情報処理方法（識別可能な要素を抽出する抽出方法）を実現するための情報処理装置の構成を示すブロック図である。

40

【0021】

本発明に係る抽出方法は、外部記憶装置9、中央処理装置（CPU）10、メモリ11、入出力装置12から構成される装置において実現される。外部記憶装置9は、本実施形態にかかる抽出方法を実現するプログラムや、配列データを保持する。また、本実施形態によって導かれた抽出結果を保持する機能を持つ。中央処理装置（CPU）10は本発明の抽出方法のプログラムを実行したり、すべての装置の制御を行ったりする。メモリ11は中央処理装置（CPU）10で処理されるプログラム、及びサブルーチンやデータを一時的に記録する。入出力装置12は、ユーザとのインタラクションを行う。また、プログラムの実行トリガーは入出力装置12を介してユーザが入力する。さらに、ユーザは入出力装置12を介して判定結果を見たり、プログラムのパラメータを設定したりすることが

50

できる。

【0022】

上記のシステムを用いて、配列が塩基配列であった場合について、識別可能な要素を抽出する方法を具体例を挙げて説明する。

【0023】

(1) 共通領域においてアライメントされたデータを取得するステップ。

まず、共通領域においてアライメントされたデータを取得する。対象となる複数の配列データに対してそれぞれアライメント処理を行い、共通領域においてアライメント処理がされた配列情報データを取得する。この共通領域は、予め範囲を指定しても良いし、アライメント処理が行なわれた後で、共通領域部分のデータのみを抽出してもよい。

10

【0024】

塩基配列のアライメント方法に関しては、すでに公知の技術をそのまま使うことができる。「Elizabeth C. Tyler, Martha R. Horton, and Philip R. Krause 1991. A Review of Algorithms for Molecular Sequence Comparison. COMPUTERS AND BIOMEDICAL RESEARCH 24:72-96」などの文献を参照して自作しても良いし、DNASIS（日立ソフト（株））や、遺伝子解析ソフトGENETYX（ソフトウェア開発（株））などの市販ソフト、ロックフェラー大学のサイト <http://cs.rockefeller.edu/index.php?page=toolkit-align> にはウェブ上で実行可能なマルチプルアライメントのツールAMAS、BoxShadeなど、いくつかまとめて紹介されているので、本発明でこれらを使用することができる。

20

【0025】

また、更に最初に識別しようとする複数の塩基配列を選び出すステップを該アライメントするステップの前に設けても良い。

【0026】

選びだされた複数の配列に対して上記手段でアライメントすることで、その共通領域は次のように行列を使って表現することが可能となる。

【0027】

【数1】

30

$$P = \begin{pmatrix} b_{11} & b_{12} & \cdots & \cdots & \cdots \\ \cdot & b_{22} & & & \\ \cdot & & b_{33} & & \\ \cdot & & & b_{mi} & b_{mj} \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{pmatrix}$$

【0028】

40

また、この共通領域の範囲は、任意に設定してよい。

【0029】

(行列の定義方法)

ここでPは、ある任意の塩基配列mをもった文字列の行ベクトルbをたとえば

$b_m = (A, T, T, T, C, G, G, T, A, A, \cdots)$

のように定義するとき、行 $m = 1, 2, 3 \cdots$ はそれぞれの塩基配列に対応し、

$\{P \in b_1, b_2, b_3, \cdots\}$

このとき定まるアライメント位置 i における列ベクトル b_i (列 $i = 1, 2, 3 \cdots$) も同じく

$\{P \in b_1, b_2, b_3, \cdots\}$

50

である。なおここでは個々の配列情報を行、アライメント位置情報を列として定義したが、行と列を入れ替えて、以降すべての取り扱いを逆にしても差し支えない。

【0030】

(要素の抽出方法)

今アライメントされた配列数が4つである場合について、本発明の具体例を示す。次のように、いずれの行から見ても、その行の要素と異なる要素が少なくとも1つ以上、同一列内に含まれるところをわかりやすく□で囲んだ。

