(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2023-80989
(P2023-80989A)
(43)公開日　令和5年6月9日(2023.6.9)

(51)Int.Cl.		F I		テーマコード（参考）
G06F 16/31	*(2019.01)*	G06F 16/31		5B175

審査請求 有　請求項の数 21　ＯＬ　（全 35 頁）

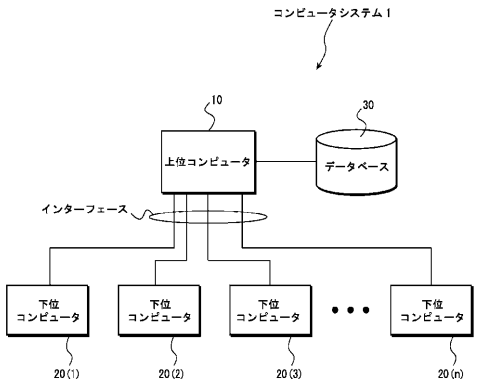
(21)出願番号　特願2021-194605(P2021-194605)
(22)出願日　令和3年11月30日(2021.11.30)

(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
(74)代理人　100103850
弁理士　田中　秀▲てつ▼
(74)代理人　100105854
弁理士　廣瀬　一
(74)代理人　100116012
弁理士　宮坂　徹
(74)代理人　100066980
弁理士　森　哲也

最終頁に続く

(54)【発明の名称】近似文字列照合方法及び該方法を実現するためのコンピュータプログラム

(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近似文字列を検索する方法である。前記方法は、前記参照文字列に基づいて階層的インデックスを作成することと、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クエリ文字列のマッピングを行うことと、前記マッピングにより同定される少なくとも１以上の前記部分文字列に基づいて、前記近似文字列を導出することを含む。前記階層的インデックスは、前記参照文字列から切り出される各キーに対して、その出現回数に従い追加キーを追加しソートすることを繰り返しながら作成される。
【選択図】　図１



【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近似文字列を検索するための方法を実現させるためのコンピュータプログラムであって、
前記方法は、
前記参照文字列に基づいて階層的インデックスを作成することと、
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、
前記のマッピングにより同定される少なくとも１以上の前記部分文字列に基づいて、前記近似文字列を導出することと、を含み、
前記階層的インデックスを作成することは、
前記参照文字列から所定長の各第１のキーを切り出すことと、
切り出された前記各第１のキーについて、所定のハッシュ関数により該第１のキーに基づいて算出されるハッシュ値を割り当てた第１のキー配列を作成することと、
作成された前記第１のキー配列を更新することと、
更新された前記第１のキー配列を前記階層的インデックスとして出力することと、を含み、
前記第１のキー配列を更新することは、
前記第１のキー配列における前記各第１のキーについて、前記参照文字列における該第１のキーの出現回数を同定することと、
同定された前記第１のキーの前記出現回数に従って、該第１のキーに第１の追加キーを追加することにより新たな第１のキーを作成し、該新たな第１のキーに基づいて前記第１のキー配列を更新することと、を含む、
コンピュータプログラム。
【請求項２】
前記第１のキー配列を作成することは、前記ハッシュ値に従って前記第１のキー配列における前記各第１のキーをソートすることを含む、
請求項１に記載のコンピュータプログラム。
【請求項３】
前記第１のキー配列を更新することは、
前記同定した出現回数が所定の許容値を超えているか否かを判断することと、
前記同定された前記出現回数が所定の許容値を超えていると判断される場合に、前記第１のキーに対して前記参照文字列における該第１のキーに続く少なくとも１以上の文字からなる前記第１の追加キーを追加することにより前記新たな第１のキーを作成することと、
前記新たな第１のキーについて、前記参照文字列における該第１のキーの出現回数を同定することと、を含む、
請求項１又は２に記載のコンピュータプログラム。
【請求項４】
前記第１のキー配列を更新することは、前記第１の追加キーに従って前記第１のキー配列における前記新たな第１のキーをソートすることを更に含む、
請求項１から３のいずれか一項に記載のコンピュータプログラム。
【請求項５】
前記第１のキー配列を更新することは、前記同定された前記出現回数が所定の許容値を超えていないと判断されるまで、現在の前記第１のキーに新たな前記第１の追加キーを順次に追加することにより新たな前記第１のキーを作成することを含む、
請求項３又は４に記載のコンピュータプログラム。
【請求項６】
前記キー配列を前記階層的インデックスとして出力することは、
前記同定された前記出現回数が所定の許容値を超えていないと判断される場合に、現在

の前記キー配列を前記階層的インデックスとして出力することを含む、
請求項３から５のいずれか一項に記載のコンピュータプログラム。
【請求項７】
前記マッピングを行うことは、
前記クエリ文字列から所定長の各第２のキーを切り出すことと、
前記クエリ文字列から切り出された前記各第２のキーについて、前記所定のハッシュ関数により該第２のキーに基づいて算出されるハッシュ値を割り当てた第２のキー配列を作成することと、
前記各第２のキーについて、前記ハッシュ値に従って、所定のサンプリング間隔で、前記階層的インデックスを参照し、該第２のキーの出現開始位置及び出現回数を同定することと、を含む、
請求項１から６のいずれか一項に記載のコンピュータプログラム。
【請求項８】
前記第２のキーの前記出現開始位置及び前記出現回数を同定することは、
前記第２のキーの前記出現回数が前記所定の許容値を超えているか否かを判断することと、
前記第２のキーの前記出現回数が前記所定の許容値を超えていると判断される場合に、前記第２のキーに対して前記クエリ文字列における該第２のキーに続く少なくとも１以上の文字からなる第２の追加キーを追加することにより新たな第２のキーを作成することと、
前記第２のキーの前記出現回数が前記所定の許容値を超えていないと判断される場合に、同定された現在の前記第２のキーを一致文字列として出力するとともに該第２のキーの前記出現開始位置を出力することと、を含む、
請求項７に記載のコンピュータプログラム。
【請求項９】
前記第２のキーの前記出現開始位置及び前記出現回数を同定することは、前記第２のキーの前記同定された前記出現回数が前記所定の許容値を超えていないと判断されるまで、現在の前記第２のキーに新たな前記第２の追加キーを順次に追加して、前記新たな第２のキーを作成することを更に含む、
請求項８に記載のコンピュータプログラム。
【請求項１０】
前記第２のキーの前記出現回数が前記所定の許容値を超えていると判断される場合に、該第２のキーの前記所定のサンプリング間隔を大きくする、
請求項８又は９に記載のコンピュータプログラム。
【請求項１１】
前記近似文字列を導出することは、
前記マッピングにより同定された前記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を受信することと、
前記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ために、所定のアラインメント処理を実行することと、
導出された前記少なくとも１つの近似文字列を出力することと、を含む、
請求項８から１０のいずれか一項に記載のコンピュータプログラム。
【請求項１２】
前記所定のアラインメント処理を実行することは、
前記被照合文字列と前記照合文字列とに基づいて所定のアラインメント表を作成することと、
前記アラインメント表の対角線上の要素を中心にした幅ｍを有する計算領域を設定することと、
設定された前記計算領域における各要素について、変異度を算出することと、
算出された前記変異度に基づいて、最大変異度を決定することと、

決定された前記最大変異度に基づいて、前記少なくとも１つの近似文字列を導出することを含む、
請求項１１に記載のコンピュータプログラム。
【請求項１３】
前記所定のアラインメント処理を実行することは、
前記最大変異度と所定の下限値とを比較して、前記最大変異度が前記所定の下限値を超えているかを判断することと、
前記最大変異度が前記所定の下限値を超えていないと判断される場合に、新たな計算領域を設定するために、前記計算領域の前記幅ｍを拡幅することと、
前記最大変異度が前記所定の下限値を超えていると判断される場合に、前記最大変異度を有する要素に基づいて、前記少なくとも１つの近似文字列を導出することと、を含み、
前記最大変異度が前記下限値を超えると判断されるまで、前記計算領域を拡幅することにより新たな計算領域を設定して前記変異度を算出することを繰り返す、
請求項１２に記載のコンピュータプログラム。
【請求項１４】
前記所定の下限値は、所定の要素列にｍ個の連続したギャップがあり、それ以外の部分は一致したと仮定した場合の変異度の値である、
請求項１３に記載のコンピュータプログラム。
【請求項１５】
前記一致文字列に対して前記参照文字列における対応する所定の文字列を追加することにより前記被照合文字列を作成することと、
前記一致文字列に対して前記クエリ文字列における対応する所定の文字列を追加することにより前記照合文字列を作成することと、を更に含む、
請求項１１から１４のいずれか一項に記載のコンピュータプログラム。
【請求項１６】
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を探索するための階層的インデックスを作成する方法を実現させるためのコンピュータプログラムであって、
前記方法は、
前記参照文字列から所定長の各第１のキーを切り出すことと、
切り出された前記各第１のキーについて、所定のハッシュ関数により該第１のキーに基づいて算出されるハッシュ値を割り当てた第１のキー配列を作成することと、
作成された前記第１のキー配列を更新することと、
更新された前記第１のキー配列を前記階層的インデックスとして出力することと、を含み、
前記第１のキー配列を更新することは、
前記第１のキー配列における前記各第１のキーについて、前記参照文字列における該第１のキーの出現開始位置及び出現回数を同定することと、
同定された前記第１のキーの前記出現開始位置及び前記出現回数に従って、該第１のキーに第１の追加キーを追加することにより新たな第１のキーを作成し、該新たな第１のキーに基づいて前記第１のキー配列を更新することと、を含む、
コンピュータプログラム。
【請求項１７】
コンピューティングデバイスに、参照文字列に対してクエリ文字列のマッピングを行う方法を実現させるためのコンピュータプログラムであって、
前記方法は、
前記参照文字列に基づく階層的インデックスを読み出すことと、
前記クエリ文字列から所定のキー長を有する各キーを切り出して、キー配列を作成することと、
前記クエリ文字列から切り出された前記各キーについて、前記所定のハッシュ関数によ

り該キーに基づいて算出されるハッシュ値を割り当てたキー配列を作成することと、
前記各キーについて、前記ハッシュ値に従って、所定のサンプリング間隔で、前記階層的インデックスを参照し、該キーの出現開始位置及び出現回数を同定することと、
前記同定した出現回数が所定のしきい値を超えているか否かを判断することと、
前記同定された前記出現回数が所定の許容値を超えていると判断される場合に、前記キーに対して前記クエリ文字列における該キーに続く少なくとも１以上の文字からなる追加キーを追加することにより新たなキーを作成することと、
前記同定された前記出現回数が所定のしきい値を超えていないと判断される場合に、同定された現在の前記キーの出現開始位置及び該キーを出力することと、を含み、
前記キーの前記出現開始位置及び前記出現回数を同定することは、前記同定された前記出現回数が所定のしきい値を超えていないと判断されるまで、現在の前記キーに新たな前記追加キーを順次に追加して、前記新たなキーを作成することを含む、
コンピュータプログラム
【請求項１８】
前記方法は、前記同定された前記出現回数が所定の許容値を超えていると判断される場合に、前記キーの前記所定のサンプリング間隔を大きくするように構成される、
請求項１７に記載のコンピュータプログラム。
【請求項１９】
コンピューティングデバイスに、参照文字列における部分文字列とクエリ文字列との間の変異を所定のアラインメント処理により同定する方法を実現させるためのコンピュータプログラムであって、
前記方法は、
マッピングにより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を受信することと、
前記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ために、所定のアラインメント処理を実行することと、
導出された前記少なくとも１つの近似文字列を出力することと、を含み、
前記所定のアラインメント処理を実行することは、
前記被照合文字列と前記照合文字列とに基づいて所定のアラインメント表を作成することと、
前記アラインメント表の対角線上の要素を中心にした幅ｍを有する計算領域を設定することと、
設定された前記計算領域における各要素について、変異度を算出することと、
算出された前記変異度に基づいて、最大変異度を決定することと、
決定された前記最大変異度に基づいて、前記少なくとも１つの近似文字列を導出することを含む、
コンピュータプログラム。
【請求項２０】
前記所定のアラインメント処理を実行することは、
前記最大変異度と所定の下限値とを比較して、前記最大変異度が前記所定の下限値を超えているかを判断することと、
前記最大変異度が前記所定の下限値を超えていないと判断される場合に、新たな計算領域を設定するために、前記計算領域の前記幅ｍを拡幅することと、
前記最大変異度が前記所定の下限値を超えていると判断される場合に、前記最大変異度を有する要素に基づいて、前記少なくとも１つの近似文字列を導出することと、を含み、
前記最大変異度が前記下限値を超えると判断されるまで、前記計算領域を拡幅することにより新たな計算領域を設定して前記変異度を算出することを繰り返す、
請求項１９に記載のコンピュータプログラム。
【請求項２１】
前記所定のアラインメント処理を実行することは、所定の要素列にｍ個の連続したギャ

