(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2024-169825
(P2024-169825A)
(43)公開日　令和6年12月6日(2024.12.6)

(51)Int.Cl.		FI		テーマコード（参考）
G16B 15/00	*(2019.01)*	G16B	15/00	5B175
G16B 20/00	*(2019.01)*	G16B	20/00	
G06F 16/332	*(2019.01)*	G06F	16/332	

審査請求 未請求　請求項の数 19　ＯＬ　（全 27 頁）

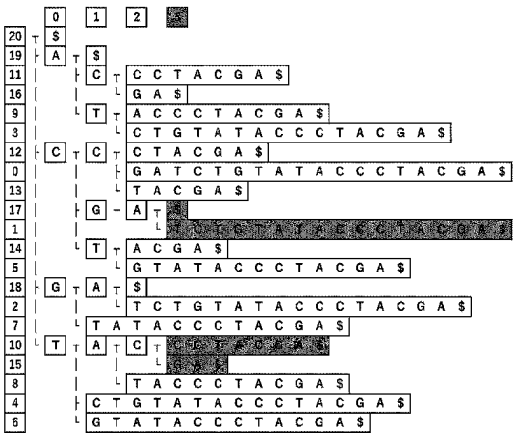
(21)出願番号　特願2023-86595(P2023-86595)
(22)出願日　令和5年5月26日(2023.5.26)

(71)出願人　720008140
先端加速システムズ株式会社
神奈川県横浜市金沢区泥亀一丁目２８番Ｂ－１３１２号
(71)出願人　501293666
株式会社ダナフォーム
神奈川県横浜市鶴見区鶴見中央２丁目６番２９号
(74)代理人　100112689
弁理士　佐原　雅史
(74)代理人　100128934
弁理士　横田　一樹

最終頁に続く

(54)【発明の名称】インデックス作成用コンピュータプログラム、マッピング用コンピュータプログラム、インデックス作成用コンピュータ装置、マッピング用コンピュータ装置、階層的インデックスのデータ構

(57)【要約】
【課題】　参照データを用いて、与えられるクエリデータの解析を高速及び／又は効率的に行う。
【解決手段】　本発明は、クエリ文字列に基づいて参照文字列における部分文字列を同定する際に利用させるインデックスを作成することを実現するインデックス作成用コンピュータプログラムであって、参照文字列の接尾辞配列を階層化した接尾辞木における最上位階層から下位に向かう階層数を所定値に設定することで、当該所定値よりも下位側の階層データを省略するステップを有するようにした。
【選択図】　図７



【特許請求の範囲】
【請求項１】
コンピューティングデバイスに、クエリ文字列に基づいて参照文字列における部分文字列を同定する際に利用されるインデックスを作成させることを実現するインデックス作成用コンピュータプログラムであって、
前記参照文字列の接尾辞配列を階層化した接尾辞木における最上位階層から下位に向かう階層数を所定値に設定することで、当該所定値よりも下位側の階層データを省略するステップを有することを特徴とする、
インデックス作成用コンピュータプログラム。
【請求項２】
前記階層数の前記所定値が１０００以下に設定されることを特徴とする、
請求項１に記載のインデックス作成用コンピュータプログラム。
【請求項３】
前記階層数の前記所定値が、前記クエリ文字列の文字数以上に設定されることを特徴とする、
請求項１に記載のインデックス作成用コンピュータプログラム。
【請求項４】
更に、前記接尾辞木を途中階層で分割することによって、前記途中階層よりも上位階層を含む上位木と、前記途中階層よりも下位階層を含み且つ前記上位木の下位側に繋がる複数の下位木（以下、下位木群と称する）を生成するステップを有することを特徴とする、
請求項１に記載のインデックス作成用コンピュータプログラム。
【請求項５】
前記下位木のデータ量が、前記コンピューティングデバイスにおけるキャッシュメモリに格納可能な値に設定されることを特徴とする
請求項４に記載のインデックス作成用コンピュータプログラム。
【請求項６】
前記下位木のデータ量が、１０ＭＢ以下に設定されることを特徴とする、
請求項４に記載のインデックス作成用コンピュータプログラム。
【請求項７】
コンピューティングデバイスに、クエリ文字列に基づいて参照文字列における部分文字列を同定する際に利用されるインデックスを作成させることを実現するインデックス作成用コンピュータプログラムであって、
前記参照文字列の接尾辞配列を階層化した接尾辞木を途中階層で分割することによって、前記途中階層よりも上位階層を含む上位木と、前記途中階層よりも下位階層を含み且つ前記上位木の下位側に繋がる複数の下位木（以下、下位木群と称する）を作成するステップを有することを特徴とする、
インデックス作成用コンピュータプログラム。
【請求項８】
コンピューティングデバイスに、クエリ文字列に基づいて、参照文字列における部分文字列を同定する方法を実現させるためのマッピング用コンピュータプログラムであって、
請求項１～７のいずれか一項のインデックス作成用コンピュータプログラムを一部に含み、
更に前記方法は、
前記クエリ文字列の少なくとも一部と一致する前記参照文字列における部分文字列を同定するために、前記階層的インデックスを参照して、前記参照文字列に対する前記クエリ文字列のマッピングを行うことを含むことを特徴とする、
マッピング用コンピュータプログラム。
【請求項９】
コンピューティングデバイスに、クエリ文字列に基づいて、参照文字列における部分文字列を同定する方法を実現させるためのマッピング用コンピュータプログラムであって、

前記参照文字列に戻づいて生成される階層的インデックスは、途中階層で分割されることで、前記途中階層よりも上位階層を含む上位木と、前記途中階層よりも下位階層を含み且つ前記上位木の下位側に繋がる複数の下位木（以下、下位木群と称する）と、を有してなり、
前記方法は、
前記クエリ文字列の少なくとも一部と一致する前記参照文字列における部分文字列を同定するために、前記階層的インデックスを参照して、前記参照文字列に対する前記クエリ文字列のマッピングを行うこととを含み、
前記クエリ文字列のマッピングを行うことは、
前記上位木を利用して複数の前記クエリ文字列の照合処理を行うことで、各々の前記クエリ文字列が、前記下位木群におけるどの前記下位木に属するかを判別する判別ステップと、
前記判別ステップによって判別された特定の前記下位木に属する複数の前記クエリ文字列について、特定の前記下位木を利用した照合処理を行うことで、前記参照文字列に対する前記クエリ文字列の出現位置を決定する出現位置決定ステップと、を有することを特徴とする、
マッピング用コンピュータプログラム。
【請求項１０】
前記下位木における前記出現位置決定ステップを並列処理することを特徴とする、
請求項９に記載のマッピング用コンピュータプログラム。
【請求項１１】
前記判別ステップを並列処理することを特徴とする
請求項９に記載のマッピング用コンピュータプログラム。
【請求項１２】
前記出現位置決定ステップにおいて、前記下位木が、前記コンピューティングデバイスにおけるキャッシュメモリに展開されることを特徴とする、
請求項９に記載のマッピング用コンピュータプログラム。
【請求項１３】
請求項１～７のいずれか一項のインデックス作成用コンピュータプログラムを一部に含むことを特徴とする、
請求項９に記載のマッピング用コンピュータプログラム。
【請求項１４】
記憶装置を含むコンピューティングデバイスを有しており、クエリ文字列に基づいて、参照文字列における部分文字列を同定する際に用いるインデックスを作成するインデックス作成用コンピュータ装置であって、
前記コンピューティングデバイスは、
前記記憶装置に記憶される前記参照文字列を読みだして、該参照文字列の接尾辞配列を階層化した接尾辞木における最上位階層から下位に向かう階層数を所定値に設定し、当該所定値よりも下位側の階層データが省略される部分接尾辞木を生成する部分接尾辞木生成手段を有することを特徴とする、
インデックス作成用コンピュータ装置。
【請求項１５】
記憶装置を含むコンピューティングデバイスを有しており、クエリ文字列に基づいて、参照文字列における部分文字列を同定する際に用いるインデックスを作成するインデックス作成用コンピュータ装置であって、
前記コンピューティングデバイスは、
前記記憶装置に記憶される前記参照文字列の接尾辞配列を階層化した接尾辞木を途中階層で分割することによって、上位階層を含む上位木と、前記上位木の下位側に繋がる複数の下位木（以下、下位木群と称する）を作成する接尾辞木分割手段を有することを特徴とする、

インデックス作成用コンピュータ装置。
【請求項１６】
記憶装置を含むコンピューティングデバイスを有しており、クエリ文字列に基づいて、参照文字列における部分文字列を同定するマッピング用コンピュータ装置であって、
前記記憶装置には、前記参照文字列に戻づいて生成される階層的インデックスが途中階層で分割されて生成された、前記途中階層よりも上位階層を含む上位木、及び、前記途中階層よりも下位階層を含み且つ前記上位木の下位側に繋がる複数の下位木（以下、下位木群と称する）が保存されており、
前記コンピューティングデバイスは、
前記クエリ文字列の少なくとも一部と一致する前記参照文字列における部分文字列を同定するために、前記上位木及び前記下位木を参照して、前記参照文字列に対する前記クエリ文字列のマッピングを行うマッピング手段を有しており、
前記マッピング手段は、
前記記憶装置に記憶される前記上位木を参照して複数の前記クエリ文字列の照合処理を行うことで、各々の前記クエリ文字列が、前記下位木群におけるどの前記下位木に属するかを判別する判別手段と、
前記判別手段によって判別された特定の前記下位木に属する複数の前記クエリ文字列について、前記記憶装置に記憶される特定の前記下位木を参照して照合処理を行うことで、前記参照文字列に対する前記クエリ文字列の出現位置を決定する出現位置決定手段と、を有することを特徴とする、
マッピング用コンピュータ装置。
【請求項１７】
コンピューティングデバイスに、クエリ文字列に基づいて参照文字列における部分文字列を同定する方法を実現させるためのマッピング用コンピュータプログラムで用いられる、前記参照文字列の階層的インデックスのデータ構造であって、
前記参照文字列の接尾辞配列を階層化した接尾辞木であり、かつ、最上位階層から下位に向かう階層数が所定値に設定されており、当該所定値よりも下位側の階層データが省略されていることを特徴とする、
階層的インデックスのデータ構造。
【請求項１８】
前記階層的インデックスは、前記接尾辞木が途中階層で分割されることにより、前記途中階層よりも上位階層を含む上位木のデータ片と、前記途中階層よりも下位階層を含み且つ前記上位木の下位側に繋がる複数の下位木のデータ片を有することを特徴とする、
請求項１６に記載の階層的インデックスのデータ構造。
【請求項１９】
コンピューティングデバイスに、クエリ文字列に基づいて、参照文字列における部分文字列を同定する方法を実現させるためのマッピング用コンピュータプログラムで用いられる、前記参照文字列の階層的インデックスのデータ構造であって、
前記階層的インデックスは、途中階層で分割されることにより、前記途中階層よりも上位階層を含む上位木のデータ片と、前記途中階層よりも下位階層を含み且つ前記上位木の下位側に繋がる複数の下位木のデータ片を有することを特徴とする、
階層的インデックスのデータ構造。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、ゲノムの解析に好適なインデックス作成用コンピュータプログラム、マッピング用コンピュータプログラム、これらのプログラムが実行される装置、これらのプログラムに利用される階層的インデックスのデータ構造に関する。
【背景技術】
【０００２】

ヒトゲノムは、人が持つ遺伝情報のセットであり、これを担っている物質が、約３０億対の塩基が連なったＤＮＡ（デオキシリボ核酸）である。塩基は、アデニン（Ａ）、グアニン（Ｇ）、シトシン（Ｃ）、及びチミン（Ｔ）がある。すなわち、人の遺伝情報は、これらの塩基の並び（配列）によって決定される。

【０００３】

ヒトゲノムの読み取りにはシークエンサと称される装置が用いられる。シークエンサは、サンプルとなるヒトゲノムを読み取って、これを所定の上限値（数百塩基対程度）に細断して増幅し、数億個となる遺伝情報のデータ片からなる膨大なデータ配列として出力する。現行のシークエンサは、一人分のヒトゲノムを１時間ほどで読み出すことができる。

【０００４】

シークエンサにより読み出されるばらばらのデータ片は、人の標準的なゲノム配列として定められたヒトゲノム参照配列と比較されることによって、元の長さのヒトゲノム配列に再構築され解析される。例えば、各データ片が、ヒトゲノム参照配列との比較において、どの位置にあるかが調べられ（マッピング）る。また、マッピング完了後は、どのような変異があるかといった解析がなされる。通常、シーケンサから読み出されるデータ片には誤差が含まれるため、１サンプルあたりヒトゲノム配列一人分の例えば３０倍の冗長データを用いて統計処理を行うことにより誤差を小さくしている。したがって、ヒトゲノム配列の解析には膨大な計算量が必要とされるため、典型的には、スーパーコンピュータやクラスタコンピュータ、ＦＰＧＡ（Field-Programmable Gate Array）ベースのコンピュータといった高性能なコンピュータが用いられる。