【0031】

【数2】

$$P = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{pmatrix} \text{1} & \text{2} & \text{3} & \text{4} & \text{5} & \text{6} & \text{7} & \text{8} & \text{9} & \text{0} & \text{1} & \text{2} & \text{3} & \text{4} & \text{5} & \text{6} & \text{7} & \text{8} & \text{9} & \text{0} & \text{1} & \text{2} & \text{3} & \text{4} \\ \text{T} & \text{A} & \text{C} & \text{C} & \text{T} & \text{G} & \text{G} & \text{A} & \text{C} & \text{A} & \text{G} & \text{A} & \text{T} & \text{A} & \text{C} & \text{T} & \text{T} & \text{C} & \text{C} & \text{A} & \text{T} & \text{G} & \text{A} & \text{C} \\ \text{T} & \text{A} & \text{C} & \text{C} & \text{T} & \text{G} & \text{G} & \text{A} & \text{C} & \text{A} & \text{G} & \text{A} & \text{T} & \text{C} & \text{C} & \text{T} & \text{T} & \text{C} & \text{C} & \text{A} & \text{T} & \text{A} & \text{A} & \text{C} \\ \text{T} & \text{A} & \text{C} & \text{C} & \text{T} & \text{G} & \text{G} & \text{A} & \text{C} & \text{A} & \text{G} & \text{A} & \text{T} & \text{C} & \text{C} & \text{T} & \text{T} & \text{C} & \text{C} & \text{A} & \text{T} & \text{A} & \text{A} & \text{C} \\ \text{T} & \text{A} & \text{G} & \text{C} & \text{T} & \text{G} & \text{G} & \text{A} & \text{C} & \text{A} & \text{G} & \text{A} & \text{T} & \text{A} & \text{C} & \text{T} & \text{T} & \text{C} & \text{C} & \text{A} & \text{T} & \text{G} & \text{A} & \text{C} \end{pmatrix}$$

10

【0032】

それ以外のところは見てのとおり、もともと変異要素が存在しないため、所定の演算を行っても何も変わらない。したがってこの後の演算はこの□で囲んだところを対象に行えば良く、演算処理を簡略化することができる。

20

次に、抽出された□の列のみに対して、演算処理を行なう。

【0033】

所定の演算として例えば、 b_1 行の配列を演算対象の配列として選び、アライメント位置22の列ベクトルから左方向に順に、定義した演算にしたがって演算結果を得ていくものであると良い。

【0034】

(より具体的な演算方法)

上記に示した演算方法をより具体的に説明する。本実施形態においては、「異なる要素を識別排除する処理」を行なう。

30

【0035】

「異なる要素を識別排除する処理」とは、例えば対象とするアライメント位置において、比較する配列を b_1 行の要素とし、他の b_2 、 b_3 、 b_4 行の配列を識別排除する処理であり、

【0036】

【数3】

$$\begin{pmatrix} \text{G} \\ \text{A} \\ \text{A} \\ \text{G} \end{pmatrix} \text{を} \begin{pmatrix} \text{G} \\ - \\ - \\ \text{G} \end{pmatrix} \cdots \textcircled{1} \quad \begin{pmatrix} \text{A} \\ \text{C} \\ \text{C} \\ \text{A} \end{pmatrix} \text{を} \begin{pmatrix} \text{A} \\ - \\ - \\ \text{A} \end{pmatrix} \cdots \textcircled{2} \quad \begin{pmatrix} \text{C} \\ \text{C} \\ \text{C} \\ \text{G} \end{pmatrix} \text{を} \begin{pmatrix} \text{C} \\ \text{C} \\ \text{C} \\ - \end{pmatrix} \cdots \textcircled{3}$$

40

とするものである。ここでは排除された要素を記号－を用いて表してある。記号は本来何でも構わない。ここでは、 b_1 行の要素（1ではG、2ではA、3ではC）と異なるものは、この演算で識別される。

【0037】

上記の演算により得られた列ベクトルに対して、他の異なる列ベクトルと演算処理する

50

ことで、さらに「異なる要素を」を排除していく。

【0038】

この場合の演算結果は、排除されないのは二つの列ベクトルの b_1 行が、いずれも排除に相当しない（両方とも－でない）ときのみで、この演算を記号@を使って表すと、

【0039】

【数4】

$$\begin{pmatrix} G \\ - \\ - \\ G \end{pmatrix} @ \begin{pmatrix} A \\ - \\ - \\ A \end{pmatrix} = \begin{pmatrix} A \\ - \\ - \\ A \end{pmatrix} \dots ④ \qquad \begin{pmatrix} A \\ - \\ - \\ A \end{pmatrix} @ \begin{pmatrix} C \\ C \\ C \\ - \end{pmatrix} = \begin{pmatrix} C \\ - \\ - \\ - \end{pmatrix} \dots ⑤$$

10

となる。

【0040】

右辺の演算結果は、4、5式いずれも要素が存続しているか否かを表しているもので、この例では左辺第二項の要素で置き換えて表示しているが、本来どのような記号であってもかまわない。

【0041】

さて、 b_1 行の条件を満たした列ベクトル、列22、列14、列3を逐次選択し、4から5の演算@を順次行った結果、演算結果の要素はただ1つの要素のみが残ることを示す最小値1に達することができた。