ップがあり、それ以外の部分は一致したと仮定した場合の変異度の値を前記所定の下限値として設定することを更に含む、
請求項２０に記載のコンピュータプログラム。

【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、近似文字列照合技術に関し、特に、ヒトゲノムの解析に利用可能な近似文字列照合装置及び近似文字列照合方法並びに該方法を実現するためのコンピュータプログラムに関する。
【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、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）、また、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】
また、データ片は、マッピングにより得られるヒトゲノム参照配列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（非特許文献２）。
【先行技術文献】
【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)
【非特許文献２】内山将夫　他，「近似文字列照合による全文検索のための接尾辞配列の高速走査法」，２００２年９月１５日，情報処理学会，Vol. 43 No. SIG 9(TOD 15)
【発明の概要】
【発明が解決しようとする課題】
【０００８】
上述した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、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。また、ＢＷＡ－ＭＥＭでは、クエリ文字列の文字数（すなわち、塩基数）に応じた回数だけメモリへのアクセスが必要となるため、メモリへのランダムアクセスが頻繁に発生し、これにより、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、クエリ文字列における部分文字列（これを「キー」と称することがある。）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の総文字数分の回数だけ実行される。これは、ヒトゲノムの解析に用いるデータの量が非常に大きいため、必要なデータ配列を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れないからである。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】
また、上述したように、シークエンサにより一人分のヒトゲノムを読み出すためには、現状、１時間ほど要している。一方で、高性能コンピュータを用いてＢＷＡ－ＭＥＭを実行した場合、シークエンサの読み出し時間以内にマッピングが完了し、両者のバランスは概ね保たれている。
【００１１】
一方で、次世代型シークエンサは、低コスト化が進み、また、読み出し時間を更に短縮し得ると言われており、これに伴って、解析時間もまた短縮することが望まれる。解析時間を短縮するための一つのアプローチとして、コンピュータの更なる高性能化が考えられるが、コンピュータの高性能化のためには非常にコストが嵩むため実用化へのハードルが高い。
【００１２】
更に、マッピングされたデータ片の解析に用いられる従前のアラインメントアルゴリズムでは、アラインメント表の全ての要素について変異度を算出するため、計算量が多くなり、時間がかかるという問題がある。
【００１３】
そこで、本発明は、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／又は効率的に行うことができる新たな技術を提供することを目的とする。
【００１４】
より具体的には、本発明の一つの目的は、与えられるクエリ文字列と参照文字列との間の近似文字列照合を高速及び／又は効率的に行うことができる近似文字列照合装置及びこ

れを用いた近似文字列照合方法を提供することである。
【００１５】
また、本発明の一つの目的は、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近似文字列照合装置及びこれを用いた近似文字列照合方法を提供することである。
【００１６】
また、本発明の一つの目的は、前記近似文字列照合に適合した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。
【課題を解決するための手段】
【００１７】
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。
【００１８】
ある観点に従う本発明は、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近似文字列を検索するための方法を実現させるためのコンピュータプログラムである。
前記方法は、前記参照文字列に基づいて階層的インデックスを作成することと、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、前記マッピングにより同定される少なくとも１以上の前記部分文字列に基づいて、前記近似文字列を導出することと、を含む。
ここで、前記階層的インデックスを作成することは、前記参照文字列から所定長の各第１のキーを切り出すことと、切り出された前記各第１のキーについて、所定のハッシュ関数により該第１のキーに基づいて算出されるハッシュ値を割り当てた第１のキー配列を作成することと、作成された前記第１のキー配列を更新することと、更新された前記第１のキー配列を前記階層的インデックスとして出力することと、を含む。
また、前記第１のキー配列を更新することは、前記第１のキー配列における前記各第１のキーについて、前記参照文字列における該第１のキーの出現回数を同定することと、同定された前記第１のキーの前記出現回数に従って、該第１のキーに第１の追加キーを追加することにより新たな第１のキーを作成し、該新たな第１のキーに基づいて前記第１のキー配列を更新することと、を含む。
【００１９】
前記第１のキー配列を作成することは、前記ハッシュ値に従って前記第１のキー配列における前記各第１のキーをソートすることを含み得る。
【００２０】
また、前記第１のキー配列を更新することは、前記同定した出現回数が所定の許容値を超えているか否かを判断することと、前記同定された前記出現回数が所定の許容値を超えていると判断される場合に、前記第１のキーに対して前記参照文字列における該第１のキーに続く少なくとも１以上の文字からなる前記第１の追加キーを追加することにより前記新たな第１のキーを作成することと、前記新たな第１のキーについて、前記参照文字列における該第１のキーの出現回数を同定することと、を含み得る。
【００２１】
また、前記第１のキー配列を更新することは、前記第１の追加キーに従って前記第１のキー配列における前記新たな第１のキーをソートすることを更に含み得る。
【００２２】
また、前記第１のキー配列を更新することは、前記同定された前記出現回数が所定の許容値を超えていないと判断されるまで、現在の前記第１のキーに新たな前記第１の追加キーを順次に追加することにより新たな前記第１のキーを作成することを含み得る。
【００２３】
また、前記キー配列を前記階層的インデックスとして出力することは、前記同定された

前記出現回数が所定の許容値を超えていないと判断される場合に、現在の前記キー配列を前記階層的インデックスとして出力することを含み得る。

【００２４】

前記マッピングを行うことは、前記クエリ文字列から所定長の各第２のキーを切り出すことと、前記クエリ文字列から切り出された前記各第２のキーについて、前記所定のハッシュ関数により該第２のキーに基づいて算出されるハッシュ値を割り当てた第２のキー配列を作成することと、前記各第２のキーについて、前記ハッシュ値に従って、所定のサンプリング間隔で、前記階層的インデックスを参照し、該第２のキーの出現開始位置及び出現回数を同定することと、を含み得る。

【００２５】

前記第２のキーの前記出現開始位置及び前記出現回数を同定することは、前記第２のキーの前記出現回数が前記所定の許容値を超えているか否かを判断することと、前記第２のキーの前記出現回数が前記所定の許容値を超えていると判断される場合に、前記第２のキーに対して前記クエリ文字列における該第２のキーに続く少なくとも１以上の文字からなる第２の追加キーを追加することにより新たな第２のキーを作成することと、前記第２のキーの前記出現回数が前記所定の許容値を超えていないと判断される場合に、同定された現在の前記第２のキーを一致文字列として出力するとともに該第２のキーの前記出現開始位置を出力することと、を含み得る。

【００２６】

また、前記第２のキーの前記出現開始位置及び前記出現回数を同定することは、前記第２のキーの前記同定された前記出現回数が前記所定の許容値を超えていないと判断されるまで、現在の前記第２のキーに新たな前記第２の追加キーを順次に追加して、前記新たな第２のキーを作成することを更に含み得る。

【００２７】

また、前記第２のキーの前記出現回数が前記所定の許容値を超えていると判断される場合に、該第２のキーの前記所定のサンプリング間隔が大きくなるように変更され得る。

【００２８】

前記近似文字列を導出することは、前記マッピングにより同定された前記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を受信することと、前記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ために、所定のアラインメント処理を実行することと、導出された前記少なくとも１つの近似文字列を出力することと、を含み得る。

【００２９】

前記所定のアラインメント処理を実行することは、前記被照合文字列と前記照合文字列とに基づいて所定のアラインメント表を作成することと、前記アラインメント表の対角線上の要素を中心にした幅ｍを有する計算領域を設定することと、設定された前記計算領域における各要素について、変異度を算出することと、算出された前記変異度に基づいて、最大変異度を決定することと、決定された前記最大変異度に基づいて、前記少なくとも１つの近似文字列を導出することを含み得る。

【００３０】

また、前記所定のアラインメント処理を実行することは、前記最大変異度と所定の下限値とを比較して、前記最大変異度が前記所定の下限値を超えているかを判断することと、前記最大変異度が前記所定の下限値を超えていないと判断される場合に、新たな計算領域を設定するために、前記計算領域の前記幅ｍを拡幅することと、前記最大変異度が前記所定の下限値を超えていると判断される場合に、前記最大変異度を有する要素に基づいて、前記少なくとも１つの近似文字列を導出することと、を含み得る。そして、前記方法は、前記最大変異度が前記下限値を超えると判断されるまで、前記計算領域を拡幅することにより新たな計算領域を設定して前記変異度を算出することを繰り返すように構成され得る。

【００３１】

前記所定の下限値は、所定の要素列にｍ個の連続したギャップがあり、それ以外の部分は完全又は実質的に一致したと仮定した場合の変異度の値であり得る。

【００３２】

また、前記方法は、前記一致文字列に対して前記参照文字列における対応する所定の文字列を追加することにより前記被照合文字列を作成することと、前記一致文字列に対して前記クエリ文字列における対応する所定の文字列を追加することにより前記照合文字列を作成することと、を更に含み得る。

【００３３】

また、ある観点に従う本発明は、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を探索するための階層的インデックスを作成する方法を実現させるためのコンピュータプログラムである。

前記方法は、前記参照文字列から所定長の各第１のキーを切り出すことと、切り出された前記各第１のキーについて、所定のハッシュ関数により該第１のキーに基づいて算出されるハッシュ値を割り当てた第１のキー配列を作成することと、作成された前記第１のキー配列を更新することと、更新された前記第１のキー配列を前記階層的インデックスとして出力することと、を含む。

ここで、前記第１のキー配列を更新することは、前記第１のキー配列における前記各第１のキーについて、前記参照文字列における該第１のキーの出現開始位置及び出現回数を同定することと、同定された前記第１のキーの前記出現開始位置及び前記出現回数に従って、該第１のキーに第１の追加キーを追加することにより新たな第１のキーを作成し、該新たな第１のキーに基づいて前記第１のキー配列を更新することと、を含む。

【００３４】

また、ある観点に従う本発明は、コンピューティングデバイスに、参照文字列に対してクエリ文字列のマッピングを行う方法を実現させるためのコンピュータプログラムである。

前記方法は、前記参照文字列に基づく階層的インデックスを読み出すことと、前記クエリ文字列から所定のキー長を有する各キーを切り出して、キー配列を作成することと、前記クエリ文字列から切り出された前記各キーについて、前記所定のハッシュ関数により該キーに基づいて算出されるハッシュ値を割り当てたキー配列を作成することと、前記各キーについて、前記ハッシュ値に従って、所定のサンプリング間隔で、前記階層的インデックスを参照し、該キーの出現開始位置及び出現回数を同定することと、前記同定した出現回数が所定のしきい値を超えているか否かを判断することと、前記同定された前記出現回数が所定の許容値を超えていると判断される場合に、前記キーに対して前記クエリ文字列における該キーに続く少なくとも１以上の文字からなる追加キーを追加することにより新たなキーを作成することと、前記同定された前記出現回数が所定のしきい値を超えていないと判断される場合に、同定された現在の前記キーの出現開始位置及び該キーを出力することと、を含む。

そして、前記キーの前記出現開始位置及び前記出現回数を同定することは、前記同定された前記出現回数が所定のしきい値を超えていないと判断されるまで、現在の前記キーに新たな前記追加キーを順次に追加して、前記新たなキーを作成することを含む。

【００３５】

ここで、前記同定された前記出現回数が所定の許容値を超えていると判断される場合に、前記キーの前記所定のサンプリング間隔が大きくなるように変更され得る。

【００３６】

また、ある観点に従う本発明は、コンピューティングデバイスに、参照文字列における部分文字列とクエリ文字列との間の変異を所定のアラインメント処理により同定する方法を実現させるためのコンピュータプログラムである。

前記方法は、マッピングにより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を受信することと、前記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ために、所定のアラインメント処理を実行することと、