【０００５】

データ片のマッピングには、例えばＢＷＡ（Burrow-Wheeler Aligner）といったプログラムツールが用いられる。ＢＷＡは、３つのアルゴリズム、ＢＷＡ－ｂａｃｋｔｒａｃｋ、ＢＷＡ－ＳＷ、及びＢＷＡ－ＭＥＭから構成される。このうち、ＢＷＡ－ＭＥＭは、Indel（挿入欠失）に対応した高速アルゴリズムとして広く利用されている。ＢＷＡ－ＭＥＭは、読み出したデータ片に基づくクエリ文字列のうち、ヒトゲノム参照配列に繰り返し現れる部分に対して、接尾辞配列を用いてインデックスを作成し、マッピングを行うアルゴリズムである（非特許文献１）。

【０００６】

なお、マッピングされたデータ片は、ヒトゲノム参照配列における同位置の部分配列と照合され、どのような変異があるかが解析される。ヒトゲノム参照配列の部分配列とデータ片との照合には、例えば、アラインメントアルゴリズムが用いられる。アラインメントアルゴリズムでは、動的計画法に従って、アラインメント表と称される配列の各要素について変異度（類似度）を算出しながら、近似文字列を同定する。

【先行技術文献】

【非特許文献】

【０００７】

【非特許文献１】Heng Li, "Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM", May 26, 2013, arXiv:1303.3997 (q-bio.GN)

【発明の概要】

【発明が解決しようとする課題】

【０００８】

上述した従来のＢＷＡ－ＭＥＭは、ギャップを許容しかつ高速マッピングが可能なアルゴリズムとして広く利用されているが、参照配列から作成されるインデックスのサイズが非常に大きくなるため、大量のメモリリソースを必要とするという問題があった。例えば、ヒトゲノムの解析の場合、ヒトゲノム参照配列のインデックスのデータ量は膨大となる。このインデックスは、より高速アクセス可能なキャッシュメモリに一度に収容しきないため、外部メモリ（メインメモリ）に格納しておく必要がある。結果、探索するクエリ文字列の文字数（すなわち、塩基数）に応じた回数だけ、外部メモリへのアクセスが必要となり、プロセッサの処理速度がメモリへのアクセス時間により律速されてしまうという問

題があった。
【０００９】
より具体的には、ＢＷＡ－ＭＥＭでは、シークエンサによって断片化状態で読み出されたクエリ文字列群が、ヒトゲノム参照配列内に出現する位置を同定するために、メインメモリへのランダムアクセスが、クエリ文字列群の総文字数分の回数だけ実行される。ここで、メインメモリへの１回のランダムアクセスの待ち時間（アクセスが発生してから実際にデータが得られるまでの時間）は、現在のコンピュータでは、１バンクあたり約１μ秒（０．０００００１秒）かかっている。したがって、３０億塩基対のヒトゲノムの解析の場合、冗長性を３０とすると、総文字数に対する総アクセス時間Ｔは、
Ｔ＝３，０００，０００，０００×０．０００００１×３０＝９０，０００（秒）
となる。つまり、ヒトゲノムの解析において、メインメモリへのランダムアクセスだけで、約９万秒（約２５時間）かかることになる。このため、たとえ、高性能なプロセッサを備えたコンピュータを用いたとしても、メモリアクセス時間が制約となって、マッピング時間を短縮するには限界があった。
【００１０】
本発明の一つの目的は、マッピング処理に好適な、参照文字列に基づく階層的インデックスを作成する技術を提供することである。また、本発明の一つの目的は、例えば、シークエンサ等によって読み出されたデータ片（クエリ文字列）について、参照配列内に出現する位置を同定するマッピング処理を高速及び／又は効率的に行う技術を提供することである。
【課題を解決するための手段】
【００１１】
上記課題を解決するための本発明は、以下に示す発明特定事項乃至は技術的特徴を含んで構成される。
【００１２】
すなわち、上記課題に関連する本発明は、コンピューティングデバイスに、クエリ文字列に基づいて参照文字列における部分文字列を同定する際に利用されるインデックスを作成させることを実現するインデックス作成用コンピュータプログラムであって、前記参照文字列の接尾辞配列を階層化した接尾辞木における最上位階層から下位に向かう階層数を所定値に設定することで、当該所定値よりも下位側の階層データを省略するステップを有することを特徴とする、インデックス作成用コンピュータプログラムである。
【００１３】
上記インデックス作成用コンピュータプログラムに関連して、前記階層数の前記所定値が１０００以下に設定されることを特徴としてもよい。
【００１４】
上記インデックス作成用コンピュータプログラムに関連して、前記階層数の前記所定値が、前記クエリ文字列の文字数以上に設定されることを特徴としてもよい。
【００１５】
上記インデックス作成用コンピュータプログラムに関連して、更に、前記接尾辞木を途中階層で分割することによって、前記途中階層よりも上位階層を含む上位木と、前記途中階層よりも下位階層を含み且つ前記上位木の下位側に繋がる複数の下位木（以下、下位木群と称する）を生成するステップを有することを特徴としてもよい。
【００１６】
上記インデックス作成用コンピュータプログラムに関連して、前記下位木のデータ量が、前記コンピューティングデバイスにおけるキャッシュメモリに格納可能な値に設定されることを特徴としてもよい。
【００１７】
上記インデックス作成用コンピュータプログラムに関連して、前記下位木のデータ量が、１０ＭＢ以下以下に設定されることを特徴としてもよい。
【００１８】

上記課題に関連する本発明は、コンピューティングデバイスに、クエリ文字列に基づいて参照文字列における部分文字列を同定する際に利用されるインデックスを作成させることを実現するインデックス作成用コンピュータプログラムであって、前記参照文字列の接尾辞配列を階層化した接尾辞木を途中階層で分割することによって、前記途中階層よりも上位階層を含む上位木と、前記途中階層よりも下位階層を含み且つ前記上位木の下位側に繋がる複数の下位木（以下、下位木群と称する）を作成するステップを有することを特徴とする、インデックス作成用コンピュータプログラムである。

【００１９】

上記課題に関連する本発明は、コンピューティングデバイスに、クエリ文字列に基づいて、参照文字列における部分文字列を同定する方法を実現させるためのマッピング用コンピュータプログラムであって、上記のインデックス作成用コンピュータプログラムを一部に含み、更に前記方法は、前記クエリ文字列の少なくとも一部と一致する前記参照文字列における部分文字列を同定するために、前記階層的インデックスを参照して、前記参照文字列に対する前記クエリ文字列のマッピングを行うことを含むことを特徴とする、マッピング用コンピュータプログラムである。

【００２０】

上記課題に関連する本発明は、コンピューティングデバイスに、クエリ文字列に基づいて、参照文字列における部分文字列を同定する方法を実現させるためのマッピング用コンピュータプログラムであって、前記参照文字列に戻づいて生成される階層的インデックスは、途中階層で分割されることで、前記途中階層よりも上位階層を含む上位木と、前記途中階層よりも下位階層を含み且つ前記上位木の下位側に繋がる複数の下位木（以下、下位木群と称する）と、を有してなり、前記方法は、前記クエリ文字列の少なくとも一部と一致する前記参照文字列における部分文字列を同定するために、前記階層的インデックスを参照して、前記参照文字列に対する前記クエリ文字列のマッピングを行うこととを含み、前記クエリ文字列のマッピングを行うことは、前記上位木を利用して複数の前記クエリ文字列の照合処理を行うことで、各々の前記クエリ文字列が、前記下位木群におけるどの前記下位木に属するかを判別する判別ステップと、前記判別ステップによって判別された特定の前記下位木に属する複数の前記クエリ文字列について、特定の前記下位木を利用した照合処理を行うことで、前記参照文字列に対する前記クエリ文字列の出現位置を決定する出現位置決定ステップと、を有することを特徴とする、マッピング用コンピュータプログラムである。

【００２１】

上記マッピング用コンピュータプログラムに関連して、前記下位木における前記出現位置決定ステップを並列処理することを特徴としてもよい。

【００２２】

上記マッピング用コンピュータプログラムに関連して、前記判別ステップを並列処理することを特徴としてもよい。

【００２３】

上記マッピング用コンピュータプログラムに関連して、前記出現位置決定ステップにおいて、前記下位木が、前記コンピューティングデバイスにおけるキャッシュメモリに展開されることを特徴としてもよい。

【００２４】

上記マッピング用コンピュータプログラムに関連して、上記のいずれか一項のインデックス作成用コンピュータプログラムを一部に含むことを特徴としてもよい。

【００２５】

上記課題に関連する本発明は、記憶装置を含むコンピューティングデバイスを有しており、クエリ文字列に基づいて、参照文字列における部分文字列を同定する際に用いるインデックスを作成するインデックス作成用コンピュータ装置であって、前記コンピューティングデバイスは、前記記憶装置に記憶される前記参照文字列を読みだして、該参照文字列の接尾辞配列を階層化した接尾辞木における最上位階層から下位に向かう階層数を所定値

に設定し、当該所定値よりも下位側の階層データが省略される部分接尾辞木を生成する部分接尾辞木生成手段を有することを特徴とする、インデックス作成用コンピュータ装置である。

【００２６】

上記課題に関連する本発明は、記憶装置を含むコンピューティングデバイスを有しており、クエリ文字列に基づいて、参照文字列における部分文字列を同定する際に用いるインデックスを作成するインデックス作成用コンピュータ装置であって、前記コンピューティングデバイスは、前記記憶装置に記憶される前記参照文字列の接尾辞配列を階層化した接尾辞木を途中階層で分割することによって、上位階層を含む上位木と、前記上位木の下位側に繋がる複数の下位木（以下、下位木群と称する）を作成する接尾辞木分割手段を有することを特徴とする、インデックス作成用コンピュータ装置である。

【００２７】

上記課題に関連する本発明は、記憶装置を含むコンピューティングデバイスを有しており、クエリ文字列に基づいて、参照文字列における部分文字列を同定するマッピング用コンピュータ装置であって、前記記憶装置には、前記参照文字列に戻づいて生成される階層的インデックスが途中階層で分割されて生成された、前記途中階層よりも上位階層を含む上位木、及び、前記途中階層よりも下位階層を含み且つ前記上位木の下位側に繋がる複数の下位木（以下、下位木群と称する）が保存されており、前記コンピューティングデバイスは、前記クエリ文字列の少なくとも一部と一致する前記参照文字列における部分文字列を同定するために、前記上位木及び前記下位木を参照して、前記参照文字列に対する前記クエリ文字列のマッピングを行うマッピング手段を有しており、前記マッピング手段は、前記記憶装置に記憶される前記上位木を参照して複数の前記クエリ文字列の照合処理を行うことで、各々の前記クエリ文字列が、前記下位木群におけるどの前記下位木に属するかを判別する判別手段と、前記判別手段によって判別された特定の前記下位木に属する複数の前記クエリ文字列について、前記記憶装置に記憶される特定の前記下位木を参照して照合処理を行うことで、前記参照文字列に対する前記クエリ文字列の出現位置を決定する出現位置決定手段と、を有することを特徴とする、マッピング用コンピュータ装置である。

【００２８】

上記課題に関連する本発明は、コンピューティングデバイスに、クエリ文字列に基づいて参照文字列における部分文字列を同定する方法を実現させるためのマッピング用コンピュータプログラムで用いられる、前記参照文字列の階層的インデックスのデータ構造であって、前記参照文字列の接尾辞配列を階層化した接尾辞木であり、かつ、最上位階層から下位に向かう階層数が所定値に設定されており、当該所定値よりも下位側の階層データが省略されていることを特徴とする、階層的インデックスのデータ構造である。

【００２９】

上記階層的インデックスのデータ構造に関連して、前記階層的インデックスは、前記接尾辞木が途中階層で分割されることにより、前記途中階層よりも上位階層を含む上位木のデータ片と、前記途中階層よりも下位階層を含み且つ前記上位木の下位側に繋がる複数の下位木のデータ片を有することを特徴としてもよい。

【００３０】

上記課題に関連する本発明は、コンピューティングデバイスに、クエリ文字列に基づいて、参照文字列における部分文字列を同定する方法を実現させるためのマッピング用コンピュータプログラムで用いられる、前記参照文字列の階層的インデックスのデータ構造であって、前記階層的インデックスは、途中階層で分割されることにより、前記途中階層よりも上位階層を含む上位木のデータ片と、前記途中階層よりも下位階層を含み且つ前記上位木の下位側に繋がる複数の下位木のデータ片を有することを特徴とする、階層的インデックスのデータ構造である。