【0042】

この1という値は、同定のための変異要素のセットの抽出に成功した、つまり自ら (b_1) と同じ配列は存在せずP内のどの配列とも分離同定可能であること意味する。また候補数は演算開始の初期値から最小値に達するまでの間、2 (1の処理結果) → 2 (4の処理結果) → 1 (5の処理結果) と変化し、候補の数を減少させることのできた列は、初期値を含めると列22と列3であるため、変異要素のセットは列22のGと列3のC、すなわち以下の下線で示されたところとなる。

30

$$b_1 - \text{set} = (T A \underline{C} C T G G A C A G A T A C T T C C A T \underline{G} A C)$$

次に同様に、 b_2 行の配列に基づいて識別排除する処理を行なう。

同様に演算@を行うと、

【0043】

【数5】

$$\begin{pmatrix} - \\ A \\ A \\ - \end{pmatrix} @ \begin{pmatrix} - \\ C \\ C \\ - \end{pmatrix} = \begin{pmatrix} - \\ C \\ C \\ - \end{pmatrix} \dots ⑥ \qquad \begin{pmatrix} - \\ C \\ C \\ - \end{pmatrix} @ \begin{pmatrix} C \\ C \\ C \\ - \end{pmatrix} = \begin{pmatrix} - \\ C \\ C \\ - \end{pmatrix} \dots ⑦$$

40

となって最小値が1とならず、2に終わる。そこで要素の存続した行を集め、お互いを比較すると、

$$b_2 - \text{set} = (T A C C T G G A C A G A T C C T T C C A T \underline{A} A C)$$

$$b_3 - \text{set} = (T A C C T G G A C A G A T C C T T C C A T \underline{A} A C)$$

選ばれたこの b_2 、 b_3 の領域にはもともとお互いを識別できる要素がなかったことが判定できる。お互いを比較する方法は、プログラミング言語上では文字列比較関数や、1文字ごとの論理比較を行えばよい。尚演算によって初期値として選択された列22の A

50

は少なくとも b_1 、 b_4 と識別可能なことを示しており、無駄とはならない。

【0044】

さらに、 b_4 行についても同様に演算②を行うと以下のようになり、

【0045】

【数6】

$$\begin{pmatrix} G \\ - \\ - \\ G \end{pmatrix} \textcircled{2} \begin{pmatrix} A \\ - \\ - \\ A \end{pmatrix} = \begin{pmatrix} A \\ - \\ - \\ A \end{pmatrix} \cdots \textcircled{8} \qquad \begin{pmatrix} A \\ - \\ - \\ A \end{pmatrix} \textcircled{2} \begin{pmatrix} - \\ - \\ - \\ G \end{pmatrix} = \begin{pmatrix} - \\ - \\ - \\ G \end{pmatrix} \cdots \textcircled{9}$$

10

【0046】

【数7】

$$b_{4\text{-set}} = (T A \underline{G} C T G G A C A G A T A C T T C C A T \underline{G} A C) \cdots \textcircled{10}$$

【0047】

ここでもしも一番左にある識別排除されていない列を開始行として、順次右方向にある識別排除されていない列に対して演算を行っていくことにより得られる結果は、

$$b_{4\text{-set}} = (T A \underline{G} C T G G A C A G A T A C T T C C A T \underline{G} A C) \cdots 11$$

20

となり、より少ない要素数で変異要素のセットを表現できる。

ちなみに

【0048】

【数8】

$$b_{1\text{-set}} = (T A \underline{C} C T G G A C A G A T A C T T C C A T \underline{G} A C)$$

$$b_{2\text{-set}} = (T A C C T G G A C A G A T C C T T C C A T \underline{A} A C)$$

$$b_{3\text{-set}} = (T A C C T G G A C A G A T C C T T C C A T \underline{A} A C)$$

$$b_{4\text{-set}} = (T A \underline{G} C T G G A C A G A T A C T T C C A T \underline{G} A C)$$

30

であるから、右方向に演算を行ったものでも、お互いを識別することは可能である。

【0049】

即ち演算（及び評価）の実施形態は左右の方向問わない。また隣同士順番に行わなくても、端から行わなくても良い。たとえば一回の演算結果の候補数が、たえず最小であるように選択しながら最小値に達するよう、アルゴリズムを工夫しても良い。

【0050】

もう少し簡単に、たとえば初期値のみ候補数が最小となるよう、列ベクトルを選んでおき、あとは単純に右または左方向から列ベクトルの演算を行うように工夫しても構わない。

【0051】

こうした工夫の特徴はなるべく少ない変異要素のセットで、その配列を特徴づけることができる点にある。この点は、DNAチップ用のプローブとして用いる場合、配置プローブ数をなるべく少なくすることができるために好ましい。

