(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7631330号
(P7631330)

(45)発行日　令和7年2月18日(2025.2.18)

(24)登録日　令和7年2月7日(2025.2.7)

(51)国際特許分類

Ｇ１６Ｂ　50/50　(2019.01)	Ｇ１６Ｂ	50/50		
Ｇ０６Ｆ　12/00　(2006.01)	Ｇ０６Ｆ	12/00		
Ｇ０６Ｆ　16/17　(2019.01)	Ｇ０６Ｆ	16/17	１００	
Ｇ０６Ｆ　16/22　(2019.01)	Ｇ０６Ｆ	16/22		
Ｈ０３Ｍ　7/30　(2006.01)	Ｈ０３Ｍ	7/30		Ｚ

ＦＩ

請求項の数　20　（全42頁）

(21)出願番号　特願2022-522858(P2022-522858)
(86)(22)出願日　令和2年10月17日(2020.10.17)
(65)公表番号　特表2022-553199(P2022-553199A)
(43)公表日　令和4年12月22日(2022.12.22)
(86)国際出願番号　PCT/EP2020/079298
(87)国際公開番号　WO2021/074440
(87)国際公開日　令和3年4月22日(2021.4.22)
審査請求日　令和5年10月12日(2023.10.12)
(31)優先権主張番号　62/923,141
(32)優先日　令和1年10月18日(2019.10.18)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　62/956,952
(32)優先日　令和2年1月3日(2020.1.3)
最終頁に続く

(73)特許権者　590000248
コーニンクレッカ フィリップス エヌヴェ
Ｋｏｎｉｎｋｌｉｊｋｅ Ｐｈｉｌｉｐｓ Ｎ．Ｖ．
オランダ国 ５６５６ アーヘー アインドーフェン ハイテック キャンパス ５２
Ｈｉｇｈ Ｔｅｃｈ Ｃａｍｐｕｓ ５２，５６５６ ＡＧ Ｅｉｎｄｈｏｖｅｎ，Ｎｅｔｈｅｒｌａｎｄｓ
(74)代理人　110001690
弁理士法人Ｍ＆Ｓパートナーズ
(72)発明者　チャンダク シュブハム
オランダ国 ５６５６ アーエー アインドーフェン ハイ テック キャンパス ５
最終頁に続く

(54)【発明の名称】　多様な表形式データの効果的な圧縮、表現、および展開のためのシステムおよび方法

(57)【特許請求の範囲】

【請求項１】

データの圧縮を制御するための方法であって、前記方法は、

複数の第１のファイル形式のうちの１つの第１のファイル形式の表形式データにアクセスするステップと、

前記表形式データから複数の属性を抽出するステップと、

前記表形式データを複数のチャンクに分割するステップと、

前記抽出された属性および前記チャンクを処理して相関付けられた情報にするステップと、

前記相関付けられた情報内で識別される前記属性および前記チャンクに対して、異なるコンプレッサおよび依存関係属性を選択するステップと、

チャンク間で共有されるコンプレッサのためのグローバルデータを保存するステップと、

前記相関付けられた情報と、前記相関付けられた情報内に示される前記チャンクおよび前記属性のための前記異なるコンプレッサを示す情報とを含む第２のファイル形式のファイルを生成するステップとを含み、

前記複数の第１のファイル形式は、互いに互換性がなく、前記第２のファイル形式とは異なり、前記異なるコンプレッサを示す前記情報は、前記相関付けられた情報内に示される前記属性および前記チャンクの選択的展開を可能にするために、前記第２のファイル形式に処理される、方法。

【請求項２】

前記相関付けられた情報は、
前記属性のうちの１つまたは複数の属性を示す第１の情報、および
前記第１の情報内に示される前記１つまたは複数の属性に関連付けられたチャンクに対応する第２の情報を含む、
少なくとも１つのテーブルを含む、請求項１に記載の方法。
【請求項３】
前記第１の情報は二次元セル配列を含み、各セルは、前記二次元セル配列に対応する前記チャンク内に含まれる１つまたは複数の対応する属性を特定する、請求項２に記載の方法。
【請求項４】
前記第１の情報は、前記チャンクに関連する次元固有属性の少なくとも１つの一次元テーブルを含む、請求項３に記載の方法。
【請求項５】
前記異なるコンプレッサのうちの少なくとも１つのコンプレッサが前記ファイルに埋め込まれている、請求項１に記載の方法。
【請求項６】
前記異なるコンプレッサのうちの少なくとも１つは、前記表形式データの１つまたは複数の属性をスパース配列として符号化する、請求項１に記載の方法。
【請求項７】
前記表形式データは異なるサイズのチャンクに分割される、請求項１に記載の方法。
【請求項８】
前記方法はさらに、
１つまたは複数の属性固有インデックス生成するステップと、
前記１つまたは複数の属性固有インデックスを前記第２のファイル形式の前記ファイルに組み込むステップとを含み、前記属性固有インデックスは、前記表形式データの特定の属性の値または範囲を含むチャンクを識別することを可能にする、請求項１に記載の方法。
【請求項９】
前記方法はさらに、
前記第２のファイル形式の前記ファイルをＭＰＥＧ－Ｇファイルに統合するステップを含む、請求項１に記載の方法。
【請求項１０】
前記方法はさらに、
前記相関付けられた情報のためのアクセス制御ポリシー情報を生成するステップと、
前記アクセス制御ポリシー情報を前記第２のファイル形式の前記ファイルに統合するステップとを含み、
前記アクセス制御ポリシー情報は、前記相関付けられた情報の第１の部分のための第１のアクセスレベルを示す第１の情報、および前記相関付けられた情報の第２の部分のための第２のアクセスレベルを示す第２の情報を含み、前記第２のアクセスレベルは前記第１のアクセスレベルとは異なる、請求項１に記載の方法。
【請求項１１】
データの圧縮を制御するための装置であって、前記装置は、
命令を保存するように構成されたメモリと、
前記命令を実行して動作を実行するように構成された少なくとも１つのプロセッサとを備え、前記動作は、
複数の第１のファイル形式のうちの１つの第１のファイル形式の表形式データにアクセスするステップと、
前記表形式データから複数の属性を抽出するステップと、
前記表形式データを複数のチャンクに分割するステップと、
前記抽出された属性および前記チャンクを処理して相関付けられた情報にするステップと、

前記相関付けられた情報内で識別される前記属性および前記チャンクに対して、異なるコンプレッサおよび依存関係属性を選択するステップと、
チャンク間で共有されるコンプレッサのためのグローバルデータを保存するステップと、
前記相関付けられた情報と、前記相関付けられた情報内に示される前記チャンクおよび前記属性のための前記異なるコンプレッサを示す情報とを含む第２のファイル形式のファイルを生成するステップとを含み、
前記複数の第１のファイル形式は、互いに互換性がなく、前記第２のファイル形式とは異なり、前記異なるコンプレッサを示す前記情報は、前記相関付けられた情報内に示される前記属性および前記チャンクの選択的展開を可能にするために、前記第２のファイル形式に処理される、装置。

【請求項１２】
前記相関付けられた情報は、
前記属性のうちの１つまたは複数の属性を示す第１の情報、および
前記第１の情報内に示される前記１つまたは複数の属性に関連付けられたチャンクに対応する第２の情報を含む少なくとも１つのテーブルを含む、請求項１１に記載の装置。

【請求項１３】
前記第１の情報は二次元セル配列を含み、各セルは、前記二次元セル配列に対応する前記チャンク内に含まれる１つまたは複数の対応する属性を特定する、請求項１２に記載の装置。

【請求項１４】
前記第１の情報は、前記チャンクに関連する次元固有属性の少なくとも１つの一次元テーブルを含む、請求項１３に記載の装置。

【請求項１５】
前記異なるコンプレッサのうちの少なくとも１つのコンプレッサが前記ファイルに埋め込まれている、請求項１に記載の方法。

【請求項１６】
前記異なるコンプレッサのうちの少なくとも１つは、前記表形式データの１つまたは複数の属性をスパース配列として符号化する、請求項１に記載の方法。

【請求項１７】
前記表形式データは異なるサイズのチャンクに分割される、請求項１１に記載の装置。

【請求項１８】
前記少なくとも１つのプロセッサは、前記命令を実行して、
１つまたは複数の属性固有インデックス生成し、
前記１つまたは複数の属性固有インデックスを前記第２のファイル形式の前記ファイルに組み込むように構成され、前記属性固有インデックスは、前記表形式データの特定の属性の値または範囲を含むチャンクを識別することを可能にする、請求項１１に記載の装置。

【請求項１９】
前記少なくとも１つのプロセッサは、前記命令を実行して、前記第２のファイル形式の前記ファイルをＭＰＥＧ－Ｇファイルに統合するように構成される、請求項１１に記載の装置。

【請求項２０】
前記少なくとも１つのプロセッサは、前記命令を実行して、
前記相関付けられた情報のためのアクセス制御ポリシー情報を生成し、
前記アクセス制御ポリシー情報を前記第２のファイル形式の前記ファイルに統合するように構成され、
前記アクセス制御ポリシー情報は、前記相関付けられた情報の第１の部分のための第１のアクセスレベルを示す第１の情報、および前記相関付けられた情報の第２の部分のための第２のアクセスレベルを示す第２の情報を含み、前記第２のアクセスレベルは前記第１のアクセスレベルとは異なる、請求項１１に記載の装置。