【００３１】

なお、本発明は、前記コンピュータプログラムを記憶した記録媒体としても成立する。また、本発明は、前記方法を実行するように構成されたハードウェア及び／又はファーム

ウェアからなる装置としても成立する。更に、前記インデックス作成用コンピュータ装置やマッピング用コンピュータ装置等が、複数のハードウェアコンポーネント（又は特定の機能を実現する機能モジュール）が論理的に協働して実現される場合、これらの装置は、インデックス作成用コンピュータシステムやマッピング用コンピュータシステムとなる。また、インデックス作成用コンピュータ装置やマッピング用コンピュータ装置等が、プロセッサモジュールで実現される場合、これらの装置は、インデックス作成用プロセッサモジュールやマッピング用プロセッサモジュールとなる。つまり各装置や機能モジュールが、単一の筐体内にあるか否かや、単一の機能部品で実現されるか否は特に問わないものである。

【００３２】

なお、本明細書等において、手段又は部（unit）とは、単に物理的手段を意味するものではなく、その手段又は部が有する機能をソフトウェアによって実現する場合も含む。また、１つの手段又は部が有する機能が２つ以上の物理的手段により実現されても、２つ以上の手段又は部の機能が１つの物理的手段により実現されても良い。

【発明の効果】

【００３３】

本発明の一つによれば、マッピング処理に好適な、参照文字列に基づく階層的インデックスを作成することができる。この階層的インデクスにより、高速及び／又は効率的なマッピング処理を実現できる。

【００３４】

また、本発明によれば、シークエンサによって読み出されたデータ片に基づくヒトゲノム参照配列を用いた解析を高速及び／又は効率的に行うことができる。

【００３５】

本発明の他の技術的特徴、目的、及び作用効果乃至は利点は、添付した図面を参照して説明される以下の実施形態により明らかにされる。本明細書に記載された効果はあくまで例示であって限定されるものではなく、また他の効果があっても良い。

【図面の簡単な説明】

【００３６】

【図１】図１は、本発明の一実施形態に係るコンピュータシステムの概略的構成の一例を示すブロックダイアグラムである。

【図２】図２は、同コンピュータシステムによる近似文字列照合処理の概略の一例を説明するフローチャートである。

【図３】図３は、同コンピュータシステムによるインデックス作成処理の一例を説明するフローチャートである。

【図４】図４は、同コンピュータシステムにおいて用いられる参照文字列から生成される巡回配列の一例を示す図である。

【図５】図５は、同巡回配列をソートした状態の一例を示す図である。

【図６】図６は、ソートされた巡回配列に基づいて生成される接尾辞木の構造の一例を示す図である。

【図７】図７は、同接尾辞木を利用した部分接尾辞木の生成過程を説明する図である。

【図８】図８は、同接尾辞木から生成される部分接尾辞木の構造の一例を示す図である。

【図９】図９（Ａ）は、同部分接尾辞木の分割過程を説明する図であり、図９（Ｂ）は分割された上位木の一例を示す図である。

【図１０】図１０は、同コンピュータシステムによるマッピング処理の一例を説明するフローチャートである。

【図１１】図１１は、同コンピュータシステムにおいて用いられるクエリ文字列の一例を示す図である。

【図１２】図１２（Ａ）～（Ｄ）は、下位木を利用した同コンピュータシステムによるマッピング処理を説明する図である。

【図１３】図１３（Ａ）～（Ｄ）は、下位木を利用した同コンピュータシステムによるマ

ッピング処理を説明する図である。
【図１４】図１４（Ａ）～（Ｄ）は、下位木を利用した同コンピュータシステムによるマッピング処理を説明する図である。
【図１５】図１５は、同部分接尾辞木の変形例を示す図である。
【図１６】図１６は、同部分接尾辞木を分割する際の変形例を説明する図である。
【図１７】図１７は、同コンピューティングデバイスのハードウェア構成の一例を示す図である。
【図１８】図１８は、同コンピューティングデバイスのハードウェア構成のプロセッサモジュールの一例を示す図である。
【図１９】図１９は、同コンピューティングデバイスの機能構成の一例を示すブロック図である。
【図２０】図２０は、同コンピュータシステムによるマッピング処理の変形例を説明するフローチャートである。
【発明を実施するための形態】
【００３７】
以下、図面を参照して本発明の実施の形態を説明する。ただし、以下に説明する実施形態は、あくまでも例示であり、以下に明示しない種々の変形や技術の適用を排除する意図はない。本発明は、その趣旨を逸脱しない範囲で種々変形（例えば各実施形態を組み合わせる等）して実施することができる。また、以下の図面の記載において、同一又は類似の部分には同一又は類似の符号を付して表している。図面は模式的なものであり、必ずしも実際の寸法や比率等とは一致しない。図面相互間においても互いの寸法の関係や比率が異なる部分が含まれていることがある。
【００３８】
図１は、本発明の一実施形態に係るコンピュータシステムの概略的構成の一例を示すブロックダイアグラムである。同図に示すように、コンピュータシステム１は、例えば、上位コンピュータ１０と、複数の下位コンピュータ２０（１）～２０（ｎ）と、データベース３０と含み構成される。上位コンピュータ１０と下位コンピュータ２０（１）～２０（ｎ）とは、所定のインタフェースを介して通信可能に接続される。本開示では、コンピュータシステム１は、上位コンピュータ１０が複数の下位コンピュータ２０（１）～２０（ｎ）を統括的に制御する中央集権型コンピュータシステムとして構成されるが、これに限られず、例えば分散型コンピュータシステムとして構成されても良い。分散型コンピュータシステムにおいては、各コンピュータが並列分散処理により協調的に動作し得るが、特定の処理に関しては、代表する一のコンピュータのみが該処理を実行する場合があっても良い。
【００３９】
上位コンピュータ１０及び下位コンピュータ２０で共通する計算ハードウェアは、図１７に例示されるコンピューティングデバイス１００となる。コンピューティングデバイス１００は、ＤＲＡＭ等の主記憶装置となるメモリモジュール１０２、中央計算機となるプロセッサモジュール１０４、プロセッサモジュール１０４と外部機器の間のブリッジとなるチップセット１０６、ＨＤＤやＳＤＤ等の内部ストレージデバイス１０８、Ｉ／Ｏコントローラ１１０、出力インタフェース１１２、入出力インタフェース１１４、通信インタフェース１１６、外付けＨＤＤ等の外部ストレージデバイス１１８を有する。
【００４０】
更に、図１８に示すように、プロセッサモジュール１０４は、互いに独立した計算処理を実行する複数のコアＡ，Ｂ，Ｃ，Ｄと、各コアＡ，Ｂ，Ｃ，Ｄに対応して設けられる内部記憶機能の第一キャッシュメモリ（Ｌ１キャッシュメモリ）Ｌ１ａ，Ｌ１ｂ，Ｌ１ｃ，Ｌ１ｄ及び第二キャッシュメモリ（Ｌ２キャッシュメモリ）Ｌ２ａ，Ｌ２ｂ，Ｌ２ｃ，Ｌ２ｄと、全コアＡ，Ｂ，Ｃ，Ｄで共有される内部記憶機能の第三キャッシュメモリ（Ｌ３キャッシュメモリ）Ｌ３ｎを有する。コアＡ，Ｂ，Ｃ，Ｄによるアクセス待ち時間は、Ｌ１キャッシュメモリＬ１ａ，Ｌ１ｂ，Ｌ１ｃ，Ｌ１ｄが最も短く、次いでＬ２キャッシュ

メモリＬ２ａ，Ｌ２ｂ，Ｌ２ｃ，Ｌ２ｄが短く、最も長いのがＬ３キャッシュメモリＬ３ｎとなる。一方で、Ｌ１，Ｌ２，Ｌ３キャッシュメモリのアクセス待ち時間は、メモリモジュール（メインメモリ）１０２よりも大幅に短い。これらの構造により、プロセッサモジュール１０４内の複数のコアＡ，Ｂ，Ｃ，Ｄ同士では、並列的にタスクを処理することができる。例えば、複数のコアＡ，Ｂ，Ｃ，Ｄのそれぞれは、与えられた個々のクエリ文字列に基づいて参照文字列との比較において解析処理を行うことができる。なお、ここではキャッシュメモリが、３段階に分割されている場合を例示しているが、本発明はこれに限定されず、１段階または２段階であってもよく、４段階以上であってもよい。以降において、第一キャッシュメモリ（Ｌ１キャッシュメモリ）Ｌ１ａ，Ｌ１ｂ，Ｌ１ｃ，Ｌ１ｄ、第二キャッシュメモリ（Ｌ２キャッシュメモリ）Ｌ２ａ，Ｌ２ｂ，Ｌ２ｃ，Ｌ２ｄ、第三キャッシュメモリ（Ｌ３キャッシュメモリ）Ｌ３ｎは、これらを総称して「キャッシュメモリ」と称することができる。本発明では、プロセッサモジュール１０４におけるコア数も特に限定されない。コア数は数個から数万個まで、さまざまに選択できる。

【００４１】

更に、図１７及び図１８に示す構成は、汎用的なハード構造を例示したが、本発明はこれに限定されない。またこれらのハード構造の全部または一部は、ＦＰＧＡ等によって個別にカスタマイズ（プログラム）されたハードウェアを採用してもよい。更に、以降において、データの記憶できるハードウェアとなる、メモリモジュール１０２、内部ストレージデバイス１０８、外部ストレージデバイス１１８、第一キャッシュメモリ（Ｌ１キャッシュメモリ）Ｌ１ａ，Ｌ１ｂ，Ｌ１ｃ，Ｌ１ｄ、第二キャッシュメモリ（Ｌ２キャッシュメモリ）Ｌ２ａ，Ｌ２ｂ，Ｌ２ｃ，Ｌ２ｄ、第三キャッシュメモリ（Ｌ３キャッシュメモリ）Ｌ３ｎは、これらを総称して「記憶装置」と称することができる。

【００４２】

図１に戻って、本開示では、下位コンピュータ２０は、上位コンピュータ１０の制御の下、並列的にタスクを処理する。例えば、複数の下位コンピュータ２０のそれぞれは、与えられた個々のクエリ文字列に基づいて参照文字列との比較において解析処理を行う。以下では、下位コンピュータ２０（１）～２０（ｎ）について、それらを特に区別する必要がない限り、単に、「下位コンピュータ２０」と表記することがある。

【００４３】

データベース３０は、上位コンピュータ１０の制御の下、各種のデータ、例えば参照文字列を格納する。一例として、データベース３０は、ヒトゲノム参照配列を格納する。ヒトゲノム参照配列は、人の標準的なゲノム配列として定められた塩基対の配列を示すデータである。また、データベース３０は、参照文字列に基づいて作成されたインデックスを格納する。インデックスは、参照文字列を探索するために用いられるある種のデータ配列構造である。なお、図中、データベース３０は、上位コンピュータ１０にのみアクセス可能に接続される構成となっているが、これに限られず、下位コンピュータ２０もまたアクセス可能に接続される構成であっても良い。また、下位コンピュータ２０も同様のデータベースを配置してもよい。ここでは、ヒトゲノム参照配列を利用する場合を例示するが、複数の生物（動物）、植物等を一まとめにしたメタゲノム参照配列を利用することもできる。

【００４４】

図２は、本発明の一実施形態に係るコンピュータシステム１による近似文字列照合処理の概略の一例を説明するフローチャートである。かかる処理は、例えば、上位コンピュータ１０及び複数の下位コンピュータ２０が、プロセッサモジュール１０２において近似文字列照合プログラムを実行することにより、他のハードウェアコンポーネントと協働し、実現される。これにより、コンピュータシステム１、または、上位コンピュータ１０、あるいは複数の下位コンピュータ２０は、図１９に示す近似文字列照合用コンピュータシステム２００として機能する。また、近似文字列照合プログラムは、複数の処理ブロック（サブルーチン）を有しており、その結果として、近似文字列照合用コンピュータシステム２００は、更に詳細には、インデックス作成用コンピュータシステム２０１と、マッピン

グ用コンピュータシステム２０２と、文字列ペア作成用コンピュータシステム２０３と、近似文字列導出用コンピュータシステム２０４を内在する。

【００４５】

図２に示すように、まず、上位コンピュータ１０は、データベース３０に格納された参照文字列を読み出し、読み出した参照文字列に基づいて、インデックスを作成する（インデックス作成処理：Ｓ２０１）。具体的には、上位コンピュータ１０は、参照文字列における部分文字列を所定の条件に従って拡張し及び／又はソートして配列化することにより、所定のデータ配列構造を有するインデックスを作成する。データ配列構造は、階層的なツリー構造からなる。本開示では、このようなインデックスを階層的インデックスと称するものとする。上位コンピュータ１０は、作成した参照文字列に基づくインデックスをデータベース３０に格納する。参照文字列に基づく階層的インデックスの作成処理の詳細は後述する。なお、参照文字列に基づく階層的インデックスが既に作成されており、データベース３０に格納されている場合には、インデックス作成処理Ｓ２０１は省略され得る。これにより、コンピュータシステム１は、インデックス作成用コンピュータシステム２０１として機能する。