40

【0052】

逆にこのような工夫をしない特徴は、変異要素のセットがやや多くの要素で表現されるために、

i アッセイのさまざまな条件に対して、やや丈夫であるということ、

ii 順番に演算を行えばなるべく狭い範囲に変異要素のセットを探せるかもしれないこと

、

iii アッセイが定量PCR法であれば、3' → 5' 方向の演算（例では左方向）とすれ

50

【数 1 1】

10

【 0 0 6 0 】

20

【 0 0 6 1 】

【数 1 2】

30

b_{2-set} = (T A C C T G G A C A G A T * C T T C C A T A A C)

【 0 0 6 2 】

40

【 0 0 6 3 】

【数 1 3】

$$\begin{array}{c}
 \text{アライメント位置:} \\
 P = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 & 1 & 2 & 3 & 4 \end{matrix} \\ \begin{pmatrix} \text{T A C C T G G A C A G} | \text{A T A C T T C C A T G A C} \\ \text{T A C C T G G A C A G} | \text{A T * C T T C C A T A A C} \\ \text{T A C C T G G A C A G} | \text{A T C C T T C C A T A A C} \\ \text{T A G C T G G A C A G} | \text{A T A C T T C C A T G A C} \end{pmatrix} \end{array}
 \end{array}$$

10

【0 0 6 4】

つまり本発明で定義するところの演算は、記号の種類や数に左右されることなく、さらに記号を増やして20個の記号で表現されるアミノ酸一文字表記であっても、また塩基配列のIUB（国際生化学連合）表記であっても全く問題ない。

【0 0 6 5】

（演算結果の評価）

本発明においては、取得した識別要素に対して、評価を行なう工程を更に有するとよい。

【0 0 6 6】

得られた識別要素からたとえばハイブリダイゼーションのプローブを選定するのであれば、欠失部分を中央部分においた、欠失記号を取り除いた配列をプローブに設定すれば、 T_m 値が他とは大きく異なるプローブが得られる。 20

【0 0 6 7】

実際このように抽出された、同定のための変異要素のセットは、DNAマイクロアレイ上でのハイブリダイゼーションや、定量PCR法のプローブ選定にとって必要な情報となる。ハイブリダイゼーションの場合、上記同定のための変異要素がほぼプローブの中心部にくるように20塩基前後（1塩基ミスマッチ検出の場合）の配列を切り出したほうが、フルマッチとミスマッチの T_m 値に大きな差が付きやすく、反応温度をうまく両者の境界に設けることによって、分離、同定がなされやすい。定量PCR法の場合、耐熱性ポリメラーゼが鎖を伸長しないよう、3'末端側に上記同定のための変異要素がくるよう20塩基前後の配列を切り出したほうが分離、同定がなされやすい。 30

【0 0 6 8】

両方法ともミスマッチをなるべく数多くプローブ内に抱えたほうが、分離同定がなされやすいプローブとなることは言うまでもない。

【0 0 6 9】

すなわち、変異要素のセットを求めるにあたって、プローブ内に含まれるミスマッチの個数を規制したり、アミノ酸レベルの変異があることをプローブ候補（または変異要素のセット）の条件としたり、することによって、本発明で定義するところの「所定の評価」で可能である。

【0 0 7 0】

本実施形態においては、演算によって存続した（排除されなかった）要素を抽出し、その複数の要素を候補において、その候補数を減少させることを目指して逐次他の列を選択し、演算を行っている。 40

【0 0 7 1】

このときの演算は、前回の演算結果と次選択の列ベクトルを演算する、という具合に、いわば論理的な積のみで候補数を減少させることができる。

【0 0 7 2】

（演算結果の評価の他の実施形態）

本実施形態においては、さらに効率的に候補数を減少させるための評価ステップとして、部分的和を考慮して評価をおこなっている。 50

【0073】

以下の例は列16から18にかけて、GAT (A s p) がAGC (S e r) に変異している。また見やすいように各行ベクトルをコドン単位でまとめて表記した。

【0074】

【数14】

$$\begin{array}{c} \text{アライメント位置:} \\ P = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} = \begin{array}{ccccccccc} & & & 1 & & 2 & & 3 & & \\ & & & 123 & 456 & 789 & 012 & 345 & 678 & 901 & 234 & 567 & 890 \\ \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{pmatrix} & = & \begin{pmatrix} \text{GAG CTG GGG} & \text{CGG CCT} & \text{GAT} & \text{GCC GAG TAC TGG} \\ \text{GAG CTG GGC} & \text{CGG CCT} & \text{AGC} & \text{GCC GAG TAC TGG} \\ \text{GAG CTG GGA} & \text{CGG CCT} & \text{AGC} & \text{GCC GAG TAC TGG} \\ \text{GAG CTG GGG} & \text{CGG CCT} & \text{AGC} & \text{GCC GAG TAC TGG} \end{pmatrix} \end{array} \end{array}$$