【発明の詳細な説明】

【関連出願の相互参照】
【０００１】
［０００１］本出願は、２０１９年１０月１８日に出願された米国仮特許出願第６２／９２３，１４１号に関連し、その全内容があらゆる目的のために参照により本明細書に援用される。
【０００２】
［０００２］本出願は、２０１９年１０月１８日に出願された米国仮特許出願第６２／９２３，１１３号に関連し、その全内容があらゆる目的のために参照により本明細書に援用される。
【０００３】
［０００３］本出願は、本出願と同時に出願された米国仮特許出願第６２／９５６，９４１号（代理人整理番号２０１９Ｐ００８３１ＵＳ０１）、“Ｃｕｓｔｏｍｉｚａｂｌｅ Ｄｅｌｉｍｉｔｅｄ Ｔｅｘｔ Ｃｏｍｐｒｅｓｓｉｏｎ Ｆｒａｍｅｗｏｒｋ”に関連し、その全内容があらゆる目的のために参照により本明細書に援用される。
【技術分野】
【０００４】
［０００４］本明細書の１つまたは複数の実施形態は、これらに限定はされないが、表形式データおよび区切られたテキストファイルなどの情報の圧縮および展開を管理することに関する。
【背景技術】
【０００５】
［０００５］ゲノムデータは多くの研究用途にとって重要であることが知られている。ＲＡＷ形式のゲノムデータは、多数の配列決定されたゲノムを含む可能性がある。場合によっては、管理できないほどに配列の数が多い。したがって、ＲＡＷデータから関連性のある情報を抽出する、またはより適切に解釈するために様々な試みがなされてきた。
【０００６】
［０００６］１つの処理技術はアノテーションツールの使用を伴う。そのようなツールは、ゲノムデータ内のコーディング領域、およびそれらの対応する位置を定めるために使用され得る。アノテーションはまた、疾患や遺伝的異常に関する情報を伝達する可能性のある反復領域の数および空間分布の指標を提供し得る。ゲノムアノテーションデータの例として、マッピング統計量、定量的ブラウザトラック、バリアント、ゲノム機能的アノテーション、遺伝子発現データ、およびＨｉ－Ｃコンタクト行列などが挙げられる。
【０００７】
［０００７］アノテーションデータ（およびその他のゲノム関連情報）は、現在、様々なファイル形式、例えば、バリアントコール形式（ＶＣＦ）、ｂｒｏｗｓｅｒ ｅｘｔｅｎｓｉｂｌｅ ｄａｔａ（ＢＥＤ）形式、およびｗｉｇｇｌｅ（ＷＩＧ）形式などで表されている。これらの形式は互いに互換性を有さない。結果として、これらの形式の使用は、相互運用性に関連する問題を伴い、データを視覚化できるようにするために頻繁に形式を変換する必要がある。また、単一の統一および標準化された形式がないため、圧縮アルゴリズムへの取り組みが妨げられ、次善の圧縮アルゴリズム（例えば、ｇｚｉｐ）が広く使用されることとなった。
【０００８】
［０００８］ゲノム情報を圧縮するための既存のアルゴリズムに関連する欠点は、とりわけ、選択性の欠如に関連する。例えば、ゲノムアノテーションデータは通常、異なる統計的特性を有する複数のフィールド（属性）を含む。既存のアルゴリズムは、すべてのフィールドをまとめて圧縮するため、個別のデータフィールド内の情報を選択的に認識、使用、および抽出することができない。また、既存のアルゴリズムでは、すべての属性を展開せずに特定のフィールドを抽出することはできない。
【０００９】
［０００９］その他の欠点は、既存の圧縮方法が一般的に適用できないことに関連する

。例えば、これらの方法は１つの標準コンプレッサ、または少数の標準コンプレッサのセットのみに依存するため、１つのタイプのアノテーションデータにしか適用できない。これらの方法は、選択的暗号化を実行することもできず、また、複数のアノテーションデータセットを相互に、およびシーケンシングデータとリンクさせることもできない。いくつかの特殊化された方法が提案されている。しかし、これらの方法の多くは、ディスクベースの配列（アレイ）管理ツール（例えば、ＴｉｌｅＤＢおよびＨＤＦ５）に基づいており、これらのツールは、高レベルのフィーチャ、例えば、限定はされないが、メタデータ、リンケージ、および属性固有インデックス付与などを有さない。

【発明の概要】

【００１０】

［００１０］様々な例示的実施形態の簡単な要約を以下に提示する。以下の要約において、いくつかの簡略化および省略がなされ得るが、これは様々な例示的実施形態のいくつかの態様を強調および紹介することを意図したものであり、本発明の範囲を限定することは意図されていない。当業者が実施形態を作成および使用することを可能にするのに十分な例示的実施形態の詳細な説明が後になされる。

【００１１】

［００１１］１つまたは複数の実施形態によれば、データの圧縮を制御するための方法は、複数の第１のファイル形式のうちの１つの第１のファイル形式のゲノムアノテーションデータにアクセスするステップと、ゲノムアノテーションデータから複数の属性を抽出するステップと、ゲノムアノテーションデータを複数のチャンクに分割するステップと、抽出された属性およびチャンクを処理して相関付けられた情報にするステップと、相関付けられた情報内で識別される属性およびチャンクに対して、異なるコンプレッサを選択するステップと、相関付けられた情報と、相関付けられた情報内に示されるチャンクおよび属性のための異なるコンプレッサを示す情報とを含む第２のファイル形式のファイルを生成するステップとを含み、複数の第１のファイル形式は互いに互換性がなく、第２のファイル形式とは異なり、異なるコンプレッサを示す情報は、相関付けられた情報内に示される属性およびチャンクの選択的展開を可能にするために、第２のファイル形式に処理される。

【００１２】

［００１２］相関付けられた情報は、属性のうちの１つまたは複数の属性を示す第１の情報、および第１の情報内に示される１つまたは複数の属性に関連付けられたチャンクに対応する第２の情報を含む少なくとも１つのテーブルを含み得る。第１の情報は二次元セル配列を含み、各セルは、二次元セル配列に対応するチャンク内に含まれる１つまたは複数の対応する属性を特定し得る。第１の情報は、チャンクに関連する次元固有属性の少なくとも１つの一次元テーブルを含み得る。ゲノムアノテーションデータは同じサイズのチャンクに分割されてもよい。ゲノムアノテーションデータは異なるサイズのチャンクに分割されてもよい。

【００１３】

［００１３］方法は、相関付けられた情報をリンクする１つまたは複数の第１のインデックスを生成するステップと、相関付けられた情報を、異なるコンプレッサを示す情報にリンクする１つまたは複数の第２のインデックスを生成するステップとを含み、１つまたは複数の第１のインデックスおよび１つまたは複数の第２のインデックスは、第２のファイル形式のファイルに処理される。方法は、第２のファイル形式のファイルをＭＰＥＧ－Ｇファイルに統合するステップを含み得る。

【００１４】

［００１４］方法は、相関付けられた情報のためのアクセス制御ポリシー情報を生成するステップと、アクセス制御ポリシー情報を第２のファイル形式のファイルに統合するステップとを含み、アクセス制御ポリシー情報は、相関付けられた情報の第１の部分のための第１のアクセスレベルを示す第１の情報、および相関付けられた情報の第２の部分のための第２のアクセスレベルを示す第２の情報を含み、第２のアクセスレベルは第１のアク

セスレベルとは異なる。抽出された属性は、染色体属性およびゲノム位置属性のうちの少なくとも１つを含み得る。

【００１５】

［００１５］１つまたは複数の実施形態によれば、データの圧縮を制御するための装置は、命令を保存するように構成されたメモリと、命令を実行して動作を実行するように構成された少なくとも１つのプロセッサとを備え、動作は、複数の第１のファイル形式のうちの１つの第１のファイル形式のゲノムアノテーションデータにアクセスするステップと、ゲノムアノテーションデータから属性を抽出するステップと、ゲノムアノテーションデータをチャンクに分割するステップと、抽出された属性およびチャンクを処理して相関付けられた情報にするステップと、相関付けられた情報内で識別される属性およびチャンクに対して、異なるコンプレッサを選択するステップと、相関付けられた情報と、相関付けられた情報内に示されるチャンクおよび属性のための異なるコンプレッサを示す情報とを含む第２のファイル形式のファイルを生成するステップとを含む。複数の第１のファイル形式は互いに互換性がなく、第２のファイル形式とは異なる。異なるコンプレッサを示す情報は第２のファイル形式に処理され、相関付けられた情報内に示される属性およびチャンクの選択的展開を可能にする。

【００１６】

［００１６］相関付けられた情報は、属性のうちの１つまたは複数の属性を示す第１の情報、および第１の情報内に示される１つまたは複数の属性に関連付けられたチャンクに対応する第２の情報を含む少なくとも１つのテーブルを含み得る。第１の情報は二次元セル配列を含み、各セルは、二次元セル配列に対応するチャンク内に含まれる１つまたは複数の対応する属性を特定する。第１の情報は、チャンクに関連する次元固有属性の少なくとも１つの一次元テーブルを含み得る。ゲノムアノテーションデータは同じサイズのチャンクに分割されてもよい。ゲノムアノテーションデータは異なるサイズのチャンクに分割されてもよい。

【００１７】

［００１７］少なくとも１つのプロセッサは、命令を実行して、相関付けられた情報をリンクする１つまたは複数の第１のインデックスを生成し、相関付けられた情報を、異なるコンプレッサを示す情報にリンクする１つまたは複数の第２のインデックスを生成するように構成され、１つまたは複数の第１のインデックスおよび１つまたは複数の第２のインデックスは、第２のファイル形式のファイルに処理される。少なくとも１つのプロセッサは、命令を実行して、第２のファイル形式のファイルをＭＰＥＧ－Ｇファイルに統合するように構成される。

【００１８】

［００１８］少なくとも１つのプロセッサは、命令を実行して、相関付けられた情報のためのアクセス制御ポリシー情報を生成し、アクセス制御ポリシー情報を第２のファイル形式のファイルに統合するように構成され、アクセス制御ポリシー情報は、相関付けられた情報の第１の部分のための第１のアクセスレベルを示す第１の情報、および相関付けられた情報の第２の部分のための第２のアクセスレベルを示す第２の情報を含み、第２のアクセスレベルは第１のアクセスレベルとは異なる。抽出された属性は、染色体属性およびゲノム位置属性のうちの少なくとも１つを含み得る。

【図面の簡単な説明】

【００１９】

［００１９］添付図面では、類似する参照番号は、個々の図にわたって、同一のまたは機能的に類似する要素を指す。添付図面は以下の詳細な説明とともに明細書に組み込まれ、その一部を形成する。添付図面は、特許請求の範囲に示される概念の例示的実施形態をさらに説明する役割を果たし、それらの実施形態の様々な原理および利点を説明する。