導出された前記少なくとも１つの近似文字列を出力することと、を含む。
そして、前記所定のアラインメント処理を実行することは、前記被照合文字列と前記照合文字列とに基づいて所定のアラインメント表を作成することと、前記アラインメント表の対角線上の要素を中心にした幅ｍを有する計算領域を設定することと、設定された前記計算領域における各要素について、変異度を算出することと、算出された前記変異度に基づいて、最大変異度を決定することと、決定された前記最大変異度に基づいて、前記少なくとも１つの近似文字列を導出することを含む。
【００３７】
また、前記所定のアラインメント処理を実行することは、前記最大変異度と所定の下限値とを比較して、前記最大変異度が前記所定の下限値を超えているかを判断することと、前記最大変異度が前記所定の下限値を超えていないと判断される場合に、新たな計算領域を設定するために、前記計算領域の前記幅ｍを拡幅することと、前記最大変異度が前記所定の下限値を超えていると判断される場合に、前記最大変異度を有する要素に基づいて、前記少なくとも１つの近似文字列を導出することと、を含み得る。
そして、前記最大変異度が前記下限値を超えると判断されるまで、前記計算領域を拡幅することにより新たな計算領域を設定して前記変異度を算出することが繰り返され得る。
【００３８】
また、前記所定のアラインメント処理を実行することは、所定の要素列にｍ個の連続したギャップがあり、それ以外の部分は完全に又は実質的に一致したと仮定した場合の変異度の値を前記所定の下限値として設定することを更に含み得る。
【００３９】
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。近似文字列照合装置は、本発明の一形態である。
【００４０】
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。また、「システム」とは、複数の装置（又は特定の機能を実現する機能モジュール）が論理的に集合した物のことをいい、各装置や機能モジュールが単一の筐体内にあるか否かは特に問わない。
【発明の効果】
【００４１】
本発明によれば、与えられるクエリデータに対する参照データ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。とりわけ、本発明によれば、与えられるクエリ文字列と参照文字列との間の近似文字列照合を高速及び／又は効率的に行うことができる。
【００４２】
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。
【００４３】
更に、本発明によれば、近似文字列照合に適合した参照文字列に基づく階層的インデックスを提供することができる。
【００４４】
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】
【００４５】
【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、本発明の一実施形態に係る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、本発明の一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。
【図４】図４は、本発明の一実施形態に係るコンピュータシステムにおいて用いられる参照文字列の一例を示す図である。
【図４Ａ】図４Ａは、図４に示される参照文字列に基づく階層的インデックスの作成過程におけるデータ配列構造の一例を説明するための図である。
【図４Ｂ】図４Ｂは、図４に示される参照文字列に基づく階層的インデックスの作成過程におけるデータ配列構造の一例を説明するための図である。
【図４Ｃ】図４Ｃは、図４に示される参照文字列に基づく階層的インデックスの作成過程におけるデータ配列構造の一例を説明するための図である。
【図５】図５は、図４に示される参照文字列に基づく階層的インデックスの作成過程におけるキー開始位置及び出現回数を示すテーブル構造の一例を示す図である。
【図６】図６は、本発明の一実施形態に係るコンピュータシステムにおいて用いられる参照文字列の一例を示す図である。
【図７】図７は、本発明の一実施形態に係るコンピュータシステムによるクエリ文字列に基づく参照文字列の探索処理の一例を説明するフローチャートである。
【図８Ａ】図８Ａは、Ｎｅｅｄｌｅｍａｎ－Ｗｕｎｓｃｈアルゴリズムを説明するためのアラインメント表の一例を示す図である。
【図８Ｂ】図８Ｂは、Ｎｅｅｄｌｅｍａｎ－Ｗｕｎｓｃｈアルゴリズムを説明するためのアラインメント表の一例を示す図である。
【図８Ｃ】図８Ｃは、Ｎｅｅｄｌｅｍａｎ－Ｗｕｎｓｃｈアルゴリズムを説明するためのアラインメント表の一例を示す図である。
【図８Ｄ】図８Ｄは、Ｎｅｅｄｌｅｍａｎ－Ｗｕｎｓｃｈアルゴリズムを説明するためのアラインメント表の一例を示す図である。
【図９】図９は、本発明の一実施形態に係るコンピュータシステムによる動的計画法を用いたアラインメント処理の一例を説明するフローチャートである。
【図１０Ａ】図１０Ａは、本発明の一実施形態に係る動的計画法を用いたアラインメントを説明するためのアラインメント表の一例を示す図である。
【図１０Ｂ】図１０Ｂは、本発明の一実施形態に係る動的計画法を用いたアラインメントを説明するためのアラインメント表の一例を示す図である。
【図１０Ｃ】図１０Ｃは、本発明の一実施形態に係る動的計画法を用いたアラインメントを説明するためのアラインメント表の一例を示す図である。
【図１１Ａ】図１１Ａは、本発明の一実施形態に係る動的計画法を用いたアラインメントにより得られる近似文字列の一例を示す図である。
【図１１Ｂ】図１１Ｂは、本発明の一実施形態に係る動的計画法を用いたアラインメントにより得られる近似文字列の一例を示す図である。
【図１２】本発明の一実施形態に係るシステムにおける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。
【発明を実施するための形態】
【００４６】
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。
【００４７】

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、所定のインター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。上位コンピュータ１０及び下位コンピュータ２０のハードウェア構成は、図１２に例示されるが、当業者にとって自明であるため、その詳細な説明は省略する。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。

【００４８】

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。

【００４９】

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、プロセッサの制御の下、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。

【００５０】

すなわち、同図に示すように、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（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される。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データベース３０に格納されている場合には、処理Ｓ２０１は省略され得る。

【００５１】

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て、マッピングを行う（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。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。

【００５２】
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、クエリ文字列における各キーについて、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各キー及びそ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、メインメモリ又はキャッシュメモリ等の高速メモリに展開され得る。

【００５３】
また、本開示に係るマッピングでは、階層的インデックスの探索に際して、クエリ文字列におけるキーの出現開始位置のサンプリング間隔が従来技術に比較してある程度大きくなるように調整される。例えばＢＬＡＳＴ（Basic Local Alignment Search Tool）と称されるＤＮＡの塩基配列やタンパク質のアミノ酸配列のシーケンスアラインメントを行うためのアルゴリズムでは、キーの出現開始位置の間隔は３～５文字に設定されるが、本開示に係る探索では、例えば１０文字又はそれ以上に設定され得る。探索に際して、キーの出現開始位置のサンプリング間隔が小さいと、探索の回数が増え、効率が低下してしまうのに対して、キーの出現開始位置のサンプリング間隔を常に大きくしてしまうと、探索の高速化と引き換えに、見落としの確率が上昇してしまう。そこで、本開示では、キーの長さを長くしながら、それが一定長を超える場合には、出現開始位置のサンプリング間隔を大きくすることにより、キーの一致を見落とすことを防ぐとともに、探索効率の向上を図っている。

【００５４】
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。

【００５５】
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（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。

【００５６】
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。

【００５７】
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。

【００５８】
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。

【００５９】

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（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。

【００６０】

以上のように、本実施形態の近似文字列照合では、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。

【００６１】

図３は、本発明の一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４は、階層的インデックスを作成するための参照文字列の一例を示す図であり、図４Ａ～４Ｃは、参照文字列に基づく階層的インデックスの作成過程におけるデータ配列構造の一例を示す図である。更に、図５は、階層的インデックスの作成過程におけるキーの出現開始位置及び出現回数を示すテーブル構造の一例を示す図であり、図６は、階層的インデックスを説明するための図である。

【００６２】

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（Ｓ３０１）。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、４種類の文字「Ａ」、「Ｃ」、「Ｇ」、及び「Ｔ」から構成される参照文字列「ＣＣＧＡＴＣＴＧＴＡＴＡＣＣＣＴＡＣＧＡ」を例にして説明する（図４参照）。

【００６３】

上位コンピュータ１０は、受信した参照文字列について、所定のキー長に従って各部分文字列（すなわち、キー）を切り出して、キー配列を作成する（Ｓ３０２）。例えば、キー長が「２」である場合、キー配列は、図４Ａ（ａ）のようになる。同図中、左端の番号は、配列番号である。また、参照文字列の末端である１９番目の「Ａ」で始まるキーは、説明の簡略化のため、ここでは省略している。

【００６４】

次に、上位コンピュータ１０は、作成したキー配列の各キーについて、所定のハッシュ関数を用いてハッシュ値を算出し、これを該キーに割り当てる（Ｓ３０３）。これにより、キー配列は、図４Ａ（ｂ）のようになる。本開示におけるハッシュ関数は、４種類の文字「Ａ」、「Ｃ」、「Ｇ」、及び「Ｔ」にそれぞれ割り当てた「０」～「３」の数値により、４進数で表現した値を出力する関数として定義されるが、これに限られない。例えば、１番目のキー「ＣＧ」については、ハッシュ値は、ｈ（ＣＧ）＝１×４＋２＝６となり、また、１１番目のキー「ＡＣ」については、ハッシュ値は、ｈ（ＡＣ）＝０×４＋１＝

１となる。

【００６５】

次に、上位コンピュータ１０は、キー配列の各キーを、割り当てたハッシュ値に従って、例えば昇順にソートする（Ｓ３０４）。これにより、キー配列は、図４Ａ（ｃ）のようになる。本例では、ソート後の各キー配列は、ソート前の配列番号を含み得る。例えば、図４Ａ（ｃ）中、ソート後のキー配列における０番目のキー「ＡＣ」は、ソート前の（元の）配列番号「１１」を保持し、また、ソート後の１番目のキー「ＡＣ」は、ソート前の配列番号「１６」を保持している。

【００６６】

なお、上記の例では、切り出された各キーについて、算出したハッシュ値を割り当てて、ソートするものとしているが、これに限られない。例えば、参照文字列から切り出されるキーに拘わらず、参照文字列に現れる全ての文字の組み合わせに基づいてハッシュ値を算出して割り当てたキー配列を用意し、切り出されるキーに対応する出現開始位置を割り当てても良い。すなわち、４種類の文字「Ａ」、「Ｃ」、「Ｇ」、及び「Ｔ」から構成される参照文字列において、例えば、キーの長さが２文字であれば、4^2個の要素を有するキー配列がまず作成され、更に、ハッシュ値がそれぞれ割り当てられる。続いて、切り出されたキーは、参照文字列における出現開始位置とともに、作成されたキー配列における対応する要素（同じキーの要素）に割り当てられることにより、図４Ａ（ｃ）に示すようなキー配列が得られる。

【００６７】

次に、上位コンピュータ１０は、現在のキー配列における各キーの出現開始位置及び出現回数を同定する（Ｓ３０５）。図５（ａ）は、図４Ａ（ｃ）に示されるキー配列における各キーの出現開始位置及び出現回数を示している。例えば、キー「ＡＣ」は、キー配列において、出現開始位置「０」（配列番号「０」）を基点にして２回出現することが示されている。また、キー「ＣＣ」は、出現開始位置「４」を基点にして３回出現することが示されている。なお、ここでは、各文字どうしの全ての組み合わせからなるキーのパターン（すなわち、１６パターン）に対するその出現開始位置及び出現回数が示されており、例示したキー配列に含まれていない例えばキー「ＡＡ」については、出現開始位置「－」及び出現回数「０」のように示されている。

【００６８】

次に、上位コンピュータ１０は、各キーについて、その出現回数が所定の許容値を超えているか否かを判断する（Ｓ３０６）。本開示において、所定の許容値は、階層的インデックスにおいて同じキーが重複して存在し得ることを許容する値である。本例では、所定の許容値は「１」としている。つまり、所定の許容値が「１」であれば、階層的インデックスにおいて各キーは唯一の存在となる。また、所定の許容値が大きいほど、階層的インデックスにおいて重複したキーが存在する可能性が高くなる一方、階層的インデックスの作成は高速化される。上位コンピュータ１０は、出現回数が所定の許容値を超えているキーがあると判断する場合（Ｓ３０６のＹｅｓ）、そのキーに対して追加キーを追加する（Ｓ３０８）。

【００６９】

追加キーは、元の文字列における該キーに続く１以上の文字である。本例では、追加キーは１文字としている。追加キーの追加により得られる部分文字列は、新たなキーとみなされる。以下では、追加キーが追加された新たなキーを元のキーと区別するために「拡張キー」と称し、その配列を拡張キー配列と称する場合がある。追加キーが追加されることにより、各キーどうしが異なるものとして識別されることになる。図４Ｂ（ａ）は、元のキーに１個の追加キーが追加された拡張キーからなる拡張キー配列の一例を示している。なお、配列番号「１２」の拡張キー「ＧＡ」については、元の文字列において「Ａ」に続く文字がないため、終端文字列として例えば「＄」を割り当てている。また、配列番号「１３」、「１７」及び「１８」のキーについては、その出現回数が１回であるため、追加キーは追加されない。