10

アミノ酸レベルの変異を評価条件にしたいのであれば、この場合アミノ酸1文字コードでベクトルを表示しなおして演算を行えば、変異要素のセットは求まる（ただし例では b_2 、 b_3 、 b_4 は識別できない。列9の1塩基変異はアミノ酸が変わるまでには至らない）。

【0075】

また今までの論理的な積（前回の演算結果に対して次選択の列ベクトルを演算するという過程）のみでは、右から左方向に順次計算すると b_2 、 b_3 、 b_4 では列18の要素が選択され、列17、列16では候補数を減少させることができないため、候補には残らない。しかし論理的な和（前回の演算結果に対して次選択の列ベクトルを演算しない、列と列はそれぞれ独立に演算するという過程）を考慮すると、 b_2 では、

20

【0076】

【数15】

$$\begin{array}{ccccccc} \text{列18} & & & \text{列17} & & & \text{列16} \\ \begin{pmatrix} - \\ C \\ C \\ C \end{pmatrix} @ \begin{pmatrix} - \\ C \\ - \\ - \end{pmatrix} = \begin{pmatrix} - \\ C \\ - \\ - \end{pmatrix} \cdots \textcircled{15} & \begin{pmatrix} - \\ G \\ G \\ G \end{pmatrix} @ \begin{pmatrix} - \\ C \\ - \\ - \end{pmatrix} = \begin{pmatrix} - \\ C \\ - \\ - \end{pmatrix} \cdots \textcircled{16} & \begin{pmatrix} - \\ A \\ A \\ A \end{pmatrix} @ \begin{pmatrix} - \\ C \\ - \\ - \end{pmatrix} = \begin{pmatrix} - \\ C \\ - \\ - \end{pmatrix} \cdots \textcircled{17} \end{array}$$

30

列16、列17、列18の演算結果はいずれも等価（残った要素数およびその行位置が同じ）であることがわかる。

【0077】

この例の場合、「隣合う列の演算結果が三回連続して等価であれば、候補として残す」という評価条件を与えれば、アミノ酸レベルの変異を抽出できることになる。

【0078】

また同様に「連続する20mer以内に二つの変異を含むこと」を評価条件に与えれば、たとえば b_3 では

40

$$b_{3-s e t} = (\text{GAG CTG GGA CGG CCT AGC GCC GAG TAC TGG}) \quad (\text{右方向へ演算})$$

となるであろうし、分離同定のためのプローブとして、より優れたものとなるよう、評価条件を設定してなんら問題ない。もしも評価条件によって、最終的に要素数が1にならなくとも、条件を順次緩和して、要素数が1に到達するよう、再帰的アルゴリズムを用意すればよい。

【0079】

また本発明の18のような形態は、実際にハイブリダイゼーションが行われる検体が、複数の似通った配列の混合物である場合、検体によるアッセイ結果の違いをシミュレート

50

するのに都合がよい。

【0080】

たとえば実際の例として、人のMHC領域のDRB抗原を分離同定するDNAマイクロアレイを設計して、実検体を用いてのハイブリダイゼーションによって、その設計したアレイのプロブ一つ一つに問題がなかったかどうかを確認する作業は、並大抵でない。現在DRBアレルの数は400種類近く報告され、検体は父方と母方のDNAの混合物として存在する。

【0081】

しかも運悪い場合、PCR後のサンプルで、たとえばDRB1とDRB3を含む4種類の混合物となることもある。仮に100アレルの2種類の混合物として、 $100C_2 = 4950$ 通りとなり、すべての検体において性能はどうかということを、とても実検体では評価しきれない。

【0082】

しかし18のような形態は、変異要素のセットの表現が

$b_{1-se} = (-AC-----A-T-----G-)$

となるため、自動的に貴重な情報を付加してくれる。たとえば定量PCR用のプロブとして末端にミスマッチを含むものを、 T_m が53℃を超えないように選んだとする。 b_0 の配列を参照して比較を行い、-記号を塩基に置き換える際に、小文字を使用すると、

【0083】

【数16】

$b_{0-part} = (TGGACAGATGCATCCATA)$

$b_{1-part} = (- - - - - A - T - - - - - G)$ より、

$b_{1-prob} = (t g g a c a g a t A c T t c c a t G)$ (T_m 計算はWallac法とした)