【００２０】

［００２０］上記および他のより詳細かつ具体的な特徴が、添付図面を参照しつつ、以下の明細書においてより完全に開示されている。

【図１】［００２１］図１は、ゲノムアノテーションデータの圧縮および展開を制御するためのシステムの実施形態を示す。
【図２】［００２２］図２は、ゲノムアノテーションデータの圧縮および展開を制御するための方法を示す。
【図３】［００２３］図３は、ゲノムアノテーションデータのための統一および標準化されたファイル形式の実施形態を示す。
【図４】［００２４］図４は、ゲノムアノテーションデータのためのアノテーションファイルに情報を処理するための実施形態を示す。
【図５】［００２５］図５は、統合および標準化されたファイルのファイルヘッダに情報を処理するための実施形態を示す。
【図６】［００２６］図６は、統一および標準化されたファイルのために情報を処理するための実施形態を示す。
【図７】［００２７］図７は、統合および標準化されたファイルのために圧縮情報を処理するための実施形態を示す。
【図８】［００２８］図８は、一次元配列のために情報を処理するための実施形態を示す。
【図９】［００２９］図９は、二次元配列に情報を処理するための実施形態を、次元固有属性のための配列とともに示す。
【図１０】［００３０］図１０は、ゲノムアノテーションデータのためにインデックス付与される属性情報を処理するための実施形態を示す。
【図１１】［００３１］図１１は、追加の属性関連情報を処理するための実施形態を示す。
【図１２】［００３２］図１２は、属性情報を処理するための実施形態を示す。
【図１３】［００３３］図１３は、データのチャンクを選択的に圧縮するための実施形態を示す。
【図１４】［００３４］図１４は、圧縮ファイルインデックス情報を処理するための実施形態を示す。
【図１５】［００３５］図１５は、インデックス情報を処理するための実施形態を示す。
【図１６】［００３６］図１６は、チャンクを処理するための実施形態を示す。
【図１７】［００３７］図１７は、固定サイズのチャンクを生成するための実施形態を示す。
【図１８】［００３８］図１８は、可変サイズのチャンクを生成するための実施形態を示す。
【図１９】［００３９］図１９は、チャンクおよび属性情報を処理するための実施形態を示す。
【図２０】［００４０］図２０は、チャンクおよび属性情報を処理するための実施形態を示す。
【図２１】［００４１］図２１は、データペイロードを処理するための実施形態を示す。
【図２２】［００４２］図２２は、統一および標準化された形式のゲノムアノテーションデータを選択的に展開するための実施形態を示す。
【図２３Ａ－２３Ｂ】［００４３］図２３Ａおよび図２３Ｂは、１つまたは複数の実施形態に係る、ファイル形式をフォルダ階層に変換するための方法を示す。

【発明を実施するための形態】

【００２１】

［００４４］図面は単に概略的なものであり、縮尺通りに描かれていないことを理解されたい。また、各図面を通して、同じまたは類似の部分を示すために同じ参照番号が使用されることを理解されたい。

【００２２】

［００４５］記載および図面は、様々な例示的実施形態の原理を示す。したがって、当業者は、本明細書に明示的に記載または図示されていないものの、本発明の原理を体現し、本発明の範囲内に含まれる様々な構成を考え出すことができることが理解されよう。さらに、本明細書に記載のすべての例は、主に、本発明の原理、および発明者らが技術の発

展のために提供する概念を読者が理解するのを助けるという教授的な目的のためのものであることを明確に意図しており、そのような具体的に述べられた例および条件に限定されないものとして解釈されるべきである。また、本明細書で使用される「または」や「もしくは」という用語は、特に明記しない限り（例えば、「さもなければ」または「または代わりに」）、非排他的選言（すなわち、および／または）を指す。また、様々な例示的な実施形態は、必ずしも相互に排他的ではなく、一部の例示的実施形態を１つまたは複数の他の例示的実施形態と組み合わせて、新しい例示的実施形態が形成され得る。「第１」、「第２」、「第３」などの記述語は、議論される要素の順序を限定することを意図したものではなく、ある要素を次の要素と区別するために使用され、一般的に交換可能である。最大値または最小値などの値は事前に決定されてもよく、用途に基づいて異なる値に設定されてもよい。

【００２３】

［００４６］例示的実施形態は、情報、例えば、これらに限定されないが、表形式データおよび区切られたテキストファイルなどの圧縮を管理するためのシステムおよび方法を説明する。圧縮される情報はゲノム情報（例えば、アノテーション、シーケンシング、または他の形態のゲノム関連データ）を含み得る。他の実施形態では、ゲノム情報に関係のない別のタイプの情報が圧縮のために管理されてもよい。

【００２４】

［００４７］ゲノム用途では、システムおよび方法は、ゲノムデータの圧縮および展開の標準化を管理し得る。これは、相互運用性に関連する問題、およびデータを視覚化できるようにするために頻繁に形式を変換する必要性を低減または排除する形式に準拠するようにデータを処理することを含む。少なくとも一部の用途では、これらのフィーチャにより、特定のタイプのデータのための圧縮アルゴリズムの改善および最適化が達成され得る。

【００２５】

［００４８］１つまたは複数の実施形態によれば、システムおよび方法は、データを処理して、データの異なるフィールド、属性、および／またはセクションの選択的圧縮を可能にし得る。例えば、一部の実装形態では、データのフィールド、属性、またはセクションはすべて（例えば、異なる統計的特性を有するものを含む）、すべてが同じ圧縮アルゴリズムを使用して圧縮されない可能性がある。そうではなく、システムおよび方法は、データの１つまたは複数のセクションを選択的に圧縮し、他のセクションは圧縮しないように実装されてもよく、または、異なる圧縮技術を使用して異なるセクションを選択的に圧縮するように実装されてもよい。選択される圧縮技術は、例えば、複数のセクションのうちの対応するそれぞれのセクション内の異なるタイプのデータにとって有益または最適であり得る。

【００２６】

［００４９］この選択的圧縮手法は、ひいては、選択的展開を可能にし得る。例えば、データを選択的に圧縮することにより、データファイル全体を展開する必要がない。そうではなく、少なくとも１つの実装形態において、関心のあるデータの１つまたは複数のセクションを、関心のない圧縮されたデータの他のセクションを展開することなく、選択的に展開することができる。これにより効率が向上し、研究中の負担やその他の遅延を回避する能力が向上する。

【００２７】

［００５０］システムおよび方法によって実行される処理はまた、少なくとも１つの実施形態が、例えば、異なるデータセクションに選択的に適用される多数の標準またはカスタマイズされた圧縮アルゴリズムを使用して、すべてのタイプのデータに一般的に適用可能となることを可能にし得る。システムおよび方法はまた、（例えば、異なる方法で）圧縮されたデータセクションの選択的暗号化を実行し得る。システムおよび方法はまた、複数のアノテーションデータセットを互いに、およびシーケンシングデータとリンクするように実装されてもよい。システムおよび方法はまた、メタデータ、リンケージ、および／または属性固有インデックス付与を含むようにデータを処理し得る。

【００２８】
［００５１］図１は、情報の圧縮を管理するためのシステムの実施形態を示す。システムは、少なくとも部分的に、例えば、実験室のワークステーション、研究施設、データセンター、大学もしくは企業環境、または、意図される用途に関連してデータが処理される別の場所に実装され得る。別の実施形態では、システムは、１つのコンピュータおよび／もしくは他のデバイス、または、本明細書に記載の処理動作を実行するために互いに通信する複数の接続もしくはネットワーク化されたコンピュータおよび／もしくは他のデバイス上に実装され得る。
【００２９】
［００５２］図１を参照して、システムは、プロセッサ１１０、メモリ１２０、および記憶領域１３０を含む。プロセッサ１１０は、情報の圧縮を管理することを含む動作を実行するためにメモリ１２０内に保存された命令を実行する。メモリ１２０は、本明細書に記載の動作を実行するようにプロセッサを制御するための命令を保存する、任意の形式の非一時的コンピュータ可読媒体であり得る。情報はデータソース１４０から受け取られてもよい。データソースは、例えば、処理される情報のタイプによって異なり得る。例えば、一実装形態では、データソース１４０は、ゲノムアノテーションデータを生成するためのソフトウェアベースのツールであり得る。別の実装形態では、データソースは、別のタイプのゲノム関連データ、または、場合によってはゲノムデータに関連しないデータを提供してもよい。
【００３０】
［００５３］プロセッサ１１０はデータソースにローカルに接続されてもよく、またはネットワークを介してデータソースに接続されてもよい。後者の場合、プロセッサとデータソースとの間にネットワーク接続を確立することができれば、システムが処理する情報を世界中のどこからでも取得できる。ネットワーク接続は、ローカル接続、インターネット接続、仮想プライベートネットワーク接続、クラウドコンピューティングもしくはクラウドベースのストレージデバイスへの接続、および／または別のタイプの接続であり得る。
【００３１】
［００５４］記憶領域１３０は、プロセッサ１１０の処理された結果を保存し得る。後により詳細に説明されるように、処理された結果は、複数のフィールド、属性、セクション、または、選択的に圧縮および／もしくはその他の形で処理され得る他の部分を含む、統一および標準化されたファイル形式に準拠する表形式データまたは区切られたテキストファイルを含む可能性がある。記憶領域は、データベース、アーカイブ、ストレージエリアネットワーク、クラウドベースのストレージシステム、プロセッサのコンピュータ、システム、もしくはデバイスに含まれるもしくは結合されたメモリ、または別のタイプの記憶領域であるか、またはこれらに含まれ得る。
【００３２】
［００５５］図１のシステムは、統一および標準化されたファイル形式に処理された情報を圧縮するための１つまたは複数の追加フィーチャを含むか、または追加フィーチャに結合され得る。これらのフィーチャは、パーサ１５０、圧縮マネージャ１６０、およびアグリゲータ１７０を含み得る。
【００３３】
［００５６］パーサ１５０は、保存され、後に記憶領域１３０から取り出されたファイルを解析してもよい。例えば、プロセッサ１１０は、ユーザ入力に基づいて、または別のデバイスからデータのリクエストを受信し得る。プロセッサは、例えば、要求されたデータに対応する記憶領域１３０内に保存された１つまたは複数のファイルを決定し、決定された１つまたは複数のファイルを取り出してもよい。プロセッサ１１０は、例えば、ファイルのコンテンツをリクエストで指定された１つまたは複数の識別子にリンクするインデックスを使用して、どのファイルが要求されたデータに対応するかを決定してもよい。ファイルが見つかると、ファイルは記憶領域１３０から取り出されてパーサに送られる。
【００３４】

［００５７］パーサは様々な形でファイルを解析することができる。例えば、ファイルは、ファイル内のデータ（または他のタイプのコンテンツ）の異なる部分を含む異なるセクション、部分、属性、フィールドなどを区切る区切り文字を含み得る。これらの区切り文字をガイドとして使用して、パーサ１５０は、ファイルを、リクエストに対応する個別のセクション（またはチャンク）に解析することができる。一実施形態では、パーサは、圧縮のためにファイルがどのように解析されるかを示す所定の圧縮スキーマに従ってこれらの動作を実行してもよい。