【００４６】

続いて、上位コンピュータ１０は、図示しない外部装置からクエリ文字列を受信し、受信したクエリ文字列に基づいて階層的インデックスを参照し、クエリ文字列の参照文字列へのマッピング処理を実行する（マッピング処理：Ｓ２０２）。一例では、外部装置は、ヒトゲノムを読み出すシークエンサであり、クエリ文字列は、シークエンサから出力される例えば１５０～２００塩基程度の塩基対のデータ片である。クエリ文字列は、一旦、データベース３０に格納されてもよい。なお、クエリ文字列の文字数は、５０以上であることが好ましく、更に好ましくは１００以上とする。一方、クエリ文字列の文字数は、１０００以下であることが好ましく、望ましくは３００以下、更に望ましくは２００以下とする。本開示では、下位コンピュータ２０もまた、上位コンピュータ１０の制御の下、割り当てられたクエリ文字列に基づいて、階層的インデックスを参照して、マッピングを行う。これにより、コンピュータシステム１は、マッピング用コンピュータシステム２０２として機能する。

【００４７】

マッピングは、クエリ文字列の少なくとも一部と一致する参照文字列における部分文字列（一致文字列）を同定する処理である。つまり、マッピングにより、参照文字列におけるクエリ文字列の少なくとも一部と一致する部分文字列の出現開始位置及び長さ（文字数）が同定される。本開示に係るマッピングでは、クエリ文字列について、階層的インデックスを検索ないしは探索することにより、参照文字列におけるクエリ文字列の出現開始位置が同定される。なお、処理の高速化のため、作成された階層的インデックスは、メモリモジュール１０２又はプロセッサモジュール１０４内のキャッシュメモリ等の高速メモリに展開されることが好ましい。

【００４８】

なお、上記の例では、上位コンピュータ１０及び下位コンピュータ２０が、それぞれ、割り当てられたクエリ文字列に従ってマッピングを行っているが、これに限られず、例えば、上位コンピュータ１０のみが、割り当てられたクエリ文字列に基づいて、階層的インデックスを参照して、探索を行っても良い。

【００４９】

次に、上位コンピュータ１０及び下位コンピュータ２０は、それぞれ、後述する近似文字列照合のための文字列ペアを作成する（文字列ペア作成処理：Ｓ２０３）。文字列ペアは、マッピングにより同定される一致文字列を含む、参照文字列における被照合文字列とクエリ文字列における照合文字列とからなる。これにより、コンピュータシステム１は、文字列ペア作成用コンピュータシステム２０３として機能する。

【００５０】

具体的には、上位コンピュータ１０及び下位コンピュータ２０は、マッピングにより同

定された一致文字列の先頭及び末尾のそれぞれに、参照文字列における対応する所定長の文字列を追加することにより、被照合文字列を作成する。つまり、被照合文字列は、参照文字列において同定された一致文字列を含む該一致文字列近傍の文字列である。例えば、参照文字列が「ＣＣＧＡＴＣＴＧＴＡＴＡＣＣＣＴＡＣＧＡ」であって、一致文字列が「ＴＡＣＣ」である場合に、例えば前後２文字ずつ追加した文字列「ＴＡＴＡＣＣＣＴ」が被照合文字列となる。ヒトゲノムの解析の場合、参照文字列について、一致文字列の先頭及び末尾に追加する塩基の長さは、それぞれ、例えば５０塩基程度であり得る。

【００５１】

また、上位コンピュータ１０は、一致文字列の末尾にクエリ文字列における対応する所定長の文字列を追加することにより、照合文字列を作成する。ヒトゲノムの解析の場合、クエリ文字列について、一致文字列の末尾に追加する文字列の長さは、例えば５０塩基程度である。

【００５２】

なお、本開示では、参照文字列について、一致文字列の先頭及び末尾に所定長の文字列が追加されるものとしたが、これに限られず、例えば、先頭又は末尾の一方にのみ所定長の文字列が追加されても良い。また、クエリ文字列について、一致文字列の先頭及び末尾に、それぞれ、所定長の文字列を追加するようにしても良いし、或いは、文字列を追加せずに、一致文字列そのものを照合文字列として扱っても良い。

【００５３】

次に、上位コンピュータ１０及び下位コンピュータ２０は、参照文字列に基づく被照合文字列とクエリ文字列に基づく照合文字列とからなる文字列ペアに基づいて少なくとも１つの近似文字列を導出する（近似文字列導出処理：Ｓ２０４）。近似文字列の導出には、動的計画法を用いた所定のアラインメントが適用される。アラインメントは、２つの配列（文字列）をその要素どうしで置換、挿入及び欠損を許容しつつ比較して、定義されたスコア／ペナルティに従って変異度（類似度）を算出する手法である。アラインメントを実現するアルゴリズムとしては、Ｓｍｉｔｈ－ＷａｔｅｒｍａｎアルゴリズムやＮｅｅｄｌｅｍａｎ－ＷｕｎｓｃｈアルゴリズムがＮｅｅｄｌｅｍａｎ－Ｗｕｎｓｃｈアルゴリズムが知られている。また、動的計画法とは、ある段階で得られた最適解に基づいて次の段階の最適解を算出する手法である。つまり、変異度の算出では、動的計画法に従って、配列状のアラインメント表の各要素に対してスコアが算出され、最大スコアを持つ要素が決定され、これにより、少なくとも１つ以上の近似文字列が導出される。これにより、コンピュータシステム１は、近似文字列導出用コンピュータシステム２０４として機能する。

【００５４】

以上のように、本実施形態の近似文字列照合処理では、参照文字列に基づく階層的インデックスが作成された後、与えられたクエリ文字列に従って、該階層的インデックスを探索することにより一致文字列（及びその長さ）が同定され、同定された一致文字列に基づく被照合文字列と照合文字列とからなる文字列ペアに対して近似文字列照合がなされることにより、近似文字列が導出される。換言すると、本実施形態の近似文字列照合プログラムは、参照文字列に基づくインデックス作成用プログラムと、クエリ文字列に従って階層的インデックスを探索するマッピング用プログラムと、被照合文字列と照合文字列とからなる文字列ペアを作成するプログラムと、文字列ペアを利用して近似文字列照合がなされることにより、近似文字列を導出するアライメント用プログラムを有している。なお、文字列ペアを作成するプログラム及びアライメント用プログラムは、公知のものを採用することができるので、これ以降の詳細説明を省略する。

【００５５】

＜インデックス作成用プログラム＞

図３は、本実施形態に係るコンピュータシステムによるインデックス作成処理の一例を説明するフローチャートである。すなわち、図３は、図２に示した階層的インデックスの作成処理（Ｓ２０１）の詳細を示している。また、図１９には、インデックス作成用コンピュータシステム２０１の機能ブロック構成が示されており、参照文字列受信手段３０２、

接尾辞木生成手段３０４、部分接尾辞木生成手段３０６、部分接尾辞木分割手段３０８を有する。各機能ブロックについて、以下、フローチャート等を参照して説明する。

【００５６】

（参照文字列受信手段３０２）

まず、図３に示すように、上位コンピュータ１０は、階層的インデックスを作成するための参照文字列を受信する（Ｓ３０２）。例えば、ヒトゲノム参照配列であれば、上位コンピュータ１０は、データベース３０にアクセスし、格納されているヒトゲノム参照配列を読み出す。以下では、理解容易のため、ヒトゲノム参照配列が、４種類の文字「Ａ」、「Ｃ」、「Ｇ」、及び「Ｔ」から構成される２０文字の参照文字列「ＣＣＧＡＴＣＴＧＴＡＴＡＣＣＣＴＡＣＧＡ」であると仮定して説明する。なお、実際のヒトゲノム参照配列は、３０億文字であることに留意されたい。

【００５７】

（接尾辞木生成手段３０４）

次に上位コンピュータ１０は、受信した参照文字列について、接尾辞木を生成する（Ｓ３０４）。具体的には、まず、参照文字列の終端文字として例えば「＄」を追加する。この終端文字は、参照文字列に含まれておらず、かつ、最も小さな文字とする。この終端文字を含めて２１文字となった参照文字列を、１文字ずつ巡回させてることで得られる２１個の巡回文字列を生成する（図４参照）。なお、図４では、各巡回文字列の先頭（左端）に、巡回文字列の配列番号を付している。各配列番号は、基準となる０番の巡回文字列（参照文字列）を基準とした、各巡回文字列の出現開始位置を意味する。なお、実際のヒトゲノム参照配列が３０億文字となる場合、巡回文字列の数（配列数）は３０億個となる。

【００５８】

更に、上位コンピュータ１０は、巡回文字列における終端文字「＄」より後方の文字列（図４の網掛け領域参照）をカットしてから、これらの巡回文字列群をソートし（図５参照）、階層ツリー構造で表現される接尾辞木を生成する。この接尾辞木を図６に示す。図６の上部には、接尾辞木の階層数が示されている。なお本実施形態では、ツリー構造において、下位階層側の接点数が１となった段階で残存する文字列は、その末端接点（葉）でひとまとめにしている。結果、終端接点（葉）を含めると、本実施形態の接尾辞木の階層数は３となる。なお、実際のヒトゲノム参照配列の場合、その階層数ははるかに大きい数（例えば約２０階層）となる。なお、接尾辞木を生成する上記手順は、参考事例であって、本発明はこれに限定されない。接尾辞木を生成するプログラムは既知の様々な手法を採用できる。

【００５９】

（部分接尾辞木生成手段３０６）

次に上位コンピュータ１０は、生成した接尾辞木から部分接尾辞木を生成する（Ｓ３０６）。具体的には、図７の網掛け領域に示すように、あらかじめ設定される所定階数より下位階層側の階層データを強制的にカットする。下位階層側がかっとされた接尾辞木を、ここでは部分接尾辞木と称し、所定階層数のことをカット階層数と称する。本実施形態では、説明の便宜上、カット階層数が２であり、これよりも下位側の階層データをカットしている。部分接尾辞木を図８に示す。なお、図８の部分接尾辞木では、終端接点（葉）に複数文字が格納されている場合、先頭文字（または先頭から所定文字）を残し、それ以降の文字列をカットすることで、木構造を単純化している。接尾辞木では木の節点毎に一致している巡回文字列の位置と長さの情報が必要だが、階層をカットすることでデータ量を大きく増加させずにこれを文字自体に置き換えることができ、検索を高速化できる。

【００６０】

図８の部分接尾辞木では、巡回文字列の各配列番号を、部分接尾辞木の末端接点（葉）の右端に付している。例えば、ルート（＃）－Ｃ－Ｇ－Ａの系統の場合、本来の接尾辞木（図７参照）では、これよりも下位側に２つの接点（枝）となる「ルート（＃）－Ｃ－Ｇ－Ａ－＄」と「ルート（＃）－Ｃ－Ｇ－Ａ－Ｔ・・・」が存在しているため、部分接尾辞木のルート（＃）－Ｃ－Ｇ－Ａの葉の右側には、ソート順における上位側（兄側）接点の

配列番号１７と、下位側（弟側）接点の配列番号１が、この順番で付されることになる。また例えば、ルート（＃）－Ｔ－Ａ－Ｃの系統の場合、本来の接尾辞木（図７参照）では、これよりも下位側に２つの接点（枝）となる「ルート（＃）－Ｔ－Ａ－Ｃ－Ｃ・・・」と「ルート（＃）－Ｔ－Ａ－Ｃ－Ｇ・・・」が存在しているため、部分接尾辞木のルート（＃）－Ｔ－Ａ－Ｃの葉の右側には、ソート順における上位側（兄側）接点の配列番号１０と、下位側（弟側）接点の配列番号１５が、この順番で付されることになる。これらの場合、同定されたクエリ文字列の出現位置が、２か所に同定されることを意味する。

【００６１】

実際のヒトゲノム参照配列の場合、（全体）接尾辞木の階層数が膨大となる。この場合、部分接尾辞木を作成するために設定されるカット階層数の値は、１０００以下であることが好ましく、望ましくは５００以下、更に好ましくは３００以下とし、一層好ましくは２００以下とする。一方、カット階層数の値は、３０以上であることが好ましく、望ましくは５０以上、更に好ましくは８０以上とする。また、設定されるカット階層数の値は、クエリ文字列の文字数以上に設定されることが好ましい。例えば、クエリ文字列が１５０塩基対の場合、カット階層数は１５０以上に設定される。具体的に、ヒトゲノム参照配列のカット階層数を１５０に設定した場合、（全体）接尾辞木（データ量約２５０ＧＢ）が、１５０階層の部分接尾辞木（データ量約１００ＧＢ）に省略されるため、データ量が大幅に削減できる。