【0084】

ここで、 b_{1-prob} 内の大文字のAおよびTは、 b_0 から見てミスマッチであるものの、変異を識別する要素のセットとはならなかったものである。しかし b_{1-prob} プロブ自体には合計3つのミスマッチを含み、1つのミスマッチであった場合に比べると、それほど悪いプロブではないことが読み取れる。さらに b_0 を変更して、検体中に考えられる例えば4種類のアレルについて、同様の検討をしてみる。さらに考え得る種々の検体で、どの検体のとき、どのプロブに起因して、どのアレルの分離同定が難しいか、演算結果を参照して容易に検討することができる。

【0085】

(実現例)

上述の方法を具体的に実現するシステムについて更に詳細に説明する。図1のフローは本発明の1実施形態を表すもので、インターネット1を介して(またはもともと存在して)遺伝子情報配列のデータベースがネットワークドライブ2上にある。

【0086】

複数の配列の選択がキーボード3を操作するオペレーターによって行われる。

【0087】

選択された複数の配列に対して、アライメント処理ルーチン4によってアライメントが行われる。アライメント後、同定のための変異要素セットを探す範囲(アライメント後の共通領域)が、オペレーターのキーボード3入力情報に基づいて行われ、演算ルーチン5によって演算がなされる。このとき演算の評価条件6(同様にオペレーターから与えても、予めプログラムされていてもよい。)に基づいて評価され、所定の結果(最小値)に到達するまで演算が試行される。試行によって存続した要素(評価条件をパスした)について、同じくオペレーターのキーボード3入力情報に基づき、プロブ抽出ルーチン7によってプロブが決定され、決定されたプロブ情報が出力8される。

【0088】

10

20

30

40

50

次に具体的に、人のMHCのDRBにおけるアレルの分離、同定のための変異要素のセットおよびそのプローブセットを抽出する例を挙げる。データベースから選ばれたアレル群は日本人によく見られる認定済みのアレルでその集合をPとすると、

【0089】

【数17】

P = { DRB1*010101 DRB1*010102 DRB1*030101 DRB1*030102 DRB1*040101
 DRB1*040102 DRB1*040301 DRB1*040302 DRB1*0404 DRB1*040501
 DRB1*040502 DRB1*040503 DRB1*040504 DRB1*0406 DRB1*040701
 DRB1*040702 DRB1*0410 DRB1*070101 DRB1*070102 DRB1*080201
 DRB1*080202 DRB1*080203 DRB1*080302 DRB1*0809 DRB1*090102
 DRB1*100101 DRB1*100102 DRB1*110101 DRB1*110102 DRB1*110103
 DRB1*110104 DRB1*120101 DRB1*120102 DRB1*120201 DRB1*120202
 DRB1*130101 DRB1*130102 DRB1*130103 DRB1*130201 DRB1*130202
 DRB1*130701 DRB1*130702 DRB1*140101 DRB1*140102 DRB1*1402
 DRB1*1403 DRB1*140501 DRB1*140502 DRB1*1406 DRB1*140701
 DRB1*140702 DRB1*1412 DRB1*1429 DRB1*150101 DRB1*150102
 DRB1*150103 DRB1*150104 DRB1*150201 DRB1*150202 DRB1*150203
 DRB1*160201 DRB1*160202 DRB3*010101 DRB3*01010201 DRB3*01010202
 DRB3*010103 DRB3*010104 DRB3*020201 DRB3*020202 DRB3*020203
 DRB3*020204 DRB3*030101 DRB3*030102 DRB4*0102 DRB4*01030101
 DRB4*010302 DRB4*010303 DRB4*010304 DRB5*010101 DRB5*010102
 DRB5*0102 DRB5*0202 }

10

20

なお、この処理方法の実行条件として、

i 同定のための変異要素セットを探す範囲をexon2の113から345までとする。

30

ii 3'末端に変異要素をもつプローブとする。

iii iiのプローブはTmが55℃を超えないものとする。

iv DRB1*010101に対する変異を大文字で表す。
 とした。次に出力結果を示す。

【0090】

(出力結果)

【0091】

【数 1 8】

Result of PROBE EXTRACTOR programed by Tsukada Mamoru, Canon Inc., Japan.

This is Program to investigate and to discriminate Allel Sequence in which includes SNPs.

Set of oligonucleotide sequences can prove that the allele is Unique by this set.

note)Each probe has direction 5' to 3' ,and is designed for using Real Time PCR method.

Capital letter indicates position of SNPs viewing from DRB1*010101.

If next number from allele name is 'I', indicates that the allele is Unique by this set.