【００７０】
次に、上位コンピュータ１０は、各キー（すなわち、拡張キー）を、図４Ｂ（ｂ）に示すように、該追加キーに従って例えば昇順に更にソートする（Ｓ３０８）。この場合、元のキーのソート順が優先される。同図中、例えば、配列番号「２」及び「３」のキー配列「ＡＴ」については、追加キーによるソートで、その順序が入れ替わっていることがわかる。続いて、上位コンピュータ１０は、処理Ｓ３０６に戻り、全てのキーの出現回数が所定の許容値を超えなくなるまで上記の処理を繰り返す。
【００７１】
すなわち、上位コンピュータ１０は、キー配列における各キーの出現開始位置及び出現回数を同定し（Ｓ３０５）、出現回数が所定の許容値を超えているキーがないと判断する場合（Ｓ３０６のＮｏ）、所望の階層的インデックスが作成されたため、処理を終了する。
【００７２】
図５は、図４に示される参照文字列に基づく階層的インデックスの作成過程におけるキーの出現開始位置及び出現回数を説明するための図である。例えば、図５（ａ）において出現回数が２以上であるキーには、追加キーが追加され（図４Ｂ（ａ））、各キー（拡張キー）は、追加キーに従ってソートされる（図４Ｂ（ｂ））。これにより、図５（ｂ）に示されるように、現在のキー配列における各キーの出現開始位置及び出現回数が同定される。同図に示す例では、拡張キー「ＣＧＡ」及び「ＴＡＣ」の出現回数が２となっている。したがって、これらの拡張キーのそれぞれについて、同様に、追加キーが追加されソートされる（図４Ｃ（ａ）及び（ｂ））。なお、元の配列番号「１７」（ハッシュ値でソート後の配列番号「８」）のキー「ＣＧ」については、キー「Ａ」に続く文字がないため、終端文字列として例えば「＄」が割り当てられている。これにより、図５（ｃ）に示されるように、該キーの出現開始位置及び出現回数が同定される。以上により、全ての拡張キーは、その出現回数が「１」となったため、インデックスの作成処理が終了する。
【００７３】
一例として、キー「ＴＡ」について考える。キー「ＴＡ」は、配列番号「１４」～「１６」にあることから（図４Ａ（ｃ）参照）、図６（ａ）に示すように、その出現開始位置は「１４」、出現回数は「３」となる。次に、キー「ＴＡ」に追加キーが追加されソートされることより（図４Ｂ（ｂ）参照）、図６（ｂ）に示すように、追加キー「Ｃ」を含む拡張キー「ＴＡＣ」については、その出現開始位置は「１４」、出現回数は「２」となる一方、追加キー「Ｔ」を含む拡張キー「ＴＡＴ」については、その出現開始位置は「１６」、出現回数は「１」となる。したがって、出現回数が「２」のキー「ＴＡＣ」について、更に追加キー「Ｃ」及び「Ｇ」がそれぞれ追加され、これにより、図６（ｃ）に示すように、キー「ＴＡＣＧ」の出現回数は「１」となる。このようにして、拡張キー配列は、階層的なツリー構造として把握される。
【００７４】
以上のようにして、上位コンピュータ１０は、例えばヒトゲノム参照配列に基づいて、階層的インデックスを作成する。このような階層的インデックスは、ハッシュ値に従ってソートされているため、特定のキー（部分文字列）に関連する階層的インデックスの部分的なデータ配列構造は、メインメモリにおける特定のアドレス領域に集約的に展開され（データのシーケンシャル化）、これにより、メインメモリ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ようになる。
【００７５】
なお、上記では、簡単化のため、極めて短い文字列を例にして説明したが、例えば、ヒトゲノムを扱う場合には、部分文字列のキー長を１０～２０程度、追加キーのキー長を２～８、所定の許容値を５～４０とすることが好ましく、部分文字列のキー長を１０～１５程度、追加キーのキー長を２～４、所定の許容値を１０～２０とすることがより好ましい。
【００７６】

図７は、本発明の一実施形態に係るコンピュータシステムによるクエリ文字列に基づく参照文字列の探索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７は、図２に示したクエリ文字列に基づく参照文字列の探索処理（Ｓ２０２）の詳細を示している。かかる探索処理により、クエリ文字列における所定のキーが参照文字列にマッピングされ、参照文字列における一致文字列及びその出現開始位置が同定される。なお、以下では、上位コンピュータ１０による探索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による探索処理も同様である。

【００７７】

同図に示すように、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を読み出して、メモリ上に展開し、記憶する（Ｓ７０１）。上位コンピュータ１０は、高速化の観点から、階層的インデックスを構成するデータをメインメモリ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。

【００７８】

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クエリ文字列を受信する（Ｓ７０２）。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、クエリ文字列は「ＴＡＣＣ」であるものとして説明する。

【００７９】

次に、上位コンピュータ１０は、受信したクエリ文字列について、所定のキー長に従って各キーを連続的に切り出す（Ｓ７０３）。本例では、所定のキー長の初期値は「２」であるものとする。したがって、切り出される各キーは、「ＴＡ」、「ＡＣ」、「ＣＣ」、及び「Ｃ＄」となる。また、本例では、各キーのサンプリング間隔の初期値は１であるものとする。キーのサンプリング間隔とは、該キーに従って階層的インデックスを順に参照する開始位置の間隔である。なお、ヒトゲノムの解析であれば、サンプリング間隔の初期値は、例えば３～５程度であり得る。後述するように、サンプリング間隔は、キーの長さに応じて伸長される。

【００８０】

続いて、上位コンピュータ１０は、切り出した各キーについて、所定のハッシュ関数を用いてハッシュ値を算出する（Ｓ７０４）。上述したように、ハッシュ関数は、４種類の文字「Ａ」、「Ｃ」、「Ｇ」、及び「Ｔ」にそれぞれ割り当てた「０」～「３」の数値により、４進数で表現した値を出力する関数として定義される。例えば、キー「ＴＡ」について、ハッシュ値は、ｈ（ＴＡ）＝３×４＋０＝１２となる。

【００８１】

次に、上位コンピュータ１０は、算出したハッシュ値に従って、階層的インデックスを参照し（Ｓ７０５）、参照文字列における各キーの出現開始位置及び出現回数を同定する（Ｓ７０６）。例えば、キー「ＴＡ」であれば、ハッシュ値に従って、階層的インデックスにおける出現開始位置は１４、出現回数は３であることが同定される（図６（ａ）参照）。

【００８２】

次に、上位コンピュータ１０は、各キーについて、その出現回数が所定の回数しきい値を超えているか否かを判断する（Ｓ７０７）。上述したように、本例では、所定の許容値は１としている。上位コンピュータ１０は、キーの出現回数が所定の許容値を超えていると判断する場合（Ｓ７０７のＹｅｓ）、続いて、現在のキー長が所定の上限値を超えているか否かを判断する（Ｓ７０８）。所定の上限値は、例えば、ヒトゲノムの解析であれば、２０程度であり得るが、これに限られない。

【００８３】

上位コンピュータ１０は、各キーについて、現在のキー長が所定の上限値を超えていないと判断する場合（Ｓ７０８のＮｏ）、該キーに対して追加キーを追加することにより、拡張キーを作成し（Ｓ７０９）、Ｓ７０６の処理に戻る。

【００８４】
例えば、上位コンピュータ１０は、キー「ＴＡ」に対して追加キー「Ｃ」を追加して、その出現回数を調べ（図６（ｂ）参照）、更に、追加キーが追加されたキー（拡張キー）に対して追加キー「Ｃ」を追加して、その出現回数を調べ（図６（ｃ）参照）、出現回数が１になるまで繰り返す。
【００８５】
一方、上位コンピュータ１０は、各キーについて、現在のキー長が所定の上限値を超えていると判断する場合（Ｓ７０８のＹｅｓ）、該キーのサンプリング間隔を伸長（大きく）し（Ｓ７１０）、Ｓ７０５の処理に戻る。なお、サンプリング間隔の伸長は、キーについて、所定回数（例えば一回）だけ行われるようにしても良い。
【００８６】
また、上位コンピュータ１０は、現在のキー（すなわち、拡張キー）について、その出現回数が所定の許容値を超えていないと判断する場合（Ｓ７０７のＮｏ）、同定された参照文字列におけるキー（一致文字列）及びその出現開始位置を出力する（Ｓ７１１）。
【００８７】
以上のようにして、上位コンピュータ１０は、階層的インデックスを用いて、参照文字列の中からクエリ文字列の少なくとも一部を探し出すことができる。とりわけ、本実施形態によれば、階層的インデックスの各キーはハッシュ値に従ってソートされているため、クエリ文字列に従った探索は、階層的インデックスにおける部分的・連続的なデータ配列構造に対して行われことになり、メインメモリ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ようになる。
【００８８】
また、階層的インデックスの探索において、現在のキー（拡張キー）の長さが一定長以上になった場合に出現開始位置のサンプリング間隔を伸長するので、探索の回数が効率的に削減され、探索の高速化を図ることができる。一方で、サンプリング間隔の伸長は、本来同定されるべきキーを見逃す確率が高くなる可能性があるが、キーの長さが一定長以上になった場合にサンプリング間隔を長くしているので、このような見逃しの発生を効果的に抑制している。
【００８９】
次に、動的計画法を用いたアラインメントについて説明する。すなわち、上位コンピュータ１０は、参照文字列における同定した出現開始位置近傍の文字列（被照合文字列）に対してクエリ文字列（照合文字列）がどれくらい変異しているか（変異度）を推定するために、動的計画法を用いたアラインメント処理を実行する。
【００９０】
アラインメントとは、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。例えば、文字列Ｘ（すなわち、参照文字列に基づく被照合文字列）と文字列Ｙ（すなわち、クエリ文字列に基づく照合文字列）との比較において、文字列Ｙの一部の文字が置換され、挿入され、又は欠損する場合に、文字列Ｙは文字列Ｘに一致しないと判断される。つまり、文字列Ｘ及びＹのそれぞれにおける各位置の文字どうしの関係は、一致する場合（一致）、一致しない場合（不一致）、及び一方の文字が存在しない場合（ギャップ）のいずれかであるといえる。ここで、一致する場合のスコアを例えば「＋２」、不一致の場合のスコアを例えば「－１」、及びギャップの場合のスコアを例えば「－２」と定義する。そして、文字列Ｘ及び文字列Ｙの各要素（文字）どうしを比較して、これらのスコアを用いて各要素の変異度が算出される。本開示では、変異度の算出のために、グローバルアラインメントの一例であるＮｅｅｄｌｅｍａｎ－Ｗｕｎｓｃｈアルゴリズムをベースにした改良アラインメントアルゴリズムが用いられる。以下、Ｎｅｅｄｌｅｍａｎ－Ｗｕｎｓｃｈアルゴリズムの基本的な考え方を説明し、更に、本発明に適用される改良アラインメントアルゴリムを説明する。
【００９１】

例えば、文字列Ｘ＝ｘ$_1$ｘ$_2$‥ ｘ$_i$と文字列Ｙ＝ｙ$_1$ｙ$_2$‥ ｙ$_j$との比較において、文字ｘ$_i$と文字ｙ$_j$との変異度Ｆ（ｉ，ｊ）は以下のように定義される。

【数１】

$$\mathrm{F(i,j)} = \max\begin{Bmatrix} F(i-1,j-1)+s(x_i,y_j) \\ F(i-1,j)-d \\ F(i,j-1)-d \end{Bmatrix} \quad \cdots式１$$

ここで、ｍａｘは与えられた式の値の中から最大値を出力する関数、ｓはスコア関数（一致：ｓ＝２、不一致：ｓ＝－１、ギャップ：ｓ＝－２）、ｄはギャップによるペナルティ（ｄ＝２）である。

【００９２】

以下では、理解容易のため、被照合文字列「ＧＣＣＴＣＧＣＴ」と照合文字列「ＧＣＣＡＴＴＣＡ」との間での動的計画法を用いたアラインメントを説明する。

【００９３】

図８Ａは、本例における比較対象の文字列どうしを配列したアラインメント表である。表中、「φ」は空文字であり、Ｆ（０，０）＝０とする。また、式１において、変異度Ｆ（ｉ，ｊ）の引数が負の値になる場合、範囲外であるため、計算は省略される（出力をＮｕｌｌとする。）。アラインメント表は、ある種のデータ構造としてメモリ上に展開され、プロセッサの利用に供される。