【００３５】

［００５８］圧縮マネージャ１６０は、ファイルが記憶領域１３０から取り出されると、統一された標準形式でファイルを受け取る。圧縮マネージャは複数のコンプレッサ１６１$_1$、１６１$_2$、・・・、１６１$_N$を含み、ここで、Ｎ≧２である。各コンプレッサは、解析されたファイルの異なるセクションを圧縮するために、異なる圧縮アルゴリズムを実行してもよい。各コンプレッサによって適用される圧縮アルゴリズムは、ファイルの解析されたデータまたはコンテンツに基づいて事前に決定されていてもよい。例えば、効率の観点から、または他の理由で、特定のタイプの情報を圧縮するのに、ある圧縮アルゴリズムがより適している可能性がある。他のタイプの情報を圧縮するには他の圧縮アルゴリズムの方が適している可能性がある。圧縮スキーマは、ファイル内の特定のデータまたはコンテンツを圧縮するためにどの圧縮アルゴリズムを使用するかを制御し得る。（また、パーサが、圧縮スキーマから導出された命令に基づいて、ファイルの異なる解析されたセクションを特定のコンプレッサにルーティングしてもよい。）したがって、このようにして、圧縮マネージャ１６０は、ファイルの異なる解析された部分を選択的に圧縮することができる。

【００３６】

［００５９］アグリゲータ１７０は、圧縮ファイルの統一および標準化されたファイル形式に従って、または異なる形式に従って、複数の異なる圧縮されたセクションを圧縮ファイルに集約または結合する。圧縮後、選択的に圧縮ファイル（例えば、圧縮された表形式データファイルまたは圧縮された区切られたテキストファイル）は、記憶領域１３０に記憶されるか、異なる記憶領域に記憶されるか、所定の場所もしくはデバイスに送られるか、または、別のスキーム、例えば、システムのための所定の制御命令によって決定されるスキームに従ってルーティングされ得る。

【００３７】

統一された／標準化されたデータ処理

［００６０］１つまたは複数の実施形態によれば、プロセッサ１１０は、ファイルの異なる部分を選択的に圧縮すること（したがって、後に選択的に展開すること）を可能にする、統一および／または標準化されたファイル形式に準拠するようにデータソースからの情報を処理する。処理は、１つまたは複数の所定の機能をサポートすることを可能にするファイル形式でゲノムアノテーションデータを保存することを可能にするために実行され得る。これらの機能の例としては、高速クエリ、ランダムアクセス、複数の解像度（例えば、ズーム）、選択的暗号化、認証、アクセス制御、およびトレーサビリティなどが挙げられる。

【００３８】

［００６１］一実施形態では、ゲノム用途データは、大幅な圧縮ゲインを可能にする形式に処理される。これは、例えば、データの複数の異なる属性を複数のセクション（例えば、複数のテーブル）に分離することによって実現され得る。その場合、複数のコンプレッサは、ゲノムアノテーションデータ内の分離された属性のそれぞれを圧縮するために使用され得る。一実施形態では、すべてのまたは一部のコンプレッサは、コンプレッサに割り当てられたファイルの特定の属性を圧縮するように設計された、特殊化またはカスタマイズされたコンプレッサであり得る。プロセッサ１１０によって実行される処理はまた、アノテーションに関連するシーケンシングデータのためのメタデータおよびリンケージ、ならびにシーケンシングデータに関連していても関連していなくてもよい他のアノテーシ

ョンデータへのリンケージを生成し得る。メタデータおよび／またはリンケージは、ゲノム用途データと同じ統一および標準化された形式で保存されてもよく、これにより、シーケンシングデータのための既存のファイル形式（例えば、ＭＰＥＧ－Ｇ）とのシームレスな統合が可能になる。

【００３９】

［００６２］図２は、統一および標準化されたファイル形式に従って情報の圧縮を管理するための方法を示す。方法は、例えば、図１のシステムまたは別のシステムによって、後述される動作を実行するように１つまたは複数のプロセッサを制御するためのメモリに保存された命令に従って、実行され得る。説明を目的として、方法が図１のシステムによって部分的にまたは全体として実行され、また、データがゲノムアノテーションデータであると仮定する。

【００４０】

［００６３］図２を参照して、方法は、２１０において、データソース１４０からデータ（例えば、ゲノムアノテーションデータ）を受け取ることを含む。ゲノムアノテーションデータは、例えば、データを生成するために使用されるアノテーションツールによって生成される既存の形式で受け取られ得る。既存の形式の一例はＭＰＥＧ－Ｇ形式であるが、別の実施形態では、アノテーションデータは異なる既存の形式で受け取られてもよい。データは、ローカル接続またはネットワーク接続を介して、データソースからプロセッサ１１０によって受け取られ得る。

【００４１】

［００６４］２２０において、統一および標準化されたファイル形式への処理のためにゲノムアノテーションデータの複数の部分が取り出される。ゲノムアノテーションデータは既存の既知の形式に従って編成されているため、既知の形式の各部分で取り出されるコンテンツは、プロセッサ１１０に知られている特定のタイプのデータを有する。したがって、プロセッサ１１０を制御する命令は、１つまたは複数の実施形態に係る統一および標準化された形式で新しいファイルを生成するために、既知の形式の各部分において特定のタイプのデータを取り出すことができる。

【００４２】

［００６５］２３０において、プロセッサ１１０は、例えば図３に示されるように、統一された標準ファイル形式３００に準拠するようにゲノムアノテーションデータを処理するために、メモリ１２０内の命令を実行する。例えば、プロセッサは、ファイルヘッダセクション３１０、ファイル保護セクション３２０、ファイルメタデータセクション３３０、ファイルトレーサビリティセクション３４０、テーブル情報およびインデックスセクション３５０、１つまたは複数の圧縮パラメータを示すセクション３６０、ならびに１つまたは複数のテーブル３７０を生成するようにデータを処理することができる。図４のチャートには、これらのセクションおよび対応する説明とともに、後により詳細に説明される他の情報が示されている。

【００４３】

［００６６］ファイルヘッダセクション３１０は様々なタイプの識別情報を含む。例えば、ファイルヘッダセクションは、ファイル名フィールド３１１、ファイルタイプフィールド３１２、およびファイルバージョンフィールド３１３を含み得る（例えば、図５を参照されたい）。ファイル名フィールドは、例えば、アノテーションツールによって、またはユーザからの情報に基づいて生成された名前をファイルに付ける文字列を含み得る。ファイルタイプフィールドは、ファイルのタイプ（例えば、バリアント、遺伝子発現など）を示す文字列を含み得る。ファイルバージョンフィールドは、ファイルの目的および／またはバージョンを識別する文字列（例えば、このファイルはアップデートを記録するためのものである）を含み得る。

【００４４】

［００６７］ファイル保護セクション３２０は、ファイルのアクセス制御ポリシーに関連する情報を含み、これは、例えば、ユーザ、管理者、データ所有者、またはデータへの

アクセスおよび／もしくはデータの配布を管理する他のエンティティによって決定され得る。アクセス制御ポリシーは、ファイル全体に対して同じレベルのアクセスを示してもよく、またはファイルの異なる部分に対して異なるレベルのアクセスを示してもよい。一実施形態では、ファイル保護セクションは、ペイロードサイズフィールド３２１およびペイロードフィールド３２２を含み得る（例えば、図６を参照されたい）。ペイロードサイズフィールドは、例えば、このフィールドまたはそれに関連する情報がスキップされる、または考慮に入れられることを可能にする保護情報ペイロードのサイズを示す１つまたは複数の整数値を含み得る。ペイロードフィールドは、ファイルの保護およびアクセス制御ポリシーに関する情報を含む。

【００４５】

［００６８］ファイルメタデータセクション３３０およびファイルトレーサビリティセクション３４０は、後により詳細に説明されるような情報を含み得る。一実施形態では、ファイル保護情報、メタデータ、バージョニング、およびトレーサビリティ情報は、例えば一般的な圧縮アルゴリズム、例えば７ｚｉｐを使用して圧縮されてもよい。一部の実装形態では、（例えば、ＭＰＥＧ－Ｇ　ｐａｒｔ　３で実行されるように）ＵＲＩ（ｕｎｉｆｏｒｍ　ｒｅｓｏｕｒｃｅ　ｉｄｅｎｔｉｆｉｅｒ）表記とともにこの情報を圧縮するためにＪＳＯＮ／ＸＭＬ／ＸＡＣＭＬベースのスキーマが使用されてもよい。

【００４６】

［００６９］テーブル情報およびインデックスセクション３５０は、ファイル内のテーブルの数（ｎ）を示すｎＴａｂｌｅｓフィールド３５１を含み得る。一実施形態では、統一および標準化された形式は、アノテーションデータを複数のテーブル（例えば、ｎ≧２）内に保存し得る。この場合、複数のテーブルの異なるテーブルは様々なタイプのアノテーション付きデータを保存し得る。あるケースでは、同じゲノムアノテーションデータを異なる解像度で保存するためにファイル内の異なるテーブルが使用されてもよい。これはテーブルの内容の一例として提供されているに過ぎない。他の実施形態では、テーブルは他のタイプのアノテーションデータを保存することができる。

【００４７】

［００７０］ｎＴａｂｌｅｓフィールド３５１に加えて、テーブル情報およびインデックスセクションは、Ｔａｂｌｅ　ＩＤ［ｉ］セクション３５２、ＴａｂｌｅＩｎｆｏ［ｉ］セクション３５３、およびＢｙｔｅＯｆｆｓｅｔ［ｉ］セクション３５４を保存することができる。Ｔａｂｌｅ　ＩＤ［ｉ］セクション３５２は、ファイル内のｎ個のテーブルのうちの対応するテーブルｉの一意の識別子を示す１つまたは複数の整数値を保存し得る。ＴａｂｌｅＩｎｆｏ［ｉ］セクション３５３は、ｎ個のテーブルのうちのテーブルｉのうちの対応するものの解像度を示す情報を保存し得る。ＢｙｔｅＯｆｆｓｅｔ［ｉ］セクション３５４は、ファイル内のｎ個のテーブルのうちのテーブルｉのうちのそれぞれのバイトオフセットを示す１つ以上の整数値を保存する。この形式を使用することで、例えばＪＳＯＮ／ＸＭＬのような形式でＴａｂｌｅＩｎｆｏフィールドを使用して、ファイル全体を読み取ることなく、ファイル内のテーブルのうちの１つまたは複数に関する基本情報（例えば、解像度レベル）が抽出され得る。同様に、圧縮ファイル内のテーブルのバイトオフセットは特定のテーブルに直接ジャンプするために利用可能であり得る。

【００４８】

［００７１］２４０において、受け取られた（ソース）ファイル内のゲノムデータが複数のチャンクまたはチャンクグループに分割される。複数のチャンクは同じ固定サイズを有してもよいし、または、例えば、チャンクが属性依存であるか、および／もしくはデータに関連する他のパラメータに基づくかに依存する可変サイズを有し得る。チャンクは、例えば、上記のようにパーサ１５０によって生成されてもよい。