10

OUTPUT FILE: search from 113 to 345

DRB1*010101	1	0	1	2	3	4	5
DRB1*010102	1	6					
DRB1*030101	1	7	8				
DRB1*030102	1	7	9	8			
DRB1*040101	1	10	11				
DRB1*040102	1	12					
DRB1*040301	1	10	13	14	8		
DRB1*040302	1	10	15	14	8		
DRB1*0404	1	10	13	8			
DRB1*040501	3	10	16				
DRB1*040502	1	17					
DRB1*040503	3	10	16				
DRB1*040504	3	10	16				
DRB1*0406	1	10	14	8			
DRB1*040701	1	10	13	14			
DRB1*040702	1	18					
DRB1*0410	1	16	8				
DRB1*070101	1	19					
DRB1*070102	1	20	19				
DRB1*080201	2	21	22	23			

20

30

【 0 0 9 2 】

【数 1 9】

DRB1*080202	2	21	22	23			
DRB1*080203	1	24					
DRB1*080302	1	16	23				
DRB1*0809	1	25	22	23			
DRB1*090102	1	26					
DRB1*100101	1	27					
DRB1*100102	1	28	29				10
DRB1*110101	2	30					
DRB1*110102	2	30					
DRB1*110103	1	30	31				
DRB1*110104	1	32	30				
DRB1*120101	1	33	34	35			
DRB1*120102	1	33	34				
DRB1*120201	1	33	22	35			
DRB1*120202	1	33	22	31			20
DRB1*130101	1	36	37	8			
DRB1*130102	1	38					
DRB1*130103	1	37	35	8			
DRB1*130201	1	36	37				
DRB1*130202	1	37	39				
DRB1*130701	1	40	41	21	22	42	
DRB1*130702	1	41	15	22	42		
DRB1*140101	1	43	44	8			
DRB1*140102	1	43	8				30
DRB1*1402	1	45	46	47	48		
DRB1*1403	1	45	48	49	23		
DRB1*140501	1	50	51				
DRB1*140502	1	50					
DRB1*1406	1	45	46	47	48	8	
DRB1*140701	2	43	44				
DRB1*140702	2	43	44				
DRB1*1412	1	23	8				
DRB1*1429	1	47	49	52			40
DRB1*150101	1	53	15	54	8		
DRB1*150102	1	55	56				
DRB1*150103	1	54	35	8			

【0 0 9 3】

【数 2 0】

DRB1*150104	1	53	54	8	
DRB1*150201	1	53	15	54	
DRB1*150202	1	53	54		
DRB1*150203	1	57			
DRB1*160201	1	58	31		
DRB1*160202	1	58	59		
DRB3*010101	3	60	61	7	9
DRB3*01010201	3	60	61	7	9
DRB3*01010202	3	60	61	7	9
DRB3*010103	1	62	7	9	
DRB3*010104	1	9	55		
DRB3*020201	1	63	64		
DRB3*020202	1	63			
DRB3*020203	1	65			
DRB3*020204	1	63	66	64	
DRB3*030101	1	67	64	8	
DRB3*030102	1	68			
DRB4*0102	1	69			
DRB4*01030101	2	70			
DRB4*010302	2	70			
DRB4*010303	1	70	44		
DRB4*010304	1	71			
DRB5*010101	1	72	31		
DRB5*010102	1	72			
DRB5*0102	1	73	74	31	
DRB5*0202	1	73	75		

---Reference Number of Oligonucleotides---

0 atc tat aac caa gag gag TC
 1 g ctg ggg cgg cct gaT
 2 g ggg cgg cct gaT gcC
 3 gg aac agc cag aag gac C
 4 gac Ctc ctg gag cag AG
 5 cac aac tac ggg gTt gGT
 6 gc atc tat aac caa gag gaA
 7 ag cag aAg cgg gGc CG