【００６２】

なお、図８の簡略化事例では、部分接尾辞木における１９個の終端接点（葉）のうち、「ルート（＃）－Ｃ－Ｇ－Ａ」系統と「ルート（＃）－Ｔ－Ａ－Ｃ」系統の２つに、複数の配列番号が付されているが、ヒトゲノム参照配列に基づいて作成される部分接尾辞木の場合、膨大な階層数が１５０階層まで削減される結果、各終端接点（葉）には、複数の配列番号が付されるが、この数は3を超えることは稀である。なお、ゲノム検索においては、実際には高確率で変異があるため、配列番号を１つまでしぼると間違った場所に一致することが多くなる。このため、マッピングの精度をあげるためには複数の配列番号が残る段階で木をカットすることが望ましい。

【００６３】

更に、上記実施形態では、接尾辞木生成手段３０４と部分接尾辞木生成手段３０６の２つの手段を経て、部分接尾辞木が生成される場合を例示しているが、部分接尾辞木の生成手順はこれに限定されない。受信した参照文字列を利用して部分接尾辞木が生成されるあらゆる工程を採用できる。

【００６４】

（部分接尾辞木分割手段３０８）

次に上位コンピュータ１０は、部分接尾辞木を途中階層（途中接点）で分割することによって、上位階層を含む上位木と、上位木の下位側に繋がる複数の下位木（以下、下位木群と称する）を作成する（Ｓ３０８）。分割が成立する途中接点の分割条件は、例えば、「終端接点から上位階層に向かって途中接点を遡っていく際に、その途中接点よりも下位階層に属する終端接点（葉）の数が、所定値以下となる場合」とする。例えば、図９（Ａ）では、「終端接点から上位階層に向かって途中接点を遡っていく際に、その途中接点よりも下位階層に属する終端接点（葉）の数が、２値以下となる場合」に分割した場合を示す。ここでは、部分接尾辞木が、ルート（＃）を含む１つの上位木と、この上位木の下位側に繋がる１４個の下位木に分割される。また、各下位木には、管理番号となるデータが付与される。この管理番号の付与ルールは任意であるが、例えば、下位木に属する配列番号中における最兄番号（ソート順における最上位番号）を採用できる。具体的に、下位木群をソート順で説明すると、「ルート（＃）－＄」の接点で分割される管理番号２０の第１下位木、「ルート（＃）－Ａ－＄」の接点で分割される管理番号１９の第２下位木、「ルート（＃）－Ａ－Ｃ」の接点で分割される管理番号１１の第３下位木、「ルート（＃）－Ａ－Ｔ」の接点で分割される管理番号９の第４下位木、「ルート（＃）－Ｃ－Ｃ－Ｃ」の接点で分割される管理番号１２の第５下位木、「ルート（＃）－Ｃ－Ｃ－Ｇ」の接点で分

割される管理番号０の第６下位木、「ルート（＃）－Ｃ－Ｃ－Ｔ」の接点で分割される管理番号１３の第７下位木、「ルート（＃）－Ｃ－Ｇ」の接点で分割される管理番号１７の第８下位木、「ルート（＃）－Ｃ－Ｔ」の接点で分割される管理番号１４の第９下位木、「ルート（＃）－Ｇ－Ａ－＄」の接点で分割される管理番号１８の第１０下位木、「ルート（＃）－Ｇ－Ａ－Ｔ」の接点で分割される管理番号２の第１１下位木、「ルート（＃）－Ｇ－Ａ－＄」の接点で分割される管理番号１８の第１０下位木、「ルート（＃）－Ｇ－Ｔ」の接点で分割される管理番号７の第１１下位木、「ルート（＃）－Ｔ－Ａ」の接点で分割される管理番号１０の第１２下位木、「ルート（＃）－Ｔ－Ｃ」の接点で分割される管理番号４の第１３下位木、「ルート（＃）－Ｔ－Ｇ」の接点で分割される管理番号６の第１４下位木が生成される。下位木群の各下位木のデータ片のデータ量は、プロセッサモジュール１０４のキャッシュメモリ（特にＬ１キャッシュメモリ）に収まるサイズに設定されることが好ましい。換言すると、分割が成立する途中接点の分割条件における「下位階層に属する終端接点（葉）の数（所定値）」は、キャッシュメモリの容量を考慮して、設定することが好ましい。

【００６５】

部分接尾辞木において、上位木、分割接点（上位木における終端接点）及び下位木管理番号を残した状態を図９（Ｂ）に示す。このように、上位木のデータ片は、下位木から切り離されることで、データ量が大幅に低減される。この上位木は、メモリモジュール（メインメモリ）に収まるデータ量とすることが好ましく、更に望ましくは、キャッシュメモリに収まるサイズとする。以上の結果、部分接尾辞木が、上位木データ片と下位木データ片群に分割された状態となる階層的インデックスデータが完成する。

【００６６】

（階層的インデックスデータのデータ構造）

図９（Ａ）に示すように、本実施形態によって生成される階層的インデックスデータのデータ構造は、参照文字列の接尾辞配列を階層化した接尾辞木構造であり、かつ、最上位階層から下位に向かう階層数を所定値に設定することで、当該所定値よりも下位側の階層データが省略されている。結果、階層的インデックスデータのデータ量を削減することが可能となっている。とりわけ、本実施形態では、最悪では参照文字列長に比例した階層数のデータ構造になるところ、当該所定値を１０００以下に設定することで、１０００階層以下の部分接尾辞木を用いているので、そのデータ量を飛躍的に削減できる。現在普及しているショットガン・シーケンス法は、シーケンサから出力されるクエリ文字列の文字数が１５０～２００程度であることから、１０００階層以下の部分接尾辞木であっても、十分にマッピングできるので、データ量削減とマッピング品質維持を合理的に両立できる。

【００６７】

また更に、本実施形態によって生成される階層的インデックスデータのデータ構造は、部分接尾辞木が途中階層で分割されており、この途中階層よりも上位階層を含む上位木のデータ片と、この途中階層よりも下位階層を含み且つ上位木の下位側に繋がる複数の下位木のデータ片を有する構造となっている。なお、「データ片に分割されている」という表現は、コンピュータが、互いに別々のデータとしてメモリモジュール１０２（記憶装置）に記憶可能な状態、あるいは、プロセッサモジュール１０４が互いに別々のデータとして処理できる状態であることを意味している。これにより、上位木データ片のデータ量や、各下位木データ片のデータ量を小さくすることができるので、これらを、メモリモジュール１０２に格納することができ、更には、キャッシュメモリに、各データ片全体を展開させたりできる。結果、いわゆる汎用コンピュータを用いる場合において、プロセッサモジュール１０４で後述するマッピング処理を並列処理する際、キャッシュメモリ内で探索処理がヒットする確率が極めて高くなり、高速なマッピングを実行できる。

【００６８】

なお、本実施形態では、部分接尾辞木における終端接点（葉）から根（ルート（＃））に向かう階層の途中に存在する分割段数が１となる場合を例示したが、本発明はこれに限定されない。例えば図１６に示すように、終端接点（葉）から根（ルート（＃））に向か

う階層の途中に、複数の分割接点（灰色接点）を設けることで、多段分割してもよい。なお、図１６では、「終端接点（葉）から根（ルート（＃））にむかって遡っていく際に通過する途中接点で、自身よりも下位階層に属する終端接点（葉）の数が３以下となる場合」に分割するようにしている。更に、この分割接点で分割された下位木については、総接点数を仮想的に１に変換してから、更に、分割接点から上位階層に向かって接点を遡り、同様に、通過する途中接点で、自身よりも下位階層に属する終端接点（葉）の数が、３以下となる場合」に追加分割している。

【００６９】

なお、ヒトゲノム参照配列に基づいて、１５０階層まで削減された部分接尾辞木について、「終端接点から上位階層に向かって途中接点を遡っていく際に、その途中接点よりも下位階層に属する終端接点（葉）の数が、所定値以下となる場合」という分割条件について、上記所定値は、３万～３０万の範囲内に設定されることが好ましい。望ましくは所定値を５万～２０万に設定する。例えば所定値を３万に設定すると、根（ルート（＃））を含む１つの最上位木と、この最上位木に属する約１００万個の下位木群に分割できる。例えば所定値を１０万に設定すると、根（ルート（＃））を含む１つの最上位木と、この最上位木に属する約３０万個の下位木群に分割できる。例えば所定値を３万に設定すると、根（ルート（＃））を含む１つの最上位木と、この最上位木に属する約１０万個の下位木群に分割できる。これらの条件の場合、最上位木のデータサイズが１０ＭＢ以下（例えば１ＭＢ程度）となり、メモリモジュール１０２（メインメモリ）に十分に展開できるサイズとなる。更にこれらの条件の場合、各下位木のデータ量が１０ＭＢ以下、望ましくは１ＭＢ以下となり、メモリモジュール１０２（メインメモリ）に十分に展開できるサイズであり、更に、キャッシュメモリにも収まるサイズとなる。

【００７０】

＜マッピング用プログラム＞

図１０は、本実施形態に係るコンピュータシステムによるマッピング処理の一例を説明するフローチャートである。すなわち、図１０は、図２に示したクエリ文字列の参照文字列へのマッピング処理（Ｓ２０２）の詳細を示している。かかるマッピング処理により、クエリ文字列が参照文字列にマッピングされ、参照文字列における出現開始位置が同定される。また、図１９には、マッピング用コンピュータシステム２０２の機能ブロック構成が示されており、上位木展開手段４０２、クエリ文字列群受信手段４０４、クエリ文字列群展開手段４０５、クエリ文字列群判別手段４０６、第一繰り返し手段４０７、下位木及びクエリ文字列群セット展開手段４０８、メモリ残量判定手段４０９、クエリ文字列出現位置決定手段４１０、第二繰り返し手段４１２を有する。各機能ブロックについて、以下、フローチャート等を参照して説明する。なお、以下では、上位コンピュータ１０によるマッピング処理が説明されるが、並列的に動作する下位コンピュータ２０によるマッピング処理も同様である。

【００７１】

（上位木展開手段４０２）

図１０に示すように、上位コンピュータ１０は、データベース３０から、階層的インデックスにおける上位木を読み出して、メモリモジュール１０２上に展開し、記憶する（Ｓ４０２）。上位コンピュータ１０は、高速化の観点から、上位木を構成するデータをメモリモジュール１０２上に連続的に展開し、記憶する。ここで、連続的に展開とは、データが同一バンクにおける連続的なメモリアドレスに配置されることを含む。

【００７２】

（クエリ文字列群受信手段４０４）

次に、上位コンピュータ１０は、参照文字列に対してマッピングを行うための複数のクエリ文字列（クエリ文字列群）を受信する（Ｓ４０４）。受信したクエリ文字列は、例えば、データベース３０等の記憶手段に一時的に保存される。例えば、参照文字列がヒトゲノム参照配列であれば、クエリ文字列は、シークエンサから読み出される塩基対のデータ片である。以下では、理解容易のため、図１１に示すように、クエリ文字列は３文字で、そ

の数が４０個になるものとして説明する。しかし、実際のシークエンサから読み出される塩基対のデータ片のクエリ文字列は約１５０文字であり、その数は約７億個になることに留意されたい。なお、ここでは一時的にクエリ文字列群をデータベース３０に保存する場合を例示するが、外部装置となるシークエンサ自体がデータベース機能を有する場合は、シークエンサ側のデータベースをそのまま活用してもよい。

【００７３】

（クエリ文字列群展開手段４０５）

次に、上位コンピュータ１０は、データベース３０に保存されるクエリ文字列群を読み出して、メモリモジュール１０２上に展開し、記憶する（Ｓ４０５）。上位コンピュータ１０は、高速化の観点から、クエリ文字列群を構成するデータをメモリモジュール１０２上に連続的に展開し、記憶する。ただし、クエリ文字列群の全データ量は大きく、また、上位木展開手段４０２によるメモリモジュール１０２への上位木の展開を優先する必要があるため、上位コンピュータ１０のメモリモジュール１０２には、一部のクエリ文字列群を展開する。