【００４９】

［００７２］２５０において、分割されたチャンクまたはチャンクグループを圧縮するために異なるコンプレッサ（または圧縮アルゴリズム）が指定される。コンプレッサは、統一および標準化されたファイル外であってもよく、かつ／または、例えば、埋め込まれ

たコードによって、もしくは圧縮パラメータテーブル内に保存された選択可能なものによって内部的に開始されてもよい。

【００５０】

［００７３］したがって、圧縮パラメータセクション３６０は、ファイルのテーブル内に含まれる異なる情報（例えば、属性）を圧縮するために使用される異なるコンプレッサ（または圧縮アルゴリズム）の数を示すフィールド３６１を含み得る。この情報は、例えば、１つまたは複数の整数値として表されてもよい。セクション３６０はまた、一意の識別子によってインデックス付与された異なるコンプレッサを列挙する情報を含むフィールド３６２を含み得る。これらはテーブル内で参照できるため、複数のテーブルにおいて、または複数の属性のために使用されるコンプレッサの繰り返しの記述が回避される。

【００５１】

［００７４］一実施形態では、図７に示されるように、フィールド３６２（または、圧縮パラメータセクション３６０内のフィールド３６２とは別の、もしくはフィールド３６２に階層的に関連するフィールド）は、ＣｏｍｐｒｅｓｓｏｒＩＤフィールド３６３、ｎＤｅｐｅｎｄｅｎｃｉｅｓフィールド３６４、ＣｏｍｐｒｅｓｓｏｒＮａｍｅＬｉｓｔフィールド３６５、およびＣｏｍｐｒｅｓｓｏｒＰａｒａｍｅｔｅｒｓＬｉｓｔフィールド３６６を含み得る。

【００５２】

［００７５］ＣｏｍｐｒｅｓｓｏｒＩＤフィールド３６３は、複数のコンプレッサのうちの対応するコンプレッサの一意の識別子を示す情報（例えば、文字列）を含んでもよい。したがって、ＣｏｍｐｒｅｓｓｏｒＩＤフィールドは、ファイルのそれぞれの属性または他の部分の圧縮に使用されるコンプレッサを識別するために使用され得る。一実施形態では、一意の識別子は、対応するコンプレッサを指し示すためにテーブル内で使用され得る。

【００５３】

［００７６］ｎＤｅｐｅｎｄｅｎｃｉｅｓフィールド３６４は、属性の圧縮が他の属性に基づいて実行されることを示す情報（例えば、１つまたは複数の整数値）を含み得る。例えば、システムプロセッサは、１つまたは複数の他の属性を副次的情報として使用して１つの属性を圧縮するための情報を処理してもよい（循環依存関係がないことを条件として）。その場合、この情報は、統一および標準化されたファイル形式のｎＤｅｐｅｎｄｅｎｃｉｅｓフィールド３６４内に含まれ得る。一実施形態では、変数ｎＤｅｐｅｎｄｅｎｃｉｅｓの指標は、展開に使用される依存関係属性の数を示し得る（例えば、図１２の属性情報構造には、対応するａｔｔｒｉｂｕｔｅＩＤの例が示されている）。コンテキストベースの算術符号化などのコンプレッサは、副次的情報の組み込みを簡単にサポートし得る。他の属性からの副次的情報を組み込む別のメカニズムは、他の属性の値に基づいて現在の属性の値を並べ替えるか、または分割することである。これにより、類似する値をまとめて、より良い圧縮を提供することができる。パラメータは、リスト内の各コンプレッサによって使用される依存関係を記述する。

【００５４】

［００７７］ＣｏｍｐｒｅｓｓｏｒＮａｍｅＬｉｓｔフィールド３６５は、コンプレッサ名のリストを示す情報（例えば、文字列）を含み得る。コンプレッサ名は、ファイルの外部の（例えば、圧縮モジュールまたはシステムプロセッサ１１０によって実装される）１つまたは複数のコンプレッサ、（例えば、埋め込まれたコードまたは実行コードを介して）ファイルに埋め込まれた１つまたは複数のコンプレッサ、または両方を指す可能性がある。コンプレッサ名およびパラメータのうちの１つまたは複数は、標準コンプレッサに対応するか、または展開メカニズムを記述し得る。一実施形態では、コンプレッサ実行コードおよびパラメータのうちの１つまたは複数は、例えばＣｏｍｐｒｅｓｓｏｒＮａｍｅを「ＥＭＢＥＤＤＥＤ」に設定することによって、ＣｏｍｐｒｅｓｓｏｒＰａｒａｍｅｔｅｒｓ内に示され得る。一部の実施形態では、同じ属性またはファイル部分を圧縮するために複数のコンプレッサが使用され得る。この場合、複数のコンプレッサが順番に適用さ

れ、統一および標準化されたファイル形式に含まれるリストに表示され得る。
【００５５】
［００７８］ＣｏｍｐｒｅｓｓｏｒＰａｒａｍｅｔｅｒｓＬｉｓｔフィールド３６６は、指定されたコンプレッサの全部または一部によって圧縮された情報の展開を実行するために使用されるべき１つまたは複数のパラメータを示す情報（例えば、リストまたは他の形式）を含んでもよい。
【００５６】
［００７９］２６０において、方法は、本明細書に記載されるような対応するフィールド内に保存されるメタデータ、トレーサビリティ情報、リンケージ情報、インデックス、および保護情報を含む追加情報を生成するように、データの所有者もしくは管理者によって生成された、受け取られた（ソース）ファイルおよび／またはポリシー情報を処理することを含む。
【００５７】
［００８０］２７０において、後に取り出すために、統一および標準化されたゲノムアノテーションデータファイルが保存される。複数の異なるデータチャンクが選択的に圧縮されているため、ファイル内の他の情報を展開する必要なく、圧縮ファイルの指定された部分のみを展開して、ファイル内の特定の情報にアクセスしてもよい。これは、任意の所与の時点で特に関心が持たれる可能性のあるゲノムデータの部分のみをターゲットにすることで、処理速度およびユーザの利便性を向上させる。
【００５８】
［００８１］統一および標準化されたファイルにおいて指定される情報のすべてまたは一部は、テーブル、構造、インテックス内に示された相関付けられた情報、またはゲノムアノテーションデータの選択的圧縮および展開、ならびにアクセスを可能にする他のタイプの相関付けられた情報に対応し得る。相関付けられた情報内のテーブルの例については、後により詳細に説明する。
【００５９】
テーブル
［００８２］各統一および標準化されたファイル内のテーブル３７０は、保存されている属性および／または他の情報に関して、互いに関連していても独立していてもよい。各テーブルは複数のセルを含み、各セルは１つまたは複数の属性を含み得る。各属性は特定のデータ型を有し、コンプレッサのうちの１つを使用して圧縮され得る。テーブル内のセルが複数の属性を含む場合、圧縮を向上させるために、および後に展開するときに属性への選択的アクセスを可能にするために、テーブルの各セル内の複数の属性に対応するデータは別々に圧縮され得る。ゲノム機能的アノテーションファイルでは、属性は、例えば、染色体、開始位置、終了位置、フィーチャＩＤ、およびフィーチャ名を含み得る。
【００６０】
［００８３］圧縮は、異なる実施形態では異なる方法で実行され、例えば、属性ごとに、セルごとに、テーブルごとに、などで実行され得る。例えば、これらの目的のために、同じテーブル内のセルグループが他のセルグループとは別に圧縮されてもよい。一緒にグループ化されたセルは、同じデータに対応する属性を有し、他のセルグループは、同じまたは異なるデータに対応する異なる属性を有し得る。一実装形態では、各セル内の属性は異なる圧縮アルゴリズムを使用して圧縮されてもよく、または、１つまたは複数のセル内の属性は第１の圧縮アルゴリズムによって圧縮され、１つまたは複数の他のセル内の属性は第２の圧縮アルゴリズムを使用して圧縮され、１つまたは複数の他のセルの属性は第３の圧縮アルゴリズムを使用して圧縮されてもよい（以下同様）。
【００６１】
［００８４］ある追加の実装形態では、コンプレッサ１６１１、１６１２、・・・、１６１Ｎの全部または一部は、複数のセルのうちのそれぞれのセル内の異なるデータ／属性タイプに特化した異なる圧縮アルゴリズムを実行することができる。以下、統一および標準化されたファイル形式の例が複数のテーブルを含むものとして議論されるが、一実施形

態では、ファイル形式は１つのテーブルのみを含んでもよい。
【００６２】
［００８５］一部の例では、各セル内のコンテンツは属性として説明されるが、コンテンツは、他の実施形態では他のタイプのデータ（例えば、データのタイプ、属性、特性など）または複数のタイプのデータの組み合わせを含み得る。例えば、１つのファイル内のテーブル３７０は、すべてが同じゲノム配列を表すコンテンツまたはデータを保存し得るが、配列内の各テーブルは、異なる解像度でそのコンテンツまたはデータを表し得る。
【００６３】
［００８６］システムプロセッサ１１０は、ファイルテーブル３７０内の情報を様々な形で処理および保存することができる。例えば、処理は、一次元テーブルおよび／または多次元テーブルを生成するために実行され得る。所与の標準化されたファイル内のすべてのテーブルが同じタイプのものであってもよく、またはタイプの組み合わせを含んでもよく、例えば、一次元テーブルおよび多次元テーブルの両方を含み得る。ゲノムアノテーション用途では、一次元テーブルは、例えば、ゲノムアノテーションデータまたは定量的ブラウザトラックを含み、多次元テーブルは、例えば、１つまたは複数のサンプルに関するバリアントコールデータおよび遺伝子発現データを含み得る。一実施形態では、一次元テーブルは複数の属性を含み得る。
【００６４】
［００８７］図８は、セルの一次元配列８１０内にデータを保存するテーブルの例を示す。配列内の各セル８２０は、属性１、属性２、および属性３とラベル付けされた複数の属性を含む。テーブル内の各セルは同じ数の属性を有しているが、一部の実施形態では、テーブルのセル内の属性の数は異なっていてもよい。テーブル構造の例、およびテーブル構造に含めるためにシステムプロセッサがコンテンツを処理し得る方法の例を後により詳細に説明する。
【００６５】
［００８８］図９は、セルの二次元配列９１０内にデータを保存する多次元テーブルの例を示す。テーブル内の各セル９２０は、データに対応する１つまたは複数の属性を含む。セル内の属性の数は同じであっても異なっていてもよい。テーブル構造の例、およびテーブル構造に含めるためにシステムプロセッサがコンテンツを処理し得る方法の例を後により詳細に説明する。
【００６６】
［００８９］一実施形態では、各ファイル内の多次元テーブル３７０のうちの１つまたは複数は、二次元配列９１０に加えて、次元固有属性を保存し得る。次元固有属性を保存するための例示的な構成が同様に図９に示されている。ここでは、配列９１０内に保存された属性とともに、テーブル３７０は、次元固有属性を２つの追加テーブル内に保存する。次元固有属性の第１の配列９４０は行属性を保存するセルを含む。次元固有属性の第２の配列９５０は列属性を保存するセルを含む。したがって、各配列９４０および９５０は、テーブル３７０内の追加の一次元テーブルの一部であると考えることができる。これらのフィーチャを含むテーブル３７０の一例は、二次元配列テーブル９１０において複数のサンプルのバリアントデータを表すことを企図し、サンプルの遺伝子型およびサンプルレベルの尤度が２Ｄ配列内の各セルの属性である。さらに、配列テーブル９４０は、バリアント位置に対応する次元固有属性を保存してもよく、配列テーブル９５０は、サンプル名に対応する次元固有属性を保存してもよい。このテーブル構造の例、およびテーブル構造に含めるためにシステムプロセッサがコンテンツを処理し得る方法の例を後により詳細に説明する。
【００６７】
［００９０］図１０は、システムプロセッサ１１０がどのようにしてゲノムアノテーションデータを特定のテーブル形式に処理し得るかの実施形態を示す。図１１は、テーブルを生成するために処理される情報のためのテーブル形式の実施形態を示す。
【００６８】