【００９４】

まず、ｊ＝０の場合において、Ｆ（１，０）については、式１より、ｍａｘ関数内のそれぞれは、

Ｆ（０，－１）＋ｓ（ｘ$_1$，ｙ$_0$）＝Ｎｕｌｌ
Ｆ（０，０）－ｄ＝０－２＝－２
Ｆ（１，－１）－ｄ＝Ｎｕｌｌ

であるから、

Ｆ（１，０）＝－２

となる。

【００９５】

次に、Ｆ（２，０）は、

Ｆ（２，０）＝Ｆ（１，０）－ｄ＝－４

となる。同様にして、

Ｆ（ｎ，０）＝－ｎｄ
Ｆ（０，ｎ）＝－ｎｄ

となるため、アラインメント表は図８Ｂに示すようになる。

【００９６】

次に、ｊ＝１の場合においても、同様に算出される。Ｆ（１，１）については、ｍａｘ関数内のそれぞれは、

Ｆ（０，０）＋ｓ（ｘ$_1$，ｙ$_1$）＝０＋２＝２
Ｆ（０，１）－ｄ＝－２－２＝－４
Ｆ（１，０）－ｄ＝－２－２＝－４

であり、これにより、

Ｆ（１，１）＝２

となり、アラインメント表は図８Ｃに示すようになる。

【００９７】

以上の計算を同様に繰り返すことにより、図８Ｄに示すようなアラインメント表が作成されることになる。作成されたアラインメント表において、要素位置（６，７）の変異度Ｆ（６，７）が最大値「７」を有している。したがって、同図に示すように、要素位置（

６，７）から要素位置（１，１）までバックトラックがなされる。表中、右下斜め方向への矢印は文字の一致を示し、右方向への矢印は欠損を示し、下方向への矢印は挿入を示している。これにより、近似文字列「ＧＣＣ＜Ａ＞Ｔ［Ｔ］Ｇ」が導出されることになる。ただし、記号「＜　＞」は、文字間への挿入を表し、記号「［　］」は置換を表すものとする。つまり、参照文字列「ＧＣＣＴＣＧ」であるところ、クエリ文字列は、参照文字列の「Ｃ」と「Ｔ」の間に「Ａ」が挿入され、参照文字列の「ＴＣＧ」の「Ｃ」が「Ｔ」に置換されていることがわかる。

【００９８】

以上のように、Ｎｅｅｄｌｅｍａｎ－Ｗｕｎｓｃｈアルゴリズムに従って、アラインメント表における変異度が最大値である要素位置を特定することにより、そこから近似文字列を導出することができる。しかしながら、Ｎｅｅｄｌｅｍａｎ－Ｗｕｎｓｃｈアルゴリズムではアラインメント表の全ての要素の変異度を算出するため、計算量が膨大となり、時間がかかっていた。そこで、本開示では、以下のような改良アラインメントアルゴリズムを提案し、これにより、計算量を削減し、処理の高速化を図っている。

【００９９】

すなわち、本発明に適用される改良アラインメントアルゴリズムは、概略的には、アラインメント表における対角線上に位置する要素を中心とする所定の幅を有する計算領域を定め、該計算領域内の要素についてのみ変異度を算出することによりその最大値を決定し、該最大値が所定の条件を満たす場合に、該最大値に基づく要素から近似文字列を導出することを含む。最大値が所定の条件を満たさない場合には、計算領域が拡大され、同様に、変異度が算出されることによりその最大値を決定し、該最大値が所定の条件を満たすまで繰り返される。

【０１００】

図９は、本発明の一実施形態に係るコンピュータシステムによる動的計画法を用いたアラインメント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９は、図２に示したアラインメント処理の（Ｓ２０４）の詳細を示している。なお、以下では、上位コンピュータ１０による一のクエリ文字列に基づくアラインメント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による他のクエリ文字列に基づくアラインメント処理も同様である。

【０１０１】

同図に示すように、上位コンピュータ１０は、被照合文字列と照合文字列とからなる文字列ペアに基づいてアラインメント表を作成する（Ｓ９０１）。上述したように、アラインメント表は、ある種のデータ構造としてメモリ上に展開される。

【０１０２】

次に、上位コンピュータ１０は、計算領域の幅ｍ（ただし、ｍは正数）を初期値に設定する（Ｓ９０２）。幅ｍの初期値は、例えば、マッピングにより得られた一致文字列の長さであり得るが、これに限られない。また、幅ｍは、アラインメント表の対角線上の要素位置を中心とすることから、奇数の値に設定されるが、これに限られるものではない。これにより、アラインメント表の対角線上の要素位置を中心とする幅ｍの計算領域が決定される。

【０１０３】

続いて、上位コンピュータ１０は、計算領域の境界を画定する各要素に所定のダミー値を設定する（Ｓ９０３）。ダミー値は、変異度Ｆの値として十分に小さい値が選択される。例えば、ダミー値は、初期値の幅ｍの例えば２～３倍程度の負の値であり、任意に設定することができる。

【０１０４】

次に、上位コンピュータ１０は、計算領域内の各要素について、式１に従って変異度Ｆを算出する（Ｓ９０４）。続いて、上位コンピュータ１０は、計算領域において最大値を有する最大変異度F_{max}を決定し、その要素の位置を特定する（Ｓ９０５）。なお、最大変異度F_{max}を持つ要素は、１つであるとは限らない。

【０１０５】
　また、上位コンピュータ１０は、アライメント表の行又は列における変異度Ｆの下限値F_{Low}を算出する（Ｓ９０６）。下限値F_{Low}は、該行又は列において、参照文字列とクエリ文字列とを比較して、ｍ個の連続したギャップがあり、それ以外の部分は完全に又は実質的に一致したと仮定した場合の変異度Ｆの値である。すなわち、下限値F_{Low}は、

$$F_{Low}=（文字列の長さ－m）\times s \quad \cdots式２$$

　ただし、ｓ＝２である。
で算出される。
【０１０６】
　次に、上位コンピュータ１０は、最大変異度F_{max}と下限値F_{Low}とを比較して、最大変異度F_{max}が下限値F_{Low}を超えているか否かを判断する（Ｓ９０７）。上位コンピュータ１０は、最大変異度F_{max}が下限値F_{Low}を超えていないと判断する場合（Ｓ９０７のＮｏ）、幅ｍを所定の大きさδだけ拡幅する（Ｓ９０８）。例えば、δは、ｍ＋１とする（ただし、ｍは文字列の文字数を超えないものとする。）。
【０１０７】
　上位コンピュータ１０は、拡幅された幅ｍの計算領域に対して、同様に処理を行い、最大値F_{max}が下限値F_{Low}を超えるまで、上記処理を繰り返す。
【０１０８】
　上位コンピュータ１０は、最大値F_{max}が下限値F_{Low}を超えていると判断する場合（Ｓ９０７のＹｅｓ）、該最大値を持つ要素の位置からバックトラックを行って、近似文字列を決定する（Ｓ９０９）。そして、上位コンピュータ１０は、決定した近似文字列を出力する（Ｓ９１０）。
【０１０９】
　例えば、被照合文字列「ＧＧＧＡＴＣＣＧＡＴＡＡＴＣＧＧＴＣＣＣＣＴＡＧＧ」（２４文字）に対して照合文字列「ＧＧＧＣＡＴＴＣＡＡＣＡＴＡＡＧＴＣＧＧＣＣＴＧ」（２４文字）との間での、本発明に係るアラインメント法による変異度の算出例を説明する。なお、変異度Ｆの算出に式１を用いる点は、上述した例と同様である。なお、被照合文字列の長さと照合文字列の長さとは一致する必要はなく、典型的には、被照合文字列の長さの方が照合文字列の長さよりも長い。
【０１１０】
　まず、上位コンピュータ１０は、比較対象の文字列どうしを配列したアラインメント表を用意し、幅ｍの初期値を設定する。本例では、幅ｍ（０）の初期値は７であるものとする。また、上位コンピュータ１０は、幅ｍに従って規定される計算領域の境界部分の要素にダミー値を設定する。本例では、ダミー値は－２０であるものとする。図１０Ａは、ダミー値が設定されたアラインメント表を示している。表中、ハッチングが描かれている要素が幅ｍ（０）＝７での計算領域Ｒ（０）である。
【０１１１】
　上位コンピュータ１０は、計算領域Ｒ（０）内の各要素の変異度Ｆを式１に従って計算する。図１０Ｂは、計算領域内の各要素の変異度Ｆが算出された状態を示している。上位コンピュータ１０は、算出された変異度Ｆの中から、最大変異度F_{max}を決定する。本例では、最大変異度F_{max}は１７である。図中、最大変異度F_{max}が１７である要素にはハッチングが示されている。
【０１１２】
　続いて、上位コンピュータ１０は、アラインメント表の行又は列における下限値F_{Low}を算出する。本例では、下限値F_{Low}は、

$$\begin{aligned} F_{Low}&=（２４－m）\times s \\ &=１７\times２ \\ &=３４ \end{aligned}$$

となる。

【０１１３】
続いて、上位コンピュータ１０は、最大変異度F_{max}と下限値F_{Low}とを比較し、これにより、最大変異度F_{max}が下限値F_{Low}を超えていないと判断するため、幅ｍをδだけ拡幅する。本例では、拡幅された幅ｍ（１）を１５に拡幅する。また、拡幅された幅ｍ（１）の計算領域をＲ（１）とする。
【０１１４】
上位コンピュータ１０は、同様にして、計算領域Ｒ（１）内の各要素の変異度Ｆを式１に従って算出する。図１０Ｃは、計算領域内の各要素の変異度Ｆが算出された状態を示している。図中、計算領域Ｒ（１）に対して拡幅により追加された領域にハッチングが描かれている。これにより、最大変異度F_{max}は１９となる。また、このときの下限値F_{Low}は１８となる。
【０１１５】
したがって、上位コンピュータ１０は、最大変異度F_{max}が下限値F_{Low}を超えていると判断するため、最大変異度F_{max}＝１９である要素からバックトラックし、これにより得られるパスに従って近似文字列を特定する。
【０１１６】
すなわち、図１０Ｃに示す例では、上位コンピュータ１０は、最大変異度F_{max}＝１９である要素（１８，２２）を起点（現在の要素位置）として、要素（０，０）方向に向けて隣接する要素のうち変異度Ｆの値が最も大きい要素を同定し、そこに遷移する。したがって、変異度Ｆ＝１７を持つ要素（１７，２１）が現在の要素位置となる。このような遷移を要素（０，０）まで繰り返すことにより、最終的に、近似文字列が導出される。なお、バックトラックにより得られるパスは、１つとは限られず、複数である場合がある。
【０１１７】
より具体的には、図１０Ｃに示すアラインメント表において、バックトラックにより得られるパスは６通りあり、各パスに従う近似文字列は、以下のとおりとなる。
（ａ）第１のパス：ＧＧＧ＜Ｃ＞Ａ＜Ｔ＞ＴＣ＜ＡＡ＞Ｃ－ＡＴＡＡ＜Ｇ＞ＴＣＧ［Ｇ］ＣＣ
（ｂ）第２のパス：ＧＧＧ＜Ｃ＞Ａ＜Ｔ＞ＴＣ＜Ａ＞［Ａ］［Ｃ］ＡＴＡＡ＜Ｇ＞ＴＣＧ［Ｇ］ＣＣ
（ｃ）第３のパス：ＧＧＧ＜Ｃ＞ＡＴ［Ｔ］Ｃ＜Ａ＞［Ａ］［Ｃ］ＡＴＡＡ＜Ｇ＞ＴＣＧ［Ｇ］ＣＣ
（ｄ）第４のパス：ＧＧＧ＜Ｃ＞ＡＴ［Ｔ］Ｃ［Ａ］＜ＡＣ＞ＡＴＡＡ＜Ｇ＞ＴＣＧ［Ｇ］ＣＣ
（ｅ）第５のパス：ＧＧＧ＜Ｃ＞Ａ＜Ｔ＞ＴＣ＜ＡＡ＞Ｃ－ＡＴＡＡ＜Ｇ＞ＴＣＧ［Ｇ］ＣＣ
（ｆ）第６のパス：ＧＧＧ＜Ｃ＞ＡＴ＜Ｔ＞Ｃ＜Ａ＞［Ａ］［Ｃ］ＡＴＡＡ＜Ｇ＞ＴＣＧ［Ｇ］ＣＣ
ただし、記号「＜　＞」は、文字間への挿入を表し、記号「［　］」は文字の置換を表し、記号「－」は文字の欠損を表すものとする。
なお、理解容易のため、上記の各パスに従う近似文字列を参照文字列との対比において図１１Ａ及び１１Ｂに示している。
【０１１８】
このように、改良されたアラインメントアルゴリズムでは、アラインメント表の全ての要素について変異度を算出するのではなく、所定の幅を有する計算領域を定め、必要に応じた範囲で計算領域を拡大させながら変異度を算出していくので、計算量を削減し、これにより、処理の高速化を図ることができるようになる。
【０１１９】
以上のように、本実施形態によれば、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。