【００７４】

（クエリ文字列群判別手段４０６）

次に、上位コンピュータ１０は、メモリモジュール１０２に展開したクエリ文字列群の中からクエリ文字列を一つずつ取り出して、同じメモリモジュール１０２に展開されている上位木（図９（Ｂ））を参照することで、どの下位木（下位木管理番号）に属するか否かを判別する（Ｓ４０６）。これにより、クエリ文字列の各々に紐づいた下位木管理番号が決定され、そのデータがメモリモジュール１０２に保存される。結果、図１１に示すように、４０個の各クエリ文字列について、所属する下位木管理番号が付された状態で、メモリモジュール１０２に記憶される。クエリ文字列群の各々に紐づいた下位木管理番号データは、最後に、データベース３０に保存される。なお、図１８に示すように、プロセッサモジュールが４個のコアＡ，Ｂ，Ｃ，Ｄを有する場合、上記判別処理を並列処理することで、高速化することが好ましい。

【００７５】

（第一繰り返し手段４０７）

クエリ文字列群展開手段４０５で述べたように、上位コンピュータ１０のメモリモジュール１０２には、その容量上限によって、一部のクエリ文字列群しか展開できない。従って、上位コンピュータ１０は、データベース３０を参照して、ステップＳ４０６の未判別のクエリ文字列群が残存するか否かを判定する（Ｓ４０７）。データベース３０に、未判別のクエリ文字列群が存在する場合は、ステップＳ４０５に戻り、残りのクエリ文字列群をメモリモジュール１０２上に展開して記憶し、各クエリ文字列について、これに紐づいた下位木管理番号を判別する（Ｓ４０６）。

【００７６】

一方、ステップＳ４０７で未判別のクエリ文字列群が無くなったら、全てのクエリ文字列が、下位木単位でグループ分けされたことになり、ステップＳ４０８に進む。なお、図１１に示すクエリ文字列で、網掛けされた文字列は、上位木を参照した判別時に使用されなかった文字列を意味する。

【００７７】

（下位木及びクエリ文字列群セット展開手段４０８）

次いで上位コンピュータ１０は、データベース３０から、階層的インデックスにおける特定の下位木を読み出して、メインメモリモジュール１０２上に展開する。更に上位コンピュータ１０は、展開された特定の下位木と同じ下位木管理番号に所属するクエリ文字列群についても、同じメモリモジュール１０２上に展開して記憶する。これにより、特定の下位木とこれに属するクエリ文字列群が、メモリモジュール１０２にセット展開されることになる（Ｓ４０８）。なお、クエリ文字列群のメモリモジュール１０２への展開方法は様々である。例えば、クエリ文字列群のクエリ文字列同士が、プロセッサモジュール１０４によって連続的に読みだされるように、前のクエリ文字列と後のクエリ文字列をポインタ

で繋ぐことで、クエリ文字列群を配列化してもよい。

【００７８】

（メモリ残量判定手段４０９）

更に上位コンピュータ１０は、メモリモジュール１０２の残量を参照して、まだ、他の特定の下位木及び当該下位木と同じ下位木管理番号に所属するクエリ文字列群を展開できる空き容量がある場合は、ステップＳ４０８に戻って、他の特定の下位木及びクエリ文字列群をセット展開する。これにより、メモリモジュール１０２には、下位木及びクエリ文字列群の複数セットが、同時に展開されることになる。なお、本実施形態では、下位木管理番号１１，９，１２，０の下位木及びクエリ文字列群の４セットが、メモリモジュール１０２に展開される。

【００７９】

（クエリ文字列出現位置決定手段４１０）

次に、上位コンピュータ１０は、セット展開されたクエリ文字列群の各クエリ文字列を順番に取り出して、所属する下位木を探索し、各クエリ文字列が最終的にどの配列番号（出現位置）に属するか否かを同定する（Ｓ４１０）。

【００８０】

より詳細に、例えば、プロセッサモジュールが４個のコアＡ，Ｂ，Ｃ，Ｄを有する場合、下位木管理番号１１，９，１２，０の下位木及びクエリ文字列群の４セットの同定処理を、各コアＡ，Ｂ，Ｃ，Ｄに割り振ることで、並列処理を行う。各下位木のデータ量は、キャッシュメモリの容量以下に予め設定されているため、図１２（Ａ）～（Ｄ）に示すように、同定処理において、各下位木データが４個のＬ１キャッシュメモリＬ１ａ，Ｌ１ｂ，Ｌ１ｃ，Ｌ１ｄ上に展開されることになる。結果、各コアＡ，Ｂ，Ｃ，Ｄにおける各下位木における探索処理が、Ｌ１キャッシュメモリＬ１ａ，Ｌ１ｂ，Ｌ１ｃ，Ｌ１ｄ内で完結するので、高速処理が実現される。。なお、図１２（Ａ）～（Ｄ）では、説明の便宜上、各下位木の上部の上部には階層数が示されており、各下位木の左側には、各下位木の管理番号が付されている。また、各下位木の左側には、下位木の終端接点（葉）毎の配列番号（出現位置）が示されている。コアＡ，Ｂ，Ｃ，Ｄで同定された各クエリ文字列の右側には、最終的な配列番号（出現位置）が示されている。なお、クエリ文字列で網掛けされた文字は、下位木を参照した出現位置決定時に使用されなかった文字（上位木の判別で使用された文字）を意味する。このステップＳ４１０では、メモリモジュール１０２にセット展開された全ての下位木及びクエリ文字列群について出現位置が同定される。その同定結果に関するデータは、データベース３０に保存される。

【００８１】

（第二繰り返し手段４１２）

次に、上位コンピュータ１０は、データベース３０において、未処理の下位木及びクエリ文字列群のセットが残存するか否かを判定する（Ｓ４１２）。図１２では、１６個の内の４セットの下位木及びクエリ文字列を処理した状態となるので、ステップＳ４０８に戻り、図１３（Ａ）～（Ｄ）に示すように、下位木管理番号１３，１７，１４，１８の４セットの下位木及びクエリ文字列を、メモリモジュール１０２にセット展開して記憶する。その後、４個のコアＡ，Ｂ，Ｃ，Ｄで、セット単位で、各クエリ文字列について所属する下位木を参照することで、各クエリ文字列が最終的にどの配列番号（出現位置）に属するか否かを同定する（Ｓ４１０）。更に、未処理の下位木及びクエリ文字列群のセットが残存するか否かを判定する（Ｓ４１２）。まだ、４セットの未処理の下位木及びクエリ文字列が存在するので、ステップＳ４０８に戻り、図１４（Ａ）～（Ｄ）に示すように、下位木管理番号７，１０，４，６の４セットの下位木及びクエリ文字列を、メモリモジュール１０２にセット展開して記憶し、４個のコアＡ，Ｂ，Ｃ，Ｄで、セット単位で、各クエリ文字列について所属する下位木を参照することで、各クエリ文字列が最終的にどの配列番号（出現位置）に属するか否かを同定する（Ｓ４１０）。次に、未処理の下位木及びクエリ文字列のセットが残存するか否かを判定する（Ｓ４１２）。未処理のセットが無くなったので、マッピング処理を終了する。

【００８２】
なお、このＳ４０８～Ｓ４１２の処理は、上位コンピュータ１０のプロセッサモジュール１０４における４個のコアＡ，Ｂ，Ｃ，Ｄで並行処理されるので、処理時間を大幅に短縮できる。特に図示しないが、複数の下位コンピュータ２０（１）～２０（ｎ））でも、このマッピング処理を同時に処理（分散処理）させることで、さらなる時間短縮も実現できる。
【００８３】
なお、図１０のマッピング処理では、ステップＳ４０７で未処理のクエリ文字列群が残存する場合に、一旦、処理済みのクエリ文字列群に関する情報をデータベース３０に書き込んでメモリモジュール１０２を解放してから、未処理のクエリ文字列群をデータベース３０からメモリモジュール１０２に読み込んで、各クエリ文字列に紐づいた下位木管理番号を判別している（Ｓ４０６）。そして、ステップＳ４０７を経て、データベース３０における全てのクエリ文字列群の判別処理が完了してから、ステップＳ４０８～ステップＳ４１２に進んで同定処理を行う場合を例示したが、本発明はこれに限定されない。例えば図２０に示すように、ステップＳ４０５でメモリモジュール１０２に展開したクエリ文字列群を対象に、ステップＳ４０６で判別処理が完了したら、そのままステップＳ４０８に進み、判別処理で判定された下位木管理番号に対応する全ての下位木を取り込んでセット展開状態とし、ステップ４１０及びステップ４１２で同定処理を行うことができる。その後、ステップＳ４０７で未処理のクエリ文字列の有無を判別して、もし、未処理のクエリ文字列がある場合は、ステップＳ４０５に戻るようにしてもよい。
【００８４】
以上のように、本実施形態のマッピング用コンピュータプログラムは、上位コンピュータ１０が、データ量が削減された上位木のデータ片を利用して、予め、複数のクエリ文字列が、どの下位木に属するかについて判別するので、高速処理を実現できる。更に、判別完了後は、下位木及びこれに属するクエリ文字列群をセット展開しておくようにし、プロセッサモジュール１０４のコアＡ，Ｂ，Ｃ，Ｄが、当該特定の下位木に属する複数のクエリ文字列を連続的に探索することで、クエリ文字列の出現位置を連続的に同定できる。同定処理では、各下位木がキャッシュメモリに取り込まれるので、メモリモジュール１０２に対するランダムアクセスの回数を大幅に減らすことができる。
【００８５】
なお、上記マッピング処理では、クエリ文字列の全体と一致する参照文字列を同定する場合を例示したが、本発明はこれに限定されない。例えば、クエリ文字列から一部文字列を抽出し、この一部文字列に対して、マッピング処理を実行してもよい。
【００８６】
＜階層的インデックスのデータ構造の変形例＞
上記実施形態における階層的インデックスで採用される接尾辞木は、ＡＣＧＴを有する参照文字列を１文字ずつ巡回させて得られる複数個の巡回文字列（図５参照）について、図６に示すように、１文字単位で階層化（ツリー化）して接尾辞木を生成する場合を例示したが、本発明はこれに限定されない。巡回文字列（図５参照）について、複数文字を１セットにして階層化した接尾辞木を生成してもよい。図１５には、巡回文字列（図５参照）について２文字単位で文字列ペアを形成し、この文字列ペアで接尾辞木を生成する場合を示す。このようにすると、階層数が削減されるので、部分接尾辞、上位木、下位木等のデータ量を更に削減することが可能となる。一方、例えば５文字以上を１セットにして階層化すると、接点数が増大してしまい、かえってデータ量が増大する可能性もある。従って、複数文字列をセットにして接尾辞木を生成する場合、そのセット文字数は２以上且つ４以下程度であることが好ましい。
【００８７】
上記各実施形態は、本発明を説明するための例示であり、本発明をこれらの実施形態にのみ限定する趣旨ではない。本発明は、その要旨を逸脱しない限り、さまざまな形態で実施することができる。例えば、本実施形態では、遺伝子情報となる参照文字列について、

階層的インデックスを作成したり、マッピングを行ったりする場合を例示したが、本発明はこれに限定されず、遺伝子情報以外の参照文字列について、階層的インデックスを作成したり、マッピングを行ったりする場合に適用することができる。

【００８８】

また例えば、本明細書に開示される方法においては、その結果に矛盾が生じない限り、ステップ、動作又は機能を並行して又は異なる順に実施しても良い。説明されたステップ、動作及び機能は、単なる例として提供されており、ステップ、動作及び機能のうちのいくつかは、発明の要旨を逸脱しない範囲で、省略でき、また、互いに結合させることで一つのものとしてもよく、また、他のステップ、動作又は機能を追加してもよい。

【００８９】

更に上記実施形態では、コンピューティングデバイスのハードウェア構成が、汎用的な構造である場合を例示したが、本発明はこれに限定されない。例えば、ＦＰＧＡ（Field Programmable Gate Array）等によって個別にカスタマイズされたハードウェアを採用してもよい。具体的には、図１９に示される機能ブロックの全部または一部を、ＦＰＧＡによって実現することができる。

【００９０】

また更に、上記実施形態で示した階層的インデックスのデータ構造を実装するにあたっての詳細な表現手法（表記手法）は特に限定されない。例えば、ＬＯＵＤＳ（Level-Ordered Unary Degree Sequence）手法によって木構造を表現すると、メモリ量を更に削減することも可能となる。