［００９１］図１０および図１１に示されるように、表形式は、統一および標準化されたファイル形式のセクションに類似する多数のセクションを含む。例えば、表形式は、テーブルヘッダセクション１０１０、テーブル保護セクション１０２０、およびテーブルメタデータセクション１０３０を含み得る。テーブルヘッダセクション１０１０は、テーブルの名前を示すＴａｂｌｅＩＤフィールド１１０５、ならびにペイロード（データ）タイプおよび／または他の情報を示すＴａｂｌｅＩｎｆｏフィールド１１１０を含み得る。テーブル保護セクション１０２０は、テーブルのアクセス制御ポリシーを示す情報を含み得る。テーブルメタデータセクション１０３０は、リンケージ、トレーサビリティ、および／または他の情報を含み得る。

【００６９】

［００９２］これらのフィーチャに加えて、テーブルは、要約統計量セクション１０４０、属性セクション１０５０、インデックスセクション１０６０、およびデータセクション１０７０を含み得る。要約統計量セクション１０４０は、平均、カウント、分布、および／または、１つまたは複数のキー値に対応し得るテーブルに保存されたデータ／属性に関連する他の情報を示す情報を含み得る。統計は、例えば、関連性のある統計量への高速アクセスを可能にするために使用され得る。この情報は図１１の参照番号１１４０に対応する。

【００７０】

［００９３］属性セクション１０５０は、本明細書に記載されるような一次元および／または多次元配列テーブルのための様々な属性を保存し、固有次元属性を有しても有さなくてもよい。例えば、次元固有属性がテーブルに含まれている場合、セクション１０５０は、各次元ｉ（ｉ＝１、・・・、Ｎ、Ｎ≧２）のサイズ、名前、およびメタデータ、ならびに、テーブルに示されているように各次元に対応する次元固有属性を示すフィールド１１５０を含み得る。二次元データの場合、ＳｙｍｍｅｔｒｙＦｌａｇフィールドはまた、次元配列が対称であるか否か、例えば、対称であるＨｉ－Ｃデータであるか否かを示す情報を保存し得る。

【００７１】

［００９４］テーブルは、メイン配列（または二次元配列）内の属性の数を示すｎＡｔｔｒｉｂｕｔｅｓＭａｉｎフィールド１１６０を含み、その後には各属性の属性情報が続く。属性情報は、本明細書でより詳細に説明されるコンテンツおよび構造を含み得る。また、フィールド１１６０は、２Ｄ配列テーブルの属性／データのバイトオフセット情報を含み得る。異なるアルゴリズムと、属性に対応するバイトオフセットとに基づく属性の選択的圧縮は、他の属性、テーブルセクション、または統一および標準化されたファイルに含まれる他のデータを展開する必要なく、展開中に該属性（ゲノムアノテーションデータの特定の部分）に選択的にアクセスすることを可能にし得る。インデックスセクション１０６０は、本明細書でより詳細に記載されるようなインデックスを保存し、データセクション１０７０は、例えば、本明細書に記載されるようなチャンクまたはチャンクグループを含むデータを保存し得る。

【００７２】

［００９５］図１２は、統一および標準化されたファイルのテーブルまたは他のセクション内に含まれ得る属性情報、およびそれに関連付けられた構造を生成するために、システムプロセッサがどのようにしてゲノムアノテーションデータを処理するかの実施形態を示す。

【００７３】

［００９６］図１２を参照して、属性情報は、ＡｔｔｒｉｂｕｔｅＩｎｆｏＳｉｚｅフィールド１２０５、Ａｔｔｒｉｂｕｔｅ　ＩＤフィールド１２１０、Ａｔｔｒｉｂｕｔｅ　Ｎａｍｅフィールド１２１５、ＡｔｔｒｉｂｕｔｅＭｅｔａｄａｔａ１２２０フィールド、およびＡｔｔｒｉｂｕｔｅ　Ｔｙｐｅフィールド１２２５を含む構造に処理され得る。ＡｔｔｒｉｂｕｔｅＩｎｆｏＳｉｚｅフィールドは、この情報のサイズを示す情報を含み、これは、プロセッサがこの構造をスキップすることを可能にする。Ａｔｔｒｉｂｕｔｅ

ＩＤフィールドは、属性のための一意の識別子を含み得る。ＡｔｔｒｉｂｕｔｅＮａｍｅフィールドは、属性の名前を示す情報を含み、Ａｔｔｒｉｂｕｔｅ Ｍｅｔａｄａｔａフィールドは、属性を１つまたは複数の他の属性とリンクまたはグループ化するための情報を含み得る。

【００７４】

［００９７］Ａｔｔｒｉｂｕｔｅ Ｔｙｐｅフィールド１２２５は、属性が２つのタイプのうちの１つであることを識別する情報を含み得る。第１のタイプは基本型であり、属性が文字、文字列（ｎｕｌｌ終端）、ｆｌｏａｔ、ｄｏｕｂｌｅ、ブーリアン、異なるビット幅を有する符号付きおよび符号なし整数に対応するかを示す。第２のタイプは派生型であり、属性が可変長配列および固定長配列のどちらに対応するかを示す。

【００７５】

［００９８］これらのフィーチャに加えて、属性情報構造は、ＤｅｆａｕｌｔＶａｌｕｅフィールド１２３０、Ｓｕｍｍａｒｙ Ｓｔａｔｉｓｔｉｃｓフィールド１２３５、およびＣｏｍｐｒｅｓｓｏｒＩＤフィールド１２４０を含み得る。ＤｅｆａｕｌｔＶａｌｕｅフィールドは、属性のほとんどの値が所定のデフォルト値と等しいか否かを示す。そうである場合、所定のタイプの符号化（例えば、スパース符号化）を使用して、少なくともファイル内の対応する属性が符号化され得る。要約統計量フィールドは、平均、カウント、分布、および／または、属性の高速分析を可能にし得る他の統計データを示す情報を含み得る。

【００７６】

［００９９］Ｃｏｍｐｒｅｓｓｉｏｎ ＩＤフィールド１２４０は、属性に対して使用されるべき圧縮アルゴリズムのタイプを示す情報を含む。この情報は、例えば、属性を圧縮するために割り当てられた、コンプレッサ１６１１、１６１２、・・・、１６１Ｎのうちの１つを識別し得る。この情報は、システムプロセッサによって圧縮形式で取り出されたときに属性を復元するためにどの展開アルゴリズムが使用されるべきかを制御する。展開プロセス中にコンプレッサが副次的情報／コンテキストを使用する場合、対応する依存関係属性も属性情報内で指定され得る。多次元配列の場合、副次的情報は、多次元配列属性から、または次元固有属性から取得され得る。例えば、ＶＣＦファイルでは、（二次元配列テーブルの属性である）遺伝子型データの圧縮のための副次的情報としてバリアント固有フィールド（例えば、次元固有属性）が使用されてもよい。一実施形態では、システムプロセッサは、展開に必要な追加データを変数ＣｏｍｐｒｅｓｓｏｒＣｏｍｍｏｎＤａｔａ内に含めるように属性情報を処理し得る。この追加データは、例えば、すべてのチャンクに共通であり得る。この追加データは、ゲノムアノテーションデータファイルの全部または一部から計算されたコードブック、辞書、または統計モデルを保存するのに有用である可能性がある。チャンクおよびチャンクに関連する処理については、後により詳細に説明する。

【００７７】

コンプレッサ

［０１００］圧縮マネージャ１６０によって受け取られた情報（例えば、属性、データ、および他の情報）は、例えば、図１に示されるシステムの異なるコンプレッサ１６１１、１６１２、・・・、１６１Ｎを使用して選択的に圧縮され得る。これらのコンプレッサの非限定的なリストは以下を含む。

・ランレングス圧縮：この圧縮アルゴリズムは、例えば、同じ値を有する長いランのために使用され、ランの値および長さに基づいて置換動作を実行することを含み得る。

・差分符号化：この圧縮アルゴリズムは、例えば、数値のシーケンスを増加させるために使用され、連続する値の間の差に基づく置換動作を含み得る。

・辞書ベース／列挙：この圧縮アルゴリズムは、例えば、少数のオプションのセットに対応する値を有する属性のために使用され得る。この場合、アルゴリズムは、セット内のインデックスに基づいて置換動作を実行し、結果として得られた辞書をＣｏｍｐｒｅｓｓｏｒＣｏｍｍｏｎＤａｔａフィールド内に保存し得る。