【０１２０】

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。

【０１２１】

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。

【０１２２】

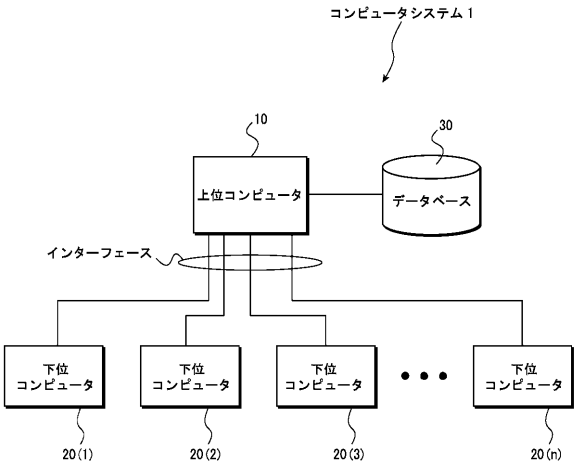
また、本明細書では、さまざまな実施形態が開示されているが、一の実施形態における特定のフィーチャ（技術的事項）を、適宜改良しながら、他の実施形態に追加し、又は該他の実施形態における特定のフィーチャと置換することができ、そのような形態も本発明の要旨に含まれる。

【符号の説明】

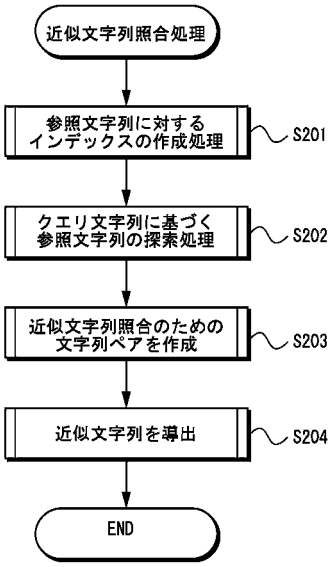
【０１２３】

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

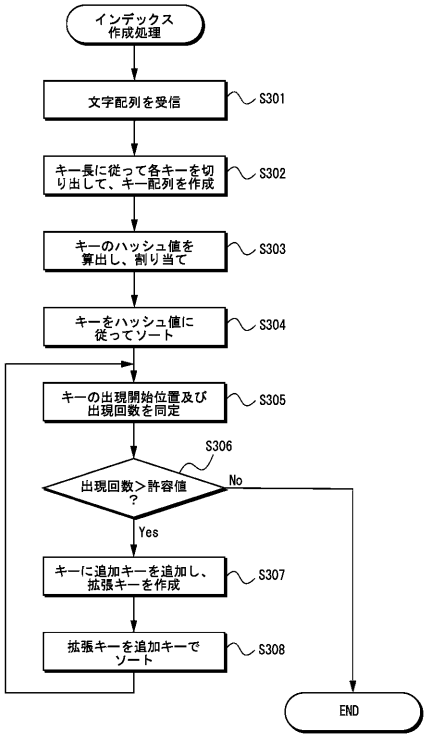
【図１】



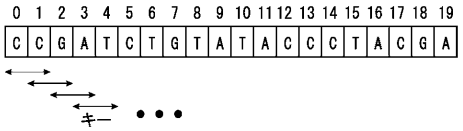
【図２】



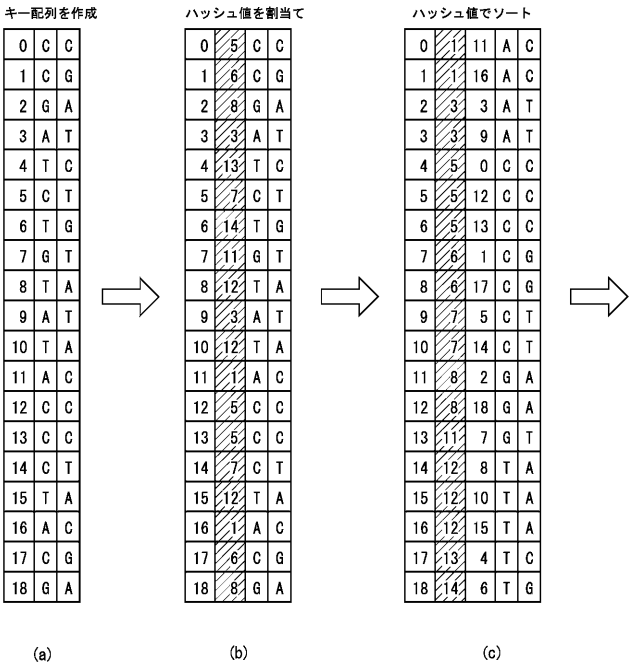
【図３】


インデックス
作成処理
文字配列を受信
S301
キー長に従って各キーを切り出して、キー配列を作成
S302
キーのハッシュ値を算出し、割り当て
S303
キーをハッシュ値に従ってソート
S304
キーの出現開始位置及び出現回数を同定
S305
S306
出現回数＞許容値？
No
Yes
キーに追加キーを追加し、拡張キーを作成
S307
拡張キーを追加キーでソート
S308
END

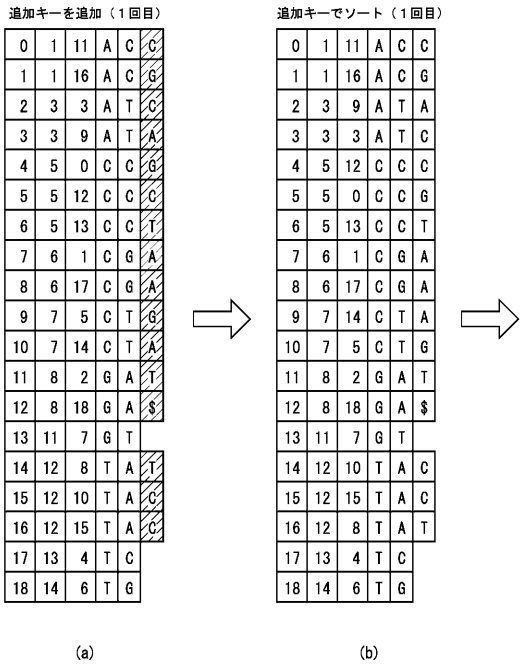
【図４】


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

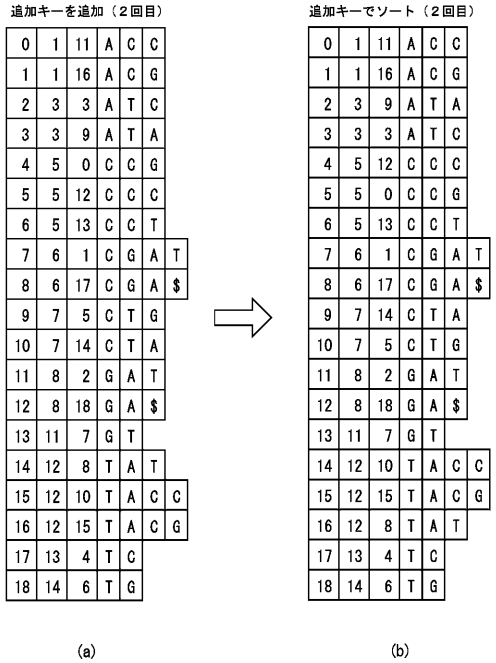
【図４Ａ】


キー配列を作成
ハッシュ値を割当て
ハッシュ値でソート
(a)
(b)
(c)

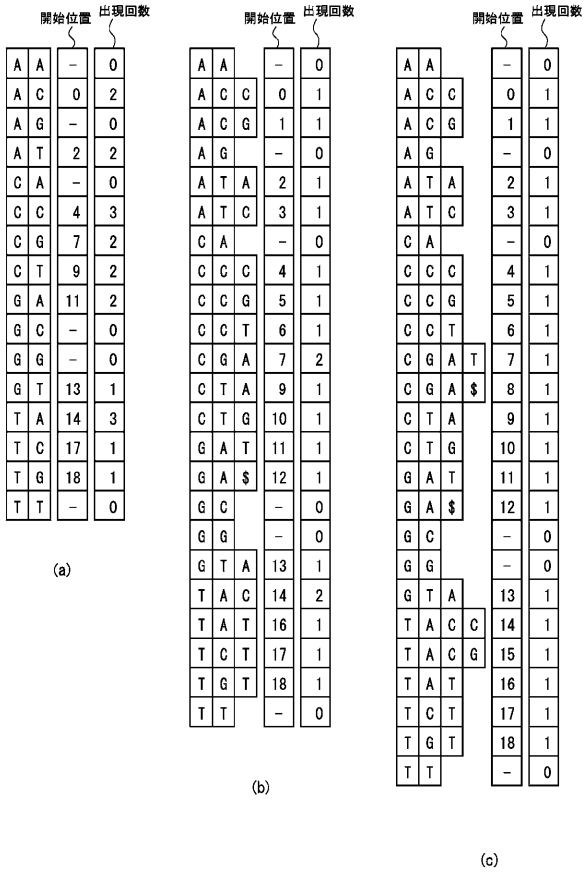
【図４Ｂ】


追加キーを追加（１回目）
追加キーでソート（１回目）
(a)
(b)

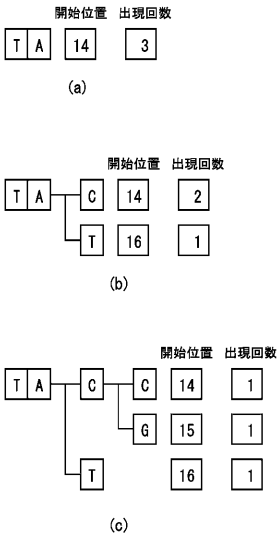
【図４Ｃ】


追加キーを追加（２回目）
追加キーでソート（２回目）
(a)
(b)

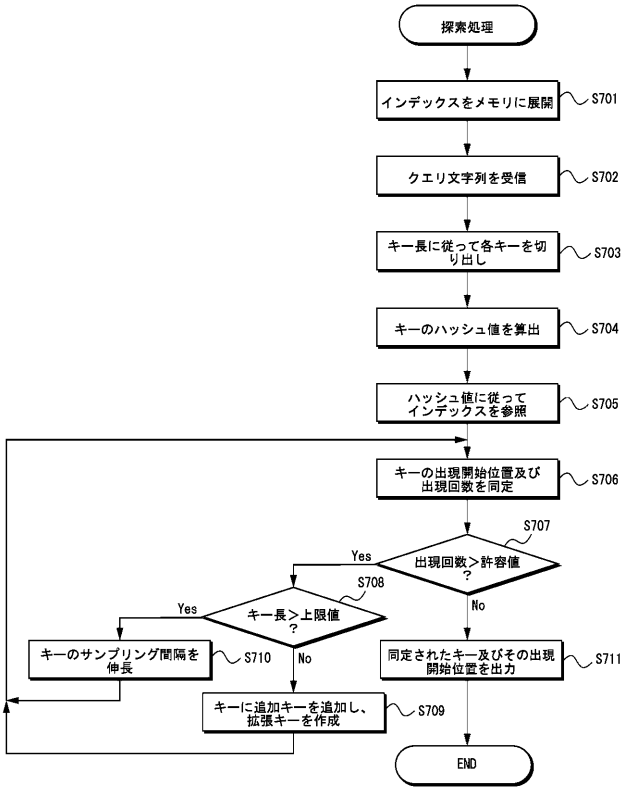
【図５】


開始位置
出現回数
(a)
(b)
(c)

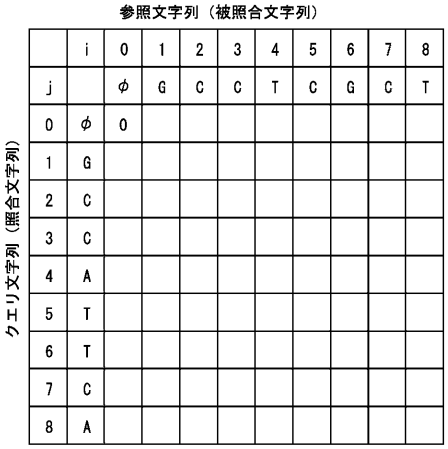
【図６】


開始位置
出現回数
(a)
(b)
(c)