【００９１】

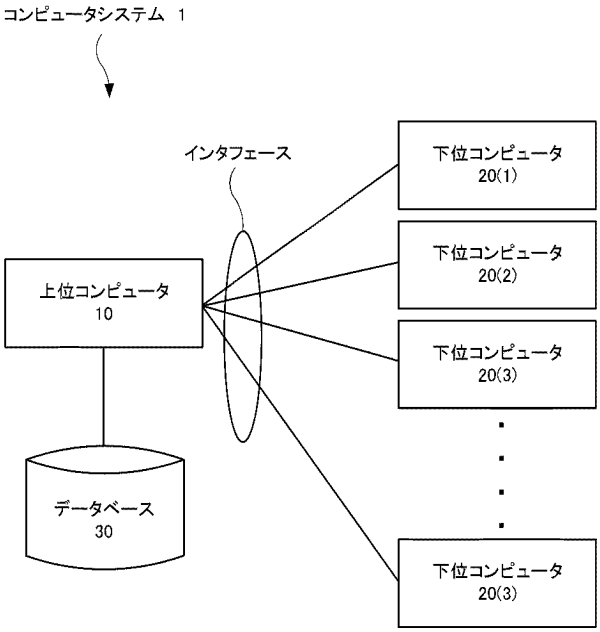
また、本明細書では、さまざまな実施形態が開示されているが、一の実施形態における特定のフィーチャ（技術的事項）を、適宜改良しながら、他の実施形態に追加し、又は該他の実施形態における特定のフィーチャと置換することができ、そのような形態も本発明の要旨に含まれる。

【符号の説明】

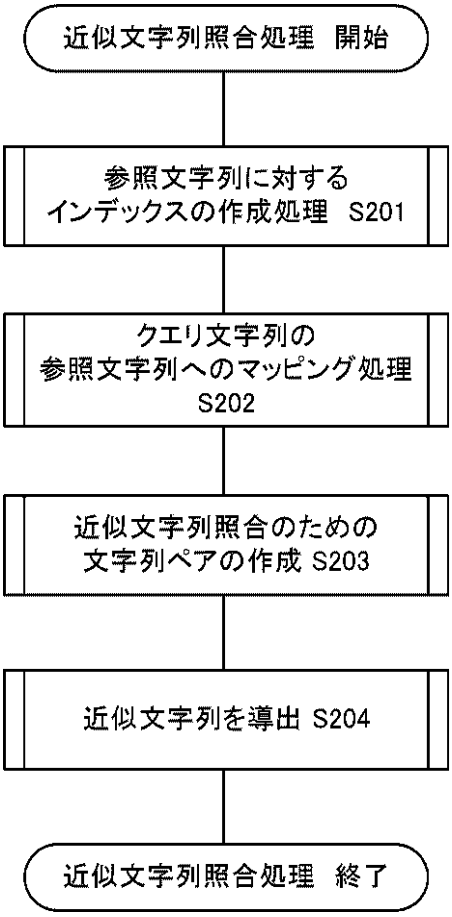
【００９２】

１　　コンピュータシステム
１０　上位コンピュータ
２０　下位コンピュータ
３０　データベース

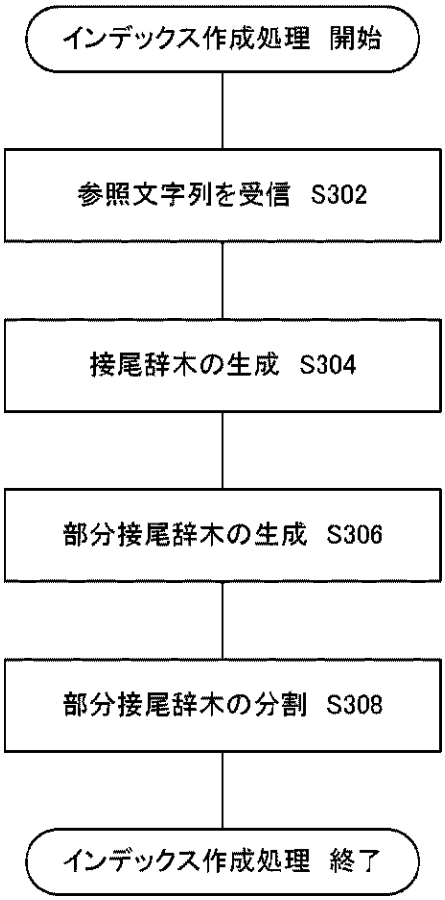
【図１】



【図２】



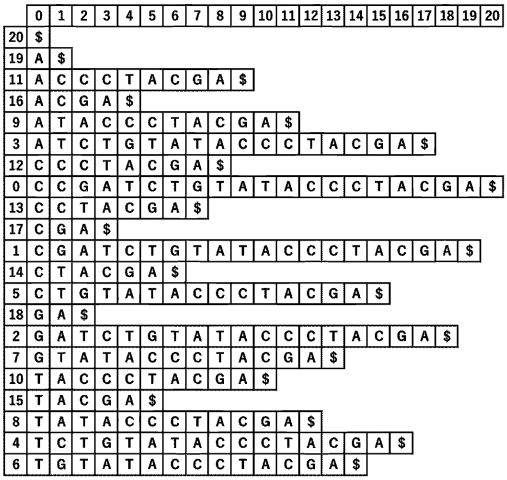
【図３】



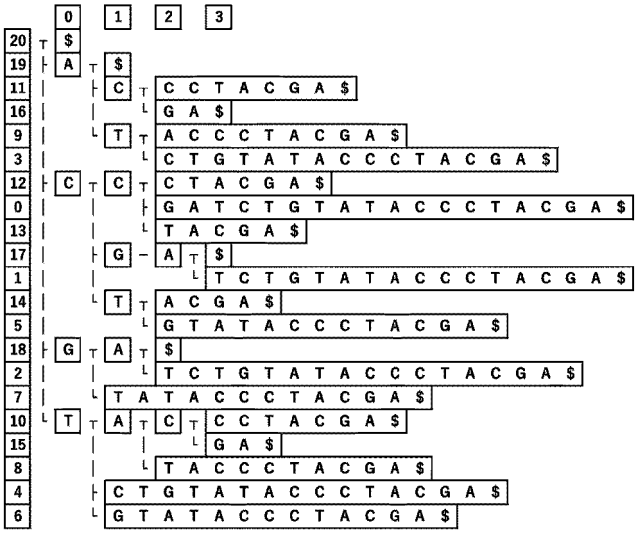
【図４】

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
0	C	C	G	A	T	C	T	G	T	A	T	A	C	C	C	T	A	C	G	A	$
1	C	G	A	T	C	T	G	T	A	T	A	C	C	C	T	A	C	G	A	$	C
2	G	A	T	C	T	G	T	A	T	A	C	C	C	T	A	C	G	A	$	C	C
3	A	T	C	T	G	T	A	T	A	C	C	C	T	A	C	G	A	$	C	C	G
4	T	C	T	G	T	A	T	A	C	C	C	T	A	C	G	A	$	C	C	G	A
5	C	T	G	T	A	T	A	C	C	C	T	A	C	G	A	$	C	C	G	A	T
6	T	G	T	A	T	A	C	C	C	T	A	C	G	A	$	C	C	G	A	T	C
7	G	T	A	T	A	C	C	C	T	A	C	G	A	$	C	C	G	A	T	C	T
8	T	A	T	A	C	C	C	T	A	C	G	A	$	C	C	G	A	T	C	T	G
9	A	T	A	C	C	C	T	A	C	G	A	$	C	C	G	A	T	C	T	G	T
10	T	A	C	C	C	T	A	C	G	A	$	C	C	G	A	T	C	T	G	T	A
11	A	C	C	C	T	A	C	G	A	$	C	C	G	A	T	C	T	G	T	A	T
12	C	C	C	T	A	C	G	A	$	C	C	G	A	T	C	T	G	T	A	T	A
13	C	C	T	A	C	G	A	$	C	C	G	A	T	C	T	G	T	A	T	A	C
14	C	T	A	C	G	A	$	C	C	G	A	T	C	T	G	T	A	T	A	C	C
15	T	A	C	G	A	$	C	C	G	A	T	C	T	G	T	A	T	A	C	C	C
16	A	C	G	A	$	C	C	G	A	T	C	T	G	T	A	T	A	C	C	C	T
17	C	G	A	$	C	C	G	A	T	C	T	G	T	A	T	A	C	C	C	T	A
18	G	A	$	C	C	G	A	T	C	T	G	T	A	T	A	C	C	C	T	A	C
19	A	$	C	C	G	A	T	C	T	G	T	A	T	A	C	C	C	T	A	C	G
20	$	C	C	G	A	T	C	T	G	T	A	T	A	C	C	C	T	A	C	G	A

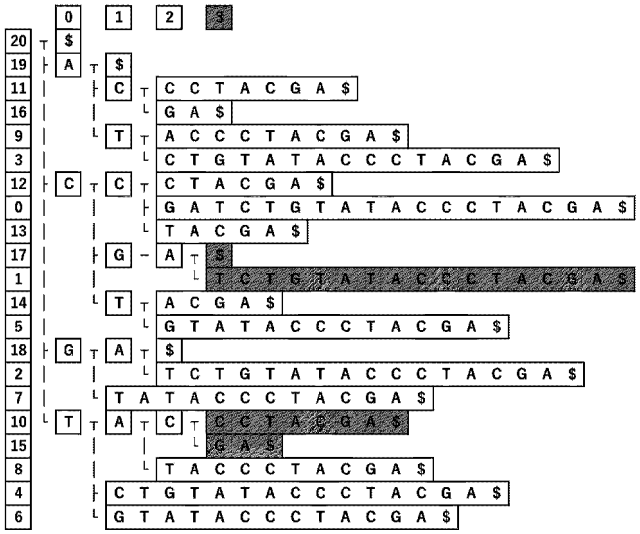
【図５】


0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
20 $
19 A $
11 A C C C T A C G A $
16 A C G A $
9 A T A C C C T A C G A $
3 A T C T G T A T A C C C T A C G A $
12 C C C T A C G A $
0 C C G A T C T G T A T A C C C T A C G A $
13 C C T A C G A $
17 C G A $
1 C G A T C T G T A T A C C C T A C G A $
14 C T A C G A $
5 C T G T A T A C C C T A C G A $
18 G A $
2 G A T C T G T A T A C C C T A C G A $
7 G T A T A C C C T A C G A $
10 T A C C C T A C G A $
15 T A C G A $
8 T A T A C C C T A C G A $
4 T C T G T A T A C C C T A C G A $
6 T G T A T A C C C T A C G A $

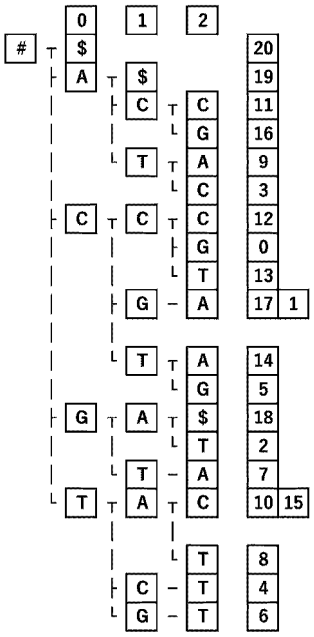
【図６】


0 1 2 3
20 $
19 A $
11 C C C T A C G A $
16 G A $
9 T A C C C T A C G A $
3 C T G T A T A C C C T A C G A $
12 C C C T A C G A $
0 G A T C T G T A T A C C C T A C G A $
13 T A C G A $
17 G A $
1 T C T G T A T A C C C T A C G A $
14 T A C G A $
5 G T A T A C C C T A C G A $
18 G A $
2 T C T G T A T A C C C T A C G A $
7 T A T A C C C T A C G A $
10 T A C C C T A C G A $
15 G A $
8 T A C C C T A C G A $
4 C T G T A T A C C C T A C G A $
6 G T A T A C C C T A C G A $

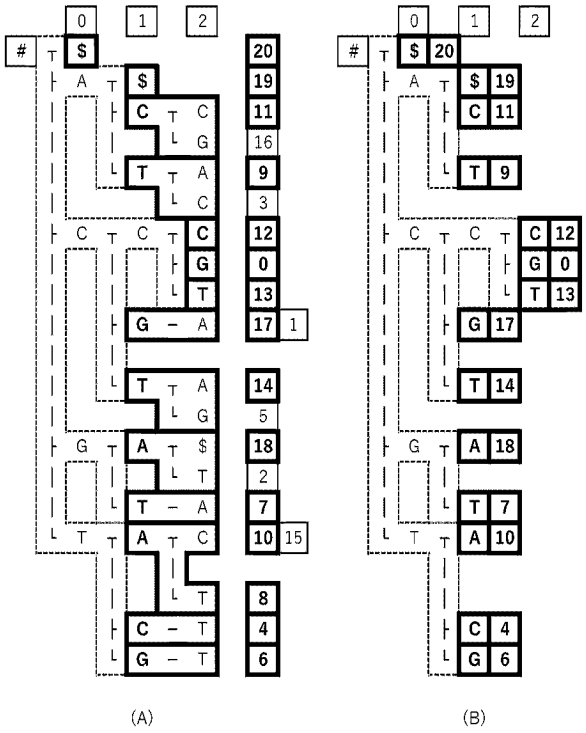
【図７】


0 1 2 3
20 $
19 A $
11 C C C T A C G A $
16 G A $
9 T A C C C T A C G A $
3 C T G T A T A C C C T A C G A $
12 C C C T A C G A $
0 G A T C T G T A T A C C C T A C G A $
13 T A C G A $
17 G A $
1 T C T G T A T A C C C T A C G A $
14 T A C G A $
5 G T A T A C C C T A C G A $
18 G A $
2 T C T G T A T A C C C T A C G A $
7 T A T A C C C T A C G A $
10 T A C C C T A C G A $
15 G A $
8 T A C C C T A C G A $
4 C T G T A T A C C C T A C G A $
6 G T A T A C C C T A C G A $