・スパース：この圧縮アルゴリズムは、例えば、例えば図１２の属性情報において示されるように、デフォルト値とわずかに（例えば、所定の値未満）異なる属性を圧縮するために使用され得る。このタイプの圧縮は、属性を、非デフォルト値のための座標位置および値として表すことを含み得る。圧縮を改善するために、座標位置は各チャンク内でさらに差分符号化されてもよく、例えば、二次元スパース配列において、行インデックスを差分符号化し、各行内で列インデックスを差分符号化してもよい。
・可変長配列：この圧縮アルゴリズムは、例えば、可変長配列を値のストリームと長さのストリームとに分離するために使用されてもよい。
・トークン化：この圧縮アルゴリズムは、例えば、構造化文字列属性について使用され、属性を異なるタイプのトークンに分割し、前の値（例えば、前のトークン、差分、新しい値などと合致）に基づいて各トークンを符号化することを含み得る。
・汎用圧縮／エントロピー符号化法：これらのタイプの圧縮法は、例えば、ｇｚｉｐ、ｂｚｉｐ２、７－ｚｉｐ、適応算術符号化、および他の可逆データ圧縮（例えば、限定はされないが、ＢＳＣの可逆ブロックソーティング圧縮）を含む。

【００７８】

［０１０１］上記の圧縮アルゴリズムのリストは限定的なリストであることを意図したものではない。一実施形態では、上記コンプレッサのうちの１つまたは複数に加えて、またはその代わりに、１つまたは複数の特殊化またはカスタマイズされた圧縮アルゴリズムが含まれていてもよい。そのような特殊化された圧縮アルゴリズムの一例はＧＴＣ（遺伝子型コンプレッサ）であり、バリアントコールデータを圧縮するために使用され得る。そのような圧縮は、圧縮されたテーブルの行／列に含まれる情報への高速ランダムアクセスをサポートする可能性がある。場合によっては、例えば、より高速な選択的アクセスを可能にするために、ファイルの一部が圧縮されなくてもよい。

【００７９】

［０１０２］コンプレッサ１６１１、１６１２、・・・、１６１Ｎのうちの１つまたは複数は複数のストリームを生成し得る。一例は、座標のストリームおよび値のストリームを生成するスパースコンプレッサである。一実施形態では、これらのストリームは、例えば、適切に指定されたパラメータを使用して実装される異なるエントロピーコーダを使用して圧縮されてもよい。例えば、以下を検討する。

```
ＣｏｍｐｒｅｓｓｏｒＮａｍｅＬｉｓｔ＝[‘ｓｐａｒｓｅ’,’ｇｚｉｐ’,’７－ｚｉｐ’]
ＣｏｍｐｒｅｓｓｏｒＰａｒａｍｅｔｅｒＬｉｓｔ＝[
｛“ｏｕｔＳｔｒｅａｍｓ”：[“ｃｏｏｒｄｉｎａｔｅ”,“ｖａｌｕｅ”]｝,
｛“ｉｎＳｔｒｅａｍｓ”：[“ｃｏｏｒｄｉｎａｔｅ”]｝,
｛“ｉｎＳｔｒｅａｍｓ”：[“ｖａｌｕｅ”]｝
]
```

【００８０】

［０１０３］この場合、ｇｚｉｐ圧縮が座標のストリームに適用され、７－ｚｉｐが値のストリームに適用されてもよい（例えば、パラメータを表すためにＪＳＯＮが使用され得る）。これにより、所定の、または最適な結果を生成し得る、データストリームごとのコンプレッサの適用が可能になる。ストリームが指定されていない場合、同じ圧縮法がすべての受け取られたストリームに適用されてもよい。

【００８１】

［０１０４］上記したように、１つまたは複数の外部コンプレッサに加えて、またはその代わりに、（統一および標準化されたファイル形式に処理された）圧縮ファイルが１つまたは複数の埋め込まれたコンプレッサを含んでもよい。すなわち、システムプロセッサは、統一および標準化されたファイル形式内で実行可能な埋め込まれたコンプレッサを、好ましくは適切なセキュリティ保護とともに含むよう、ゲノムアノテーションデータを処理し得る。埋め込まれたコンプレッサの場合、対応する展開実行コードが、悪意のあるソフトウェアから保護するための発信元および真正性の証明としての１つまたは複数デジタル署名とともに、圧縮パラメータに含まれ得る。異なるプラットフォーム間での相互運用

性のために、展開実行コードのために標準化された仮想マシンバイトコードが使用されてもよい。

【００８２】

チャンクおよびインデックス付与

［０１０５］図１３は、統一および標準化されたファイル形式に構成された情報を処理するための方法の実施形態を示す。この方法では、圧縮ファイルは、展開および情報復元中に圧縮された情報に効率的にランダムアクセスできるように処理される。

【００８３】

［０１０６］図１３を参照して、方法は、１３１０において、属性、データ、（１Ｄおよび／または２Ｄ）テーブル、および／または各テーブル３７０（またはファイル全体）内の他の情報をチャンクに分割することを含む。チャンクはすべて同じ固定サイズを有してもよく、または可変サイズを有してもよい。すべての属性に同じチャンクが使用されてもよいし、そうでなくてもよい。一実施形態では、効率的なランダムアクセスを可能にするために、各テーブル３７０内の属性は矩形のチャンクに分割される。

【００８４】

［０１０７］１３２０において、分割後、チャンクは、複数のコンプレッサのうちの異なるものによって選択的に圧縮される（例えば、システムプロセッサ１１０によって実行される命令によって指し示されるように）。

【００８５】

［０１０８］次に、１３３０において、プロセッサは、統一および標準化された形式の圧縮ファイルに含められる圧縮されたチャンクまたはチャンクグループごとに、少なくとも１つのインデックスを生成する。インデックスは、圧縮ファイル内の各特定のチャンクの位置を効率的に決定するための情報を含み得る。データをインデックスに処理することにより、関心のある１つまたは複数のチャンクとは無関係な、または関連性の無い他のチャンクを展開することなく、対応するチャンクを選択的に展開するだけで、データ内の任意の位置に高速アクセスすることが可能となり得る。特定の属性の値に基づく高速ランダムアクセスをサポートするために、システムプロセッサはまた、属性固有インデックスを生成するようにデータを処理してもよい。一実施形態では、各圧縮ファイルはまた、複数のチャンクにわたって展開を実行するためのコードブックまたは統計モデルの共有を可能にする情報を含み得る。

【００８６】

［０１０９］図１４は、プロセッサによって生成されるインデックス構造の例を示す。この例は、以下で説明するように、フラグＡｔｔｒｉｂｕｔｅＤｅｐｅｎｄｅｎｔＣｈｕｎｋｓフィールドが偽である場合の一次元テーブルのためのインデックス構造に対応する。インデックスは第１のセクション１４１０および第２のセクション１４２０を含む。第１のセクション１４１０は複数のチャンクに対応する情報を含み、各チャンクは、開始インデックス情報１４１１、終了インデックス情報１４１２、およびバイトオフセット情報１４１３を含む。第２のセクション１４２０は、チャンク、または関連付けられた圧縮ファイルの他の部分に含まれる属性のための複数の追加インデックスを含む。追加インデックスはそれぞれ、インデックス付与された属性情報１４２１、インデックスタイプ情報１４２２、およびインデックスデータ情報１４２３を含み得る。インデックスファイル構造は、例えば、（例えば、図１０に示されるような）各テーブルのインデックスセクションに、または圧縮された統合および標準化されたファイルの選択されたテーブルに含まれ得る。

【００８７】

［０１１０］図１５は、１つまたは複数の実施形態に係るインデックス構造の例を示し、図１６は、１つまたは複数の実施形態に係る、図１４のインデックス構造に対応し、そのより良い説明を提供し得るチャンク構造の例を示す。

【００８８】

［０１１１］まず図１６を参照して、チャンク構造はｎＣｈｕｎｋｓフィールド１６０

５およびＶａｒｉａｂｌｅＳｉｚｅＣｈｕｎｋｓフィールド１６１０を含み得る。ｎＣｈｕｎｋｓフィールド１６０５は、特定の属性に対応するチャンクの数を示す情報（例えば、整数）を含む。ＶａｒｉａｂｌｅＳｉｚｅＣｈｕｎｋｓフィールド１６１０は、ｎＣｈｕｎｋｓが同じ固定サイズを有するか、または異なるサイズを有するかを示すフラグを含む。固定サイズのチャンクは処理がより単純であるが（特に、多次元配列テーブルに含まれている場合）、可変サイズのチャンクは、例えば、データのスパース性が大きく変動するため、固定チャンクサイズの選択が最適でない場合に有用であり得る。場合によっては、可変サイズのチャンクがゲノムアノテーションデータの特定の属性（例えば、染色体、ゲノム位置など）により適している可能性があり、したがって、これらの属性に関してより高速なランダムアクセスを可能にし得る。

【００８９】

［０１１２］チャンク構造が、可変サイズのチャンクが使用されていることを示す場合、各次元に沿った各チャンクの開始インデックス１６１５および終了インデックス１６２０がチャンク構造内に含まれ得る。この情報は図１４の圧縮ファイルインデックス構造内にも示されている。固定サイズのチャンクの場合、各次元に沿ってチャンクサイズが示され得る。いずれの場合も（固定サイズまたは可変サイズ）、ランダムアクセスを可能にするために、チャンクごとにファイル内にバイトオフセット情報１６２５が含まれ得る。

【００９０】

［０１１３］図１７は、特定の属性に関連するゲノムアノテーションデータファイルの一部分の例を示す。この例では、データは二次元配列に配置され、特定のＢｙｔｅＯｆｆｓｅｔ［ｊ］とともに、同じ固定サイズのチャンク１７１０に分割されている（例えば、ＣｈｕｎｋＳｉｚｅ（１）は５であり、ＣｈｕｎｋＳｉｚｅ（２）は１１である）。図示されているこのデータ部分は計１５個のチャンク（ｎＣｈｕｎｋｓ＝１５）を含み、ＶａｒｉａｂｌｅＳｉｚｅＣｈｕｎｋｓフィールドは固定サイズを指定するために偽値を示す。