【図７】


探索処理
インデックスをメモリに展開
S701
クエリ文字列を受信
S702
キー長に従って各キーを切り出し
S703
キーのハッシュ値を算出
S704
ハッシュ値に従ってインデックスを参照
S705
キーの出現開始位置及び出現回数を同定
S706
出現回数＞許容値？
S707
Yes
No
キー長＞上限値？
S708
Yes
No
キーのサンプリング間隔を伸長
S710
同定されたキー及びその出現開始位置を出力
S711
キーに追加キーを追加し、拡張キーを作成
S709
END

【図８Ａ】

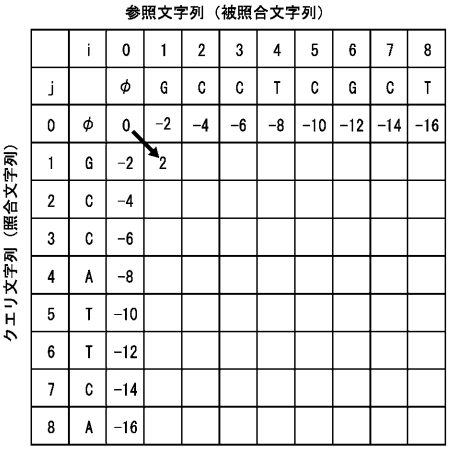


	i	0	1	2	3	4	5	6	7	8
j		φ	G	C	C	T	C	G	C	T
0	φ	0								
1	G									
2	C									
3	C									
4	A									
5	T									
6	T									
7	C									
8	A									

【図８Ｂ】

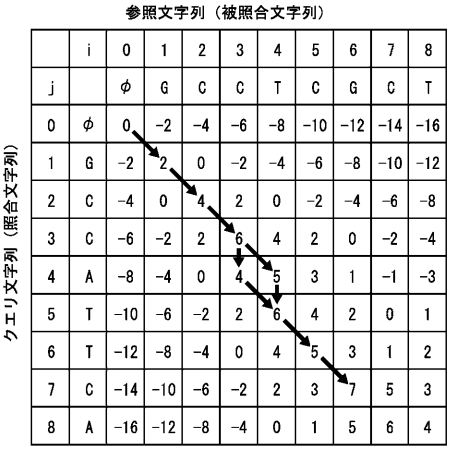
	i	0	1	2	3	4	5	6	7	8
j		φ	G	C	C	T	C	G	C	T
0	φ	0	-2	-4	-6	-8	-10	-12	-14	-16
1	G	-2								
2	C	-4								
3	C	-6								
4	A	-8								
5	T	-10								
6	T	-12								
7	C	-14								
8	A	-16								

【図８Ｃ】



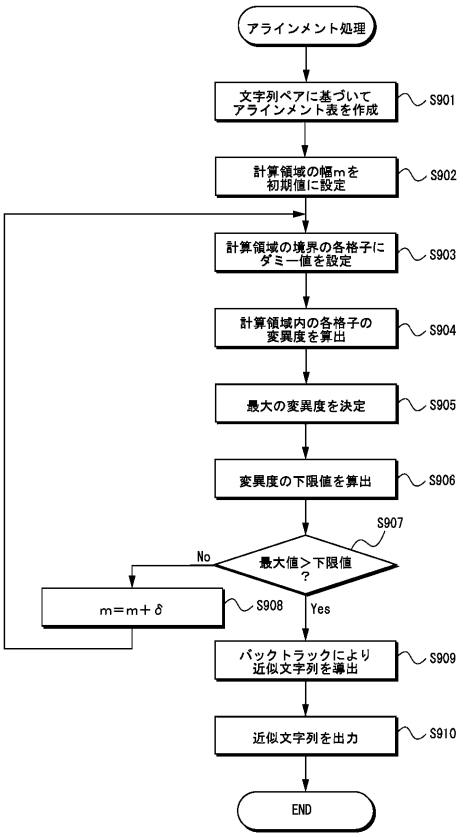
	i	0	1	2	3	4	5	6	7	8
j		φ	G	C	C	T	C	G	C	T
0	φ	0	-2	-4	-6	-8	-10	-12	-14	-16
1	G	-2	2							
2	C	-4								
3	C	-6								
4	A	-8								
5	T	-10								
6	T	-12								
7	C	-14								
8	A	-16								

【図８Ｄ】

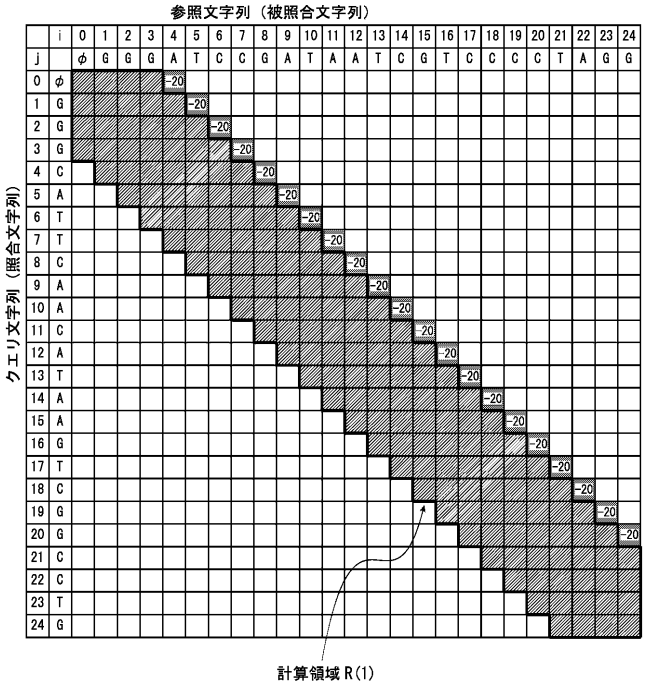


	i	0	1	2	3	4	5	6	7	8
j		φ	G	C	C	T	C	G	C	T
0	φ	0	-2	-4	-6	-8	-10	-12	-14	-16
1	G	-2	2	0	-2	-4	-6	-8	-10	-12
2	C	-4	0	4	2	0	-2	-4	-6	-8
3	C	-6	-2	2	6	4	2	0	-2	-4
4	A	-8	-4	0	4	5	3	1	-1	-3
5	T	-10	-6	-2	2	6	4	2	0	1
6	T	-12	-8	-4	0	4	5	3	1	2
7	C	-14	-10	-6	-2	2	3	7	5	3
8	A	-16	-12	-8	-4	0	1	5	6	4

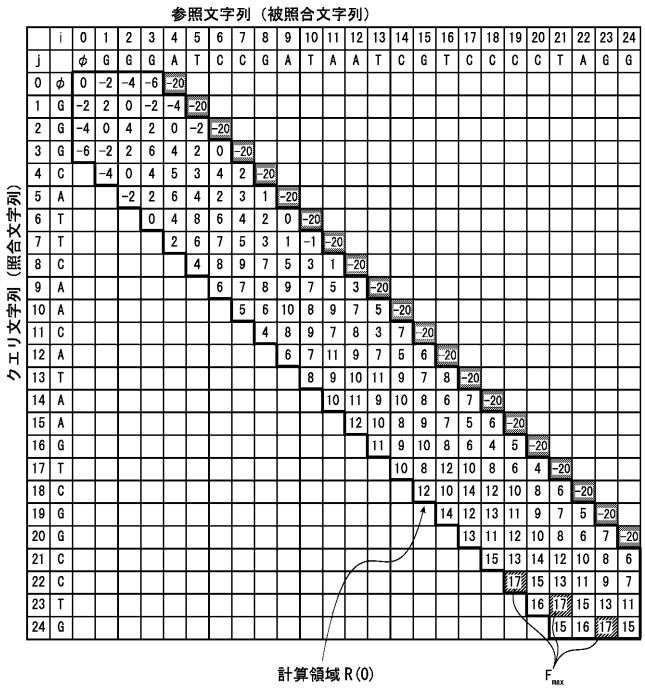
【図９】



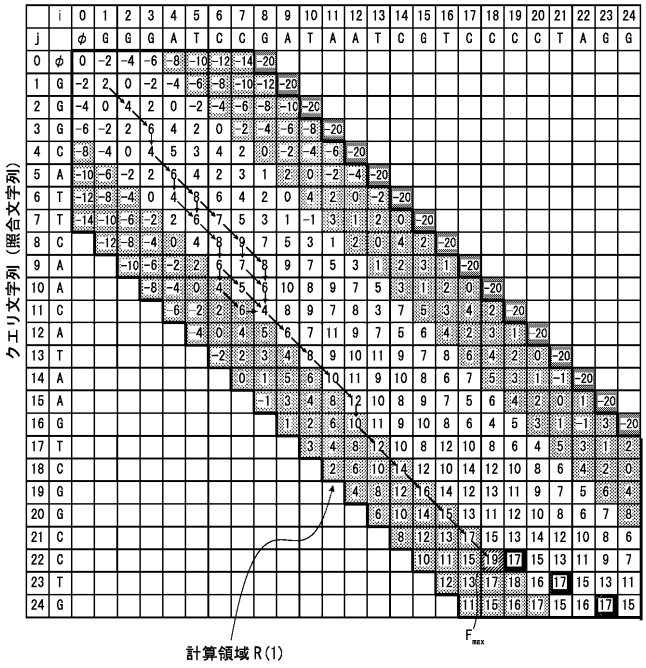
【図１０Ａ】



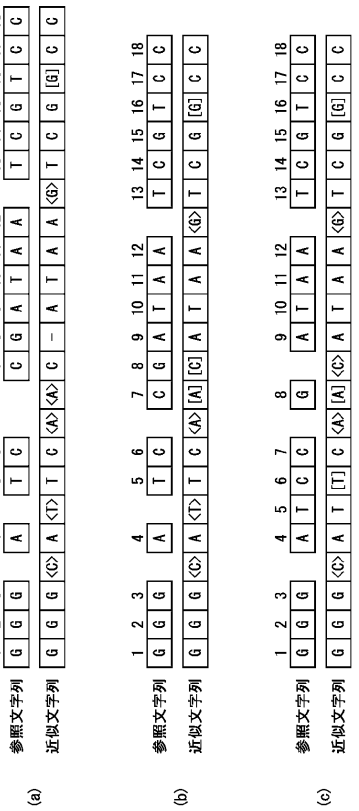
【図１０Ｂ】



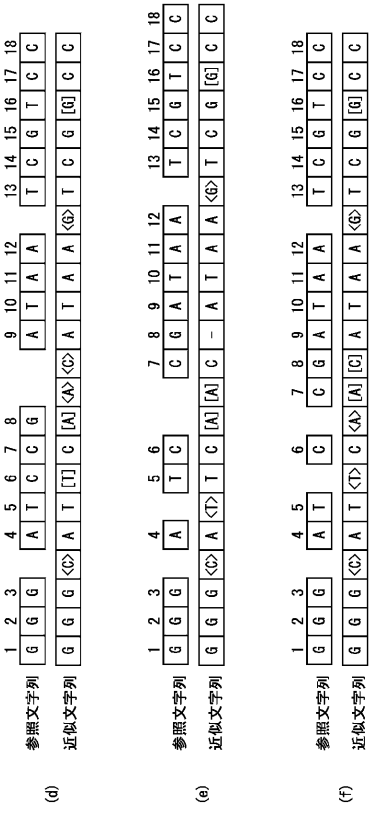
【図１０Ｃ】



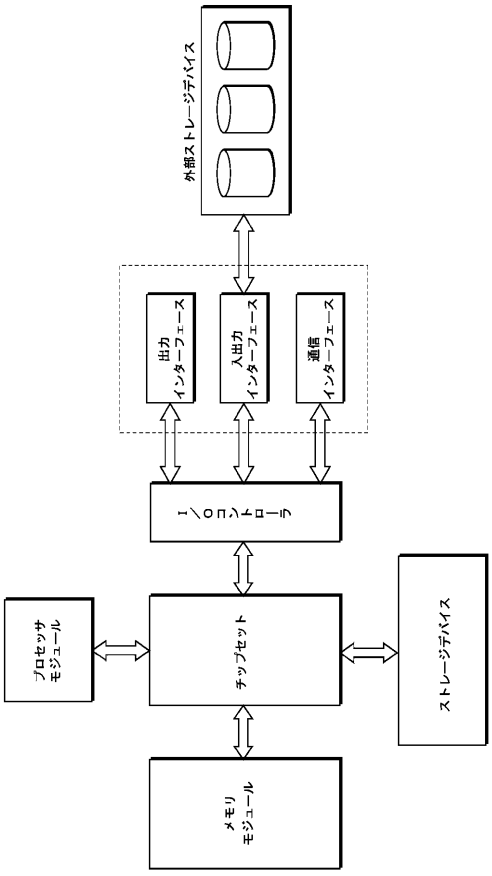
【図１１Ａ】



【図１１Ｂ】



【図１２】



【手続補正書】
【提出日】令和5年1月26日(2023.1.26)
【手続補正１】
【補正対象書類名】特許請求の範囲
【補正対象項目名】全文
【補正方法】変更
【補正の内容】
【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近似文字列を検索するための方法を実行させるためのコンピュータプログラムであって、
前記方法は、
前記参照文字列に基づいて階層的インデックスを作成することと、
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、
前記のマッピングにより同定される前記部分文字列に基づいて作成される、前記参照文字列に基づく被照合文字列と前記クエリ文字列に基づく照合文字列とに基づいて、前記被照合文字列に近似する文字列を前記近似文字列として導出することと、を含み、
前記階層的インデックスを作成することは、
前記参照文字列から所定長の各第１のキーを切り出すことと、
切り出された前記各第１のキーについて、所定のハッシュ関数により該第１のキーに基づいて算出されるハッシュ値を割り当てた第１のキー配列を作成することと、
作成された前記第１のキー配列を更新することと、
更新された前記第１のキー配列を前記階層的インデックスとして出力することと、を含み、
前記第１のキー配列を更新することは、
前記第１のキー配列における前記各第１のキーについて、前記参照文字列における該第１のキーの出現回数を同定することと、
同定された前記第１のキーの前記出現回数に従って、該第１のキーに対して前記参照文字列における該第１のキーに続く少なくとも１以上の文字からなる第１の追加キーを追加することにより前記第１のキー配列を更新することと、を含む、
コンピュータプログラム。
【請求項２】
前記第１のキー配列を作成することは、前記ハッシュ値に従って前記第１のキー配列における前記各第１のキーをソートすることを含む、
請求項１に記載のコンピュータプログラム。
【請求項３】
前記第１のキー配列を更新することは、
前記同定された出現回数が所定の許容値を超えているか否かを判断することと、
前記同定された出現回数が前記所定の許容値を超えていると判断される場合に、前記第１のキーに対して前記参照文字列における該第１のキーに続く少なくとも１以上の文字からなる前記第１の追加キーを追加することにより新たな第１のキーを作成することと、
前記新たな第１のキーについて、前記参照文字列における該新たな第１のキーの出現回数を同定することと、を含む、
請求項１又は２に記載のコンピュータプログラム。
【請求項４】
前記第１のキー配列を更新することは、前記第１の追加キーに従って前記第１のキー配列における前記新たな第１のキーをソートすることを更に含む、
請求項１から３のいずれか一項に記載のコンピュータプログラム。

【請求項５】
前記第１のキー配列を更新することは、前記同定された前記出現回数が所定の許容値を超えていないと判断されるまで、現在の前記第１のキーに新たな前記第１の追加キーを順次に追加することにより新たな前記第１のキーを作成することを含む、
請求項３又は４に記載のコンピュータプログラム。
【請求項６】
前記第１のキー配列を前記階層的インデックスとして出力することは、
前記同定された前記出現回数が所定の許容値を超えていないと判断される場合に、現在の前記第１のキー配列を前記階層的インデックスとして出力することを含む、
請求項３から５のいずれか一項に記載のコンピュータプログラム。
【請求項７】
前記マッピングを行うことは、
前記クエリ文字列から所定長の各第２のキーを切り出すことと、
前記クエリ文字列から切り出された前記各第２のキーについて、前記所定のハッシュ関数により該第２のキーに基づいて算出されるハッシュ値を割り当てた第２のキー配列を作成することと、
前記各第２のキーについて、前記ハッシュ値に従って、所定のサンプリング間隔で、前記階層的インデックスを参照し、該第２のキーの出現開始位置及び出現回数を同定することと、を含む、
請求項１から６のいずれか一項に記載のコンピュータプログラム。
【請求項８】
前記第２のキーの前記出現開始位置及び前記出現回数を同定することは、
前記第２のキーの前記出現回数が前記所定の許容値を超えているか否かを判断することと、
前記第２のキーの前記出現回数が前記所定の許容値を超えていると判断される場合に、前記第２のキーに対して前記クエリ文字列における該第２のキーに続く少なくとも１以上の文字からなる第２の追加キーを追加することにより新たな第２のキーを作成することと、
前記第２のキーの前記出現回数が前記所定の許容値を超えていないと判断される場合に、同定された現在の前記第２のキーを前記部分文字列として出力するとともに該第２のキーの前記出現開始位置を出力することと、を含む、
請求項７に記載のコンピュータプログラム。
【請求項９】
前記第２のキーの前記出現開始位置及び前記出現回数を同定することは、前記第２のキーの前記同定された前記出現回数が前記所定の許容値を超えていないと判断されるまで、現在の前記第２のキーに新たな前記第２の追加キーを順次に追加して、前記新たな第２のキーを作成することを更に含む、
請求項８に記載のコンピュータプログラム。
【請求項１０】
前記第２のキーの前記出現回数が前記所定の許容値を超えていると判断される場合に、該第２のキーの前記所定のサンプリング間隔を大きくする、
請求項８又は９に記載のコンピュータプログラム。
【請求項１１】
前記近似文字列として導出することは、
前記マッピングにより同定された前記部分文字列に基づく、前記被照合文字列と前記照合文字列とからなる文字列ペアを受信することと、
前記文字列ペアに基づいて少なくとも１つの前記近似文字列を導出するために、所定のアラインメント処理を実行することと、
導出された前記少なくとも１つの近似文字列を出力することと、を含む、
請求項８から１０のいずれか一項に記載のコンピュータプログラム。

【請求項１２】
前記所定のアラインメント処理を実行することは、
前記被照合文字列と前記照合文字列とに基づいて所定のアラインメント表を作成することと、
前記アラインメント表の対角線上の要素を中心にした幅ｍを有する計算領域を設定することと、
設定された前記計算領域における各要素について、変異度を算出することと、
算出された前記変異度に基づいて、最大変異度を決定することと、
決定された前記最大変異度に基づいて、前記少なくとも１つの近似文字列を導出することを含む、
請求項１１に記載のコンピュータプログラム。
【請求項１３】
前記所定のアラインメント処理を実行することは、
前記最大変異度と所定の下限値とを比較して、前記最大変異度が前記所定の下限値を超えているかを判断することと、
前記最大変異度が前記所定の下限値を超えていないと判断される場合に、新たな計算領域を設定するために、前記計算領域の前記幅ｍを拡幅することと、
前記最大変異度が前記所定の下限値を超えていると判断される場合に、前記最大変異度を有する要素に基づいて、前記少なくとも１つの近似文字列を導出することと、を含み、
前記最大変異度が前記下限値を超えると判断されるまで、前記計算領域を拡幅することにより新たな計算領域を設定して前記変異度を算出することを繰り返す、
請求項１２に記載のコンピュータプログラム。
【請求項１４】
前記所定の下限値は、所定の要素列にｍ個の連続したギャップがあり、それ以外の部分は一致したと仮定した場合の変異度の値である、
請求項１３に記載のコンピュータプログラム。
【請求項１５】
前記部分文字列に対して前記参照文字列における対応する所定の文字列を追加することにより前記被照合文字列を作成することと、
前記部分文字列に対して前記クエリ文字列における対応する所定の文字列を追加することにより前記照合文字列を作成することと、を更に含む、
請求項１１から１４のいずれか一項に記載のコンピュータプログラム。
【請求項１６】
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を探索するための階層的インデックスを作成する方法を実行させるためのコンピュータプログラムであって、
前記方法は、
前記参照文字列から所定長の各第１のキーを切り出すことと、
切り出された前記各第１のキーについて、所定のハッシュ関数により該第１のキーに基づいて算出されるハッシュ値を割り当てた第１のキー配列を作成することと、
作成された前記第１のキー配列を更新することと、
更新された前記第１のキー配列を前記階層的インデックスとして出力することと、を含み、
前記第１のキー配列を更新することは、
前記第１のキー配列における前記各第１のキーについて、前記参照文字列における該第１のキーの出現開始位置及び出現回数を同定することと、
同定された前記第１のキーの前記出現開始位置及び前記出現回数に従って、該第１のキーに対して前記参照文字列における該第１のキーに続く少なくとも１以上の文字からなる第１の追加キーを追加することにより前記第１のキー配列を更新することと、を含む、
コンピュータプログラム。

【請求項１７】
コンピューティングデバイスに、参照文字列に対してクエリ文字列のマッピングを行う方法を実行させるためのコンピュータプログラムであって、
前記方法は、
前記参照文字列から切り出される各第１のキー、及び前記各第１のキーについて、該第１のキーに基づいて所定のハッシュ関数により算出されるハッシュ値が割り当てられた第１のキー配列であって、前記参照文字列における前記第１のキーの出現開始位置及び出現回数に従って前記第１のキーに追加キーが追加された該第１のキー配列からなる階層的インデックスを読み出すことと、
前記クエリ文字列から所定のキー長を有する各第２のキーを切り出すことと、
前記クエリ文字列から切り出された前記各第２のキーについて、所定のハッシュ関数により該第２のキーに基づいて算出されるハッシュ値を割り当てた第２のキー配列を作成することと、
前記各第２のキーについて、前記ハッシュ値に従って、所定のサンプリング間隔で、前記階層的インデックスを参照し、前記第１のキー配列における前記第１のキーとの比較により、該第２のキーの出現開始位置及び出現回数を同定することと、
前記同定した出現回数が所定のしきい値を超えているか否かを判断することと、
前記同定された前記出現回数が所定の許容値を超えていると判断される場合に、前記第２のキーに対して前記クエリ文字列における該第２のキーに続く少なくとも１以上の文字からなる追加キーを追加することにより新たな第２のキーを作成することと、
前記同定された前記出現回数が所定のしきい値を超えていないと判断される場合に、同定された現在の前記第２のキーの出現開始位置及び該キーを出力することと、を含み、
前記第２のキーの前記出現開始位置及び前記出現回数を同定することは、前記同定された前記出現回数が所定のしきい値を超えていないと判断されるまで、現在の前記第２のキーに新たな前記追加キーを順次に追加して、前記新たな第２のキーを作成することを含む、
コンピュータプログラム。
【請求項１８】
前記方法は、前記同定された前記出現回数が所定の許容値を超えていると判断される場合に、前記第２のキーの前記所定のサンプリング間隔を大きくするように構成される、請求項１７に記載のコンピュータプログラム。
【請求項１９】
コンピューティングデバイスに、参照文字列における部分文字列とクエリ文字列との間の変異を所定のアラインメント処理により同定する方法を実行させるためのコンピュータプログラムであって、
前記方法は、
マッピングにより同定される前記部分文字列に基づいて作成される、前記参照文字列に基づく被照合文字列と前記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を受信することと、
前記文字列ペアに基づいて、前記被照合文字列に近似する文字列を少なくとも１つの近似文字列として導出するために、所定のアラインメント処理を実行することと、
導出された前記少なくとも１つの近似文字列を出力することと、を含み、
前記所定のアラインメント処理を実行することは、
前記被照合文字列と前記照合文字列とに基づいて所定のアラインメント表を作成することと、
前記アラインメント表の対角線上の要素を中心にした幅ｍを有する計算領域を設定することと、
設定された前記計算領域における各要素について、変異度を算出することと、
算出された前記変異度に基づいて、最大変異度を決定することと、
決定された前記最大変異度に基づいて、前記少なくとも１つの近似文字列を導出するこ

とを含む、
コンピュータプログラム。
【請求項２０】
前記所定のアラインメント処理を実行することは、
前記最大変異度と所定の下限値とを比較して、前記最大変異度が前記所定の下限値を超えているかを判断することと、
前記最大変異度が前記所定の下限値を超えていないと判断される場合に、新たな計算領域を設定するために、前記計算領域の前記幅ｍを拡幅することと、
前記最大変異度が前記所定の下限値を超えていると判断される場合に、前記最大変異度を有する要素に基づいて、前記少なくとも１つの近似文字列を導出することと、を含み、
前記最大変異度が前記下限値を超えると判断されるまで、前記計算領域を拡幅することにより新たな計算領域を設定して前記変異度を算出することを繰り返す、
請求項１９に記載のコンピュータプログラム。
【請求項２１】
前記所定のアラインメント処理を実行することは、所定の要素列にｍ個の連続したギャップがあり、それ以外の部分は一致したと仮定した場合の変異度の値を前記所定の下限値として設定することを更に含む、
請求項２０に記載のコンピュータプログラム。

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
(72)発明者　姫野　龍太郎
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　DA10　FA01　HB01　KA04　KA08