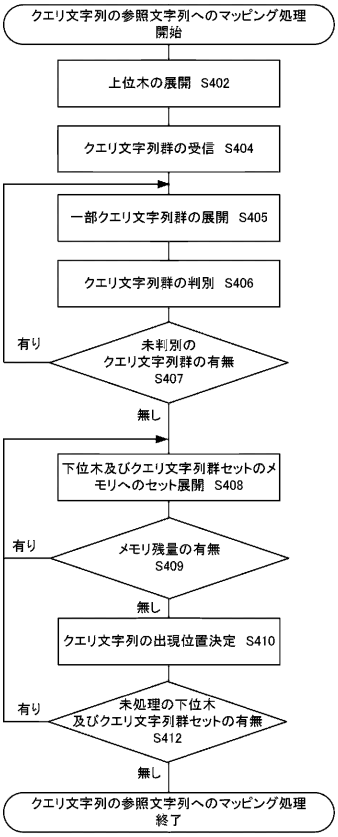
【図８】


0 1 2
$ 20
A $ 19
C C 11
G 16
T A 9
C 3
C C C 12
G 0
T 13
G A 17 1
T A 14
G 5
G A $ 18
T 2
T A 7
T A C 10 15
T 8
C T 4
G T 6

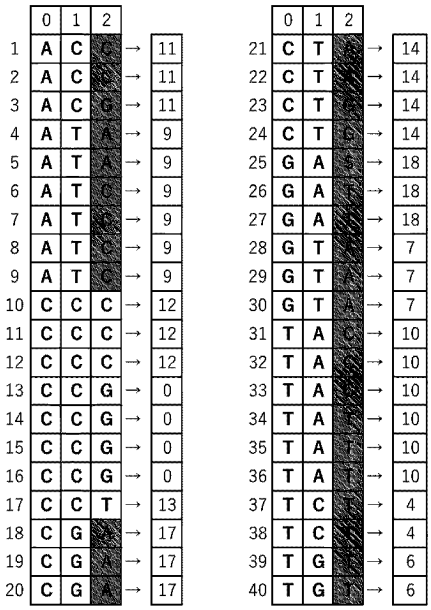
【図９】



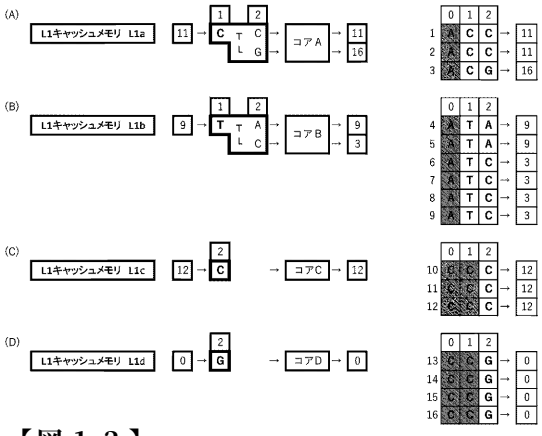
【図１０】



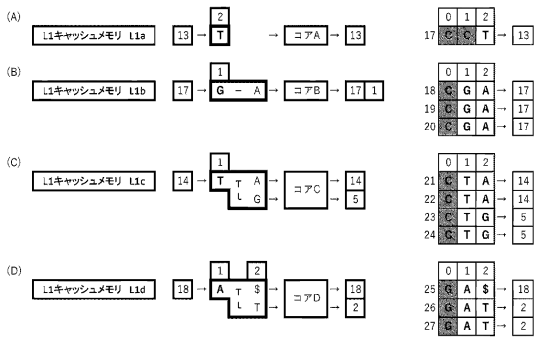
【図１１】



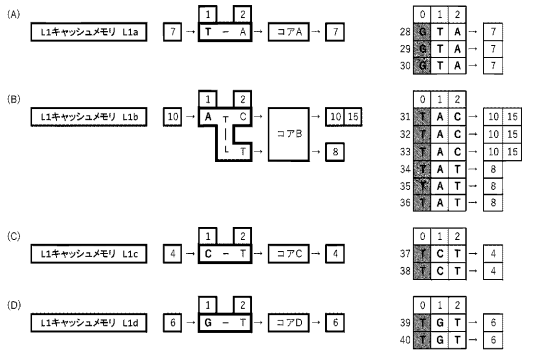
【図１２】



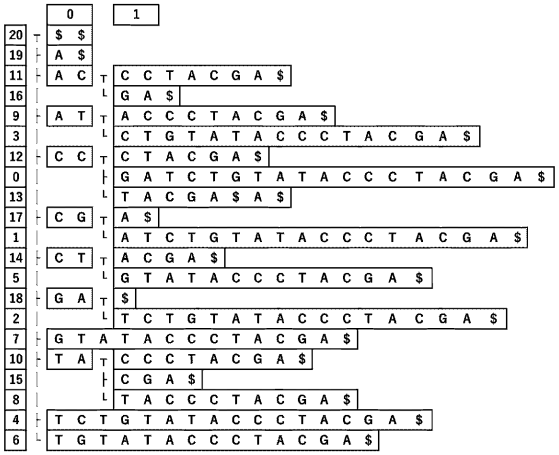
【図１３】



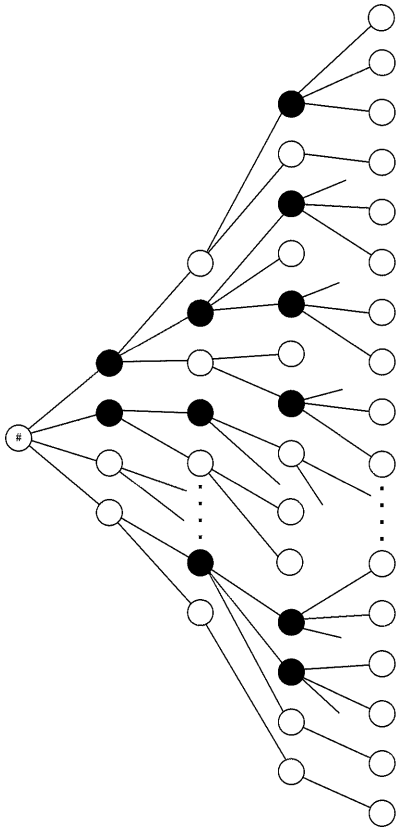
【図１４】



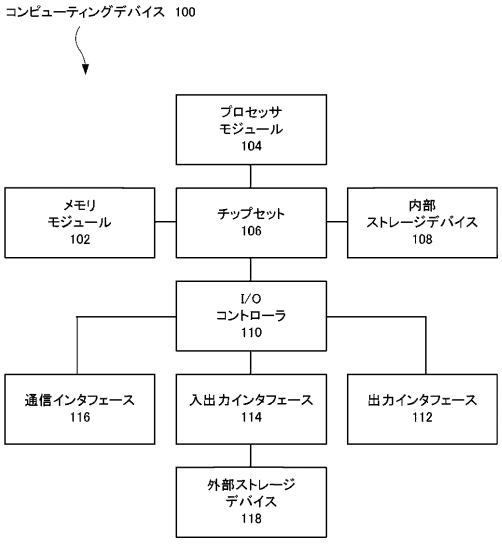
【図１５】



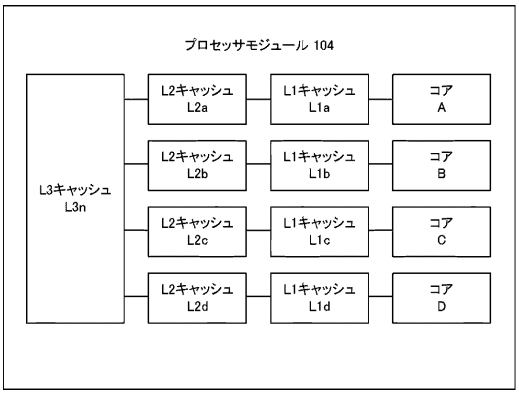
【図１６】



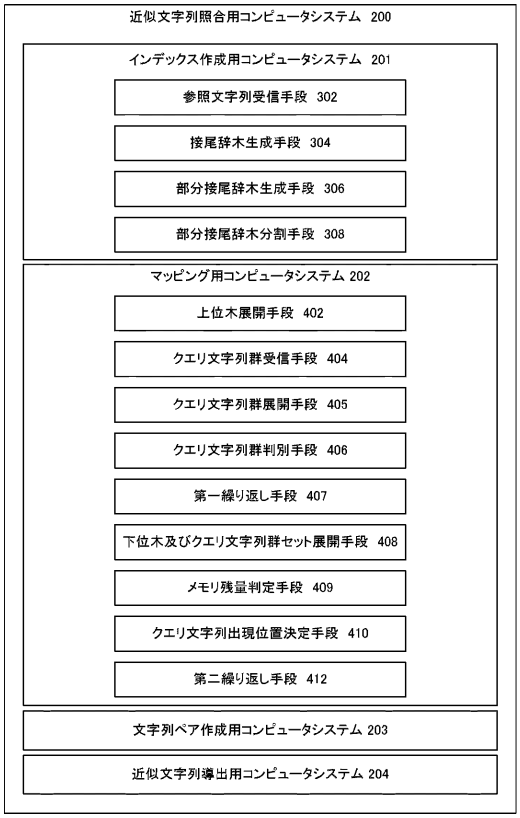
【図１７】



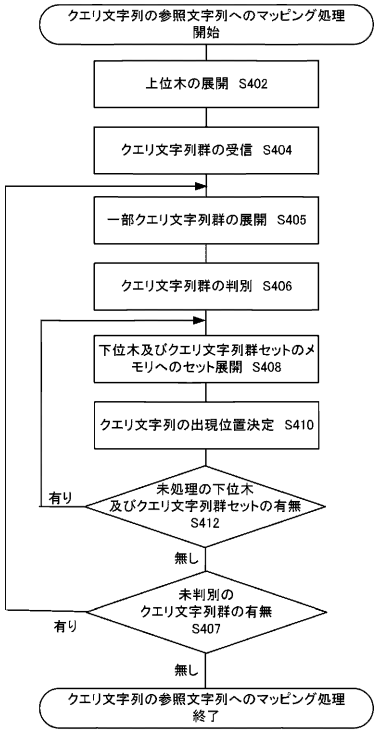
【図１８】



【図１９】


近似文字列照合用コンピュータシステム 200
インデックス作成用コンピュータシステム 201
参照文字列受信手段 302
接尾辞木生成手段 304
部分接尾辞木生成手段 306
部分接尾辞木分割手段 308
マッピング用コンピュータシステム 202
上位木展開手段 402
クエリ文字列群受信手段 404
クエリ文字列群展開手段 405
クエリ文字列群判別手段 406
第一繰り返し手段 407
下位木及びクエリ文字列群セット展開手段 408
メモリ残量判定手段 409
クエリ文字列出現位置決定手段 410
第二繰り返し手段 412
文字列ペア作成用コンピュータシステム 203
近似文字列導出用コンピュータシステム 204

【図２０】


クエリ文字列の参照文字列へのマッピング処理
開始
上位木の展開 S402
クエリ文字列群の受信 S404
一部クエリ文字列群の展開 S405
クエリ文字列群の判別 S406
下位木及びクエリ文字列群セットのメモリへのセット展開 S408
クエリ文字列の出現位置決定 S410
未処理の下位木及びクエリ文字列群セットの有無 S412
有り
無し
未判別のクエリ文字列群の有無 S407
有り
無し
クエリ文字列の参照文字列へのマッピング処理
終了

フロントページの続き

(72)発明者　牧野　淳一郎
神奈川県横浜市金沢区泥亀一丁目２８番Ｂ－１３１２　先端加速システムズ株式会社内

Ｆターム(参考) 5B175 DA01　FA01　KA06

(54)【発明の名称】インデックス作成用コンピュータプログラム、マッピング用コンピュータプログラム、インデックス作成用コンピュータ装置、マッピング用コンピュータ装置、階層的インデックスのデータ構造。