【００９１】

［０１１４］図１８は、特定の属性に関連するゲノムアノテーションデータファイルの一部分の例を示す。この例では、データは二次元配列に配置され、可変サイズを有するチャンク１８１０～１８４０に分割されている。チャンクサイズは可変であるため、開始インデックスおよび終了インデックスが、関連付けられたＢｙｔｅＯｆｆｓｅｔとともにブロックごとに提供される。例えば、第１の矩形チャンク（チャンク１）の第１の方向についてはＳｔａｒｔｉｎｄｅｘ［１］［１］＝１およびＥｎｄｉｎｄｅｘ［１］［１］＝１７が示され、第１の矩形チャンクの第２の方向についてはＳｔａｒｔｉｎｄｅｘ［１］［２］＝１およびＥｎｄｉｎｄｅｘ［１］［２］＝１７が示される。チャンク１についても対応するＢｙｔｅＯｆｆｓｅｔ情報が提供される。同様に、第２の矩形チャンク（チャンク２）の第１の方向についてはＳｔａｒｔｉｎｄｅｘ［２］［１］＝１およびＥｎｄｉｎｄｅｘ［２］［１］＝１１が示され、第２の矩形チャンクの第２の方向についてはＳｔａｒｔｉｎｄｅｘ［２］［２］＝１８およびＥｎｄｉｎｄｅｘ［２］［２］＝３１が示される。図示されているこのデータ部分は計４個のチャンク（ｎＣｈｕｎｋｓ＝４）を含み、ＶａｒｉａｂｌｅＳｉｚｅＣｈｕｎｋｓフィールドは可変サイズを指定するために真値を示す。残りのチャンク３および４も同様に指定され得る。該属性についての４つのチャンクすべてに対応する情報が、チャンク構造、インデックス構造、対応するテーブル、および／または、ゲノムアノテーションデータの統一および標準化されたファイルに関連する他の情報に組み込まれ得る。

【００９２】

［０１１５］再び図１５を参照して、インデックス構造は、ＡｔｔｒｉｂｕｔｅＤｅｐｅｎｄｅｎｔＣｈｕｎｋｓフィールド１５１０を含む情報を含む。このフィールドは、チャンクサイズが１つまたは複数の属性に依存しているか、または、ファイルに関連付けられているすべての属性について同じチャンク化（例えば、同じチャンクサイズ）が使用されているかを示すフラグを含み得る。フラグが第１の値を有する場合、１５１５において、すべての属性について同じチャンク化が使用される。すべての属性について同じチャン

ク化を使用することで、インデックス構造のサイズが大幅に削減され、例えば、ほとんどの場合においてチャンク内のすべての属性が照会される場合に有用であり得る。

【００９３】

［０１１６］図１９は、ＡｔｔｒｉｂｕｔｅＤｅｐｅｎｄｅｎｔＣｈｕｎｋｓフラグが第１の値（例えば、偽）を有する場合における、一次元のケースのためのデータペイロード構造に含まれる情報の例を示す。この例では、データ１９１０は複数のチャンク１９２０（例えば、ｎＣｈｕｎｋｓ＝５）に分割され、各チャンクは１つまたは複数の属性１９３０に対応する。このケースでは、チャンク１は３つの属性に対応する。また、各属性はペイロードサイズおよびペイロードを示す情報を含む。

【００９４】

［０１１７］再び図１５を参照して、フラグが第２の値を有する場合、１５２０において、図１６のチャートに示されるようにして属性依存チャンク化が使用される。属性依存チャンク化は、例えば、圧縮およびランダムアクセスに関する互いに異なる属性の最適なチャンクサイズが大幅に異なる場合（例えば、所定の値を上回る場合）に有用であり得る。例えば、一部の属性がスパース（疎）で他の属性がデンス（密）である場合、同じチャンクサイズを使用すると、圧縮が最適化されない可能性がある。属性依存チャンク化は、単一の属性のすべてのチャンクが頻繁に照会される場合、例えば、所定の頻度値を上回る頻度で照会される場合にも有用であり得る。

【００９５】

［０１１８］図２０は、ＡｔｔｒｉｂｕｔｅＤｅｐｅｎｄｅｎｔＣｈｕｎｋｓフラグが第２の値（例えば、真）を有する場合における、一次元のケースのためのデータペイロード構造の例を示す。この例では、データ２０１０は１つまたは複数の属性２０２０に対応する。各属性は、データを分割して得られる１つまたは複数のチャンク２０３０を含むか、またはそれに対応する。また、各チャンクはペイロードサイズおよびペイロードを示す情報を含む。図１５および図２０の比較から明らかなように、一実施形態によれば、チャンクおよび属性の編成は動作モードに依存し、例えば、属性依存チャンク化が実行されるか否かに依存する。

【００９６】

［０１１９］一実施形態では、選択的展開プロセス中のランダムアクセスは、行番号および／または列番号を使用せずに実行されてもよい。この実施形態は、例えば、特定の属性に関してランダムアクセスが実行される用途のために、例えば、ゲノム位置に関してランダムアクセスが実行される用途のために実行され得る。

【００９７】

［０１２０］これらの場合、図１５のインデックス構造は１つまたは複数の属性固有インデックス１５２５を含み得る。例えば、属性依存チャンク化が実行される場合、ｎＡｄｄｉｔｉｏｎａｌＩｎｄｅｘｅｓが使用され得る（ｎ≧１）。行／列ごとではなく、属性ごとのより高速な照会を実行するために、インデックス構造は１つまたは複数の属性インデックス含み得る。例えば、ゲノムアノテーションデータの場合、染色体および位置の属性のためのインデックスがシステムプロセッサ１１０によって生成され、図１５のインデックス構造に含められ得る。システムプロセッサによって照会が実行されると、クエリの染色体および位置属性にマッチするチャンクまたはチャンク番号が選択的展開のために返され得る。他の実施形態では異なる属性および／または異なる数の属性が使用されてもよい。

【００９８】

［０１２１］インデックス構造はまた、ｎＡｄｄｉｔｉｏｎａｌインデックスに関連する他の情報を含み得る。例えば、属性固有インデックスごとに、インデックス構造は、ＡｔｔｒｉｂｕｔｅＩＤｓＩｎｄｅｘｅｄフィールド１５３０、ＩｎｄｅｘＴｙｐｅフィールド１５３５、ＩｎｄｅｘＳｉｚｅフィールド１５４０、およびＩｎｄｅｘＤａｔａフィールド１５４５を含み得る。ＡｔｔｒｉｂｕｔｅＩＤｓＩｎｄｅｘｅｄフィールド１５３０は、インデックスが付与された１つまたは複数の属性（例えば、染色体、ゲノム位置な

ど）のリストを含み得る。

【００９９】

［０１２２］ＩｎｄｅｘＴｙｐｅフィールド１５３５は、ＡｔｔｒｉｂｕｔｅＩＤｓＩｎｄｅｘｅｄフィールド内にリストされる各属性について実行されたインデックス付与のタイプを示す情報を含み得る。一実施形態では、１つまたは複数の属性のインデックス付与のタイプは、標準セットとは異なる可能性がある。例えば、染色体属性およびゲノム位置属性のそれぞれについて、ならびに範囲クエリについてＲツリーまたはＣＳＩインデックスが使用されてもよい。データベースタイプのクエリについてＢツリーインデックスが使用されてもよい。ゲノム範囲インデックス付与は、各チャンクの左端および右端の座標を保存でき、これは、照会された範囲と重複するチャンクを高速で識別することを可能にする。同様に、Ｂツリーインデックスは、属性値から、値およびチャンク内の値の位置を含むチャンクへのマップを保存できる。

【０１００】

［０１２３］ＩｎｄｅｘＳｉｚｅフィールド１５４０は、リストされた属性ごとにシステムプロセッサによって生成されたインデックスのサイズを示す情報を含み得る。

【０１０１】

［０１２４］ＩｎｄｅｘＤａｔａフィールド１５４５は、実際のインデックス付与データを所定の（例えば、バイナリ形式）で含むことができる。形式のタイプは、例えば、属性について生成されたインデックスのタイプに依存し得る。

【０１０２】

［０１２５］図２１は、（例えば、図１９および／または図２０のデータ内に示されるような）ペイロード情報を生成するためにシステムプロセッサ１１０によって実行され得る例示的なロジックを示す。このロジックは、ＡｔｔｒｉｂｕｔｅＤｅｐｅｎｄｅｎｔＣｈｕｎｋｓフィールド内の値に基づいて実装される。より具体的には、図１９と図２０との比較から明らかなように、ペイロードの構造はＡｔｔｒｉｂｕｔｅＤｅｐｅｎｄｅｎｔＣｈｕｎｋｓフィールドの値によって異なる。ペイロードのデータのチャンクが属性に依存しない場合、システムプロセッサは、上記のような図１９に示される構造を生成するためのロジックを実装する。ペイロードのデータのチャンクが属性に依存する場合、システムプロセッサは、上記のような図２０に示される構造を生成するための論理を実装する。

【０１０３】

［０１２６］図２２は、インデックス構造（例えば、限定はされないが、図１５のインデックス構造）に基づいてアノテーション付きゲノムデータを選択的に識別および展開するためにシステムプロセッサ１１０によって実行され得る照会方法の一実施形態を示す。照会は、次のように実装されるルックアップ動作に基づいて実行され得る。まず、２２１０において、システムプロセッサ１１０は、関心のある１つまたは複数の属性を示す情報を含むユーザクエリを受け取る。２２２０において、システムプロセッサはクエリを解析し、クエリ内の１つまたは複数の属性を識別する。クエリ内において、１つまたは複数の属性は様々な形で指定され、例えば、「ａｂｃｄ」などの文字列、またはそれぞれの属性を識別する一連の番号のうちの特定の属性番号などの形で指定され得る。２２３０において、システムプロセッサは、属性固有インデックスを検索して、クエリ内で識別される属性に対応する１つもしくは複数のチャンクまたは１つもしくは複数のチャンク番号を決定する。２２４０において、システムプロセッサは、ゲノムアノテーションデータを検索して、クエリ属性とマッチする１つもしくは複数のチャンクまたは１つもしくは複数のチャンク番号を取り出す（または他の方法でそのようなチャンクまたはチャンク番号へのアクセスを取得する）。チャンクインデックスを使用してチャンクを復元することで、条件に合致する値がフィルタリングされ得る（チャンクには合致しない値も含まれる可能性があるため）。

【０１０４】

［０１２７］２２５０において、復元されたチャンクが選択的に展開され得る（例えば、指定された属性に関係のない他のチャンクを展開することなく展開される）。複数のデ

ータチャンクが先に互いに独立して選択的に圧縮されていたため、選択的展開が可能になる。展開は、図１２の属性情報構造、および／またはインデックス、テーブル、もしくは本明細書に記載される他の構造において示されるアルゴリズムに基づいて実行され得る。チャンクがグローバル圧縮された場合、グローバル圧縮データは、例えば、属性情報構造内に示されるＣｏｍｐｒｅｓｓｏｒＣｏｍｍｏｎＤａｔａメカニズムを使用して、チャンク間で共有され得る。一実施形態では、対称二次元配列テーブルの場合（例えば、図１１のテーブルのＳｙｍｍｅｔｒｙＦｌａｇが真である場合）、チャンクは一部、例えば下の三角部分および対角線のみをカバーしてもよい。この場合、展開プロセスは、対応する下三角値で埋めることにより