【0 0 9 4】

【数 2 1】

8 cac aac tac ggg gtt gTG
 9 gg gGc CGg gtg gac aAT
 10 C ctg gaC aga tAc Ttc tat C
 11 g gac etc ctg gag cag aA
 12 aga tAc Ttc tat Cac caa gaA
 13 Ttc tat Cac caa gag gag tA
 14 ag cag agg cgg gcc gA
 15 g ggg cgg cct gaC gcT
 16 ag ctg ggg cgg cct AG
 17 caa gag gag tAc gtg cgG
 18 gac etc ctg gag cag agA
 19 gg gcg gtg acg gag ctA
 20 t ttc Ctg tgg cag GGt aaA
 21 c Ttc tat aac caa gag gag tA
 22 gg aac agc cag aag gac T
 23 gaA GaC agg cgg gcc CT
 24 ag tac cgg gcg gtg acA
 25 tc Cat aac caG gag gag tT
 26 cg gag cgg gtg cgg tAT
 27 c ctg gag cGg agg cgT
 28 gg ttg ctg gaa aga Cgc G
 29 c ctg gag cGg agg cgC
 30 ctg ggg cgg cct gat gA
 31 tc ctg gaA GaC agg cgC
 32 Ct gaG tgt cat ttc ttc aaC
 33 tc Cat aac caG gag gag C
 34 gg aac agc cag aag gac A
 35 cc gcg gtg gac acc taT
 36 ag gac Atc ctg gaA GaC G
 37 g gac Atc ctg gaA GaC GA
 38 tg gaA GaC GAg cgg gcT
 39 tc ctg gaA GaC GAg cgC
 40 tc ttg GAg TaC TCt aCg tC
 41 cgg ttC ctg gaC aga tAc T
 42 g aag gac Ttc ctg gaA GaC
 43 cgg cct gCi gcG gag C

10

20

30

40

【0 0 9 5】

【数 2 2】

44	gcc gAg gtg gac acc taT	
45	cgt ttc ttg GAg TaC TCt aC	
46	gg gtg egg ttC ctg gaG	
47	aG aga tAc Ttc Cat aac caG	
48	Ttc Cat aac caG gag gag A	
49	tc Cat aac caG gag gag AA	10
50	C TCt aCg tCt gaG tgt caA	
51	g ggg egg cct gat gcT	
52	c aga cac aac tac ggg gC	
53	agc gac gtg ggg gag tT	
54	gac Atc ctg gag cag GC	
55	c tgc aga cac aac tac ggA	
56	a cac aac tac ggA gtt gTG	
57	ctg gaC aga tAc Ttc tat aaT	20
58	Ctg tgg cag cCt aag AGG	
59	aag gac ctc ctg gaA GaC	
60	g ctg ggg egg cct gTC	
61	gg cct gTC gcc gag tC	
62	egg cct gTt gcc gag tC	
63	gag tac cgg gcg gtg aG	
64	gg gGc CAg gtg gac aAT	
65	g gtg gac aAT tac tgc agG	30
66	g ggg egg cct gat gcG	
67	gag cag aAg cgg gGc CA	
68	tgc aga cac aac tac ggC	
69	g cgg gcc gAg gtg gG	
70	t aag tGt gaG tgt cat ttc C	
71	gCg cgc tAc Aac agT gaT	
72	c tat aac caa gag gag GAc T	
73	g egg ttC ctg CaC aga G	40
74	c atc tat aac caa gag gag A	
75	cac aac tac ggg gCt gTG	

【0 0 9 6】

(他の実施形態)

なお、本発明は、複数の機器（例えばホストコンピュータ、インタフェイス機器、リーダ、プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置など）に適用してもよい。

【0 0 9 7】

また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。

【0098】

この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【0099】

プログラムコードを供給するための記憶媒体としては、例えば、フロッピー（登録商標）ディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0100】

また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0101】

さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【産業上の利用可能性】

【0102】

本発明の方法は、配列を同定するための変異要素のセットを効率的に抽出することができるので、人為的なミスがなく、膨大な配列種から特定の要素を抽出することができる。

【0103】

この結果、DNA分析や診断に使用するDNAプローブの選択に利用することができる。

【図面の簡単な説明】

【0104】

【図1】本発明の1実施形態を表すフロー図である。

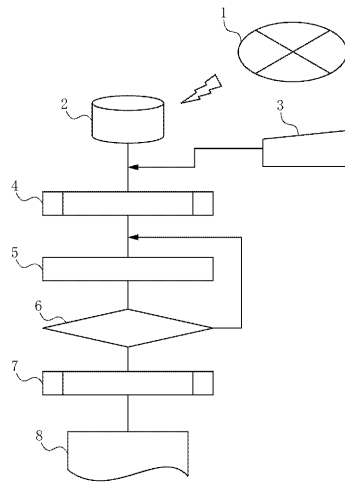
【図2】本発明の一実施形態にかかる情報処理方法（識別可能な要素を抽出する抽出方法）を実現するための情報処理装置の構成を示すブロック図である。

【符号の説明】

【0105】

- 1 インターネット
- 2 ネットワークドライブ
- 3 キーボード
- 4 アライメント処理ルーチン
- 5 演算ルーチン
- 6 演算の評価条件
- 7 プローブ抽出ルーチン
- 8 出力
- 9 外部記憶装置
- 10 中央処理装置（CPU）
- 11 メモリ
- 12 入出力装置

【図 1】



【図 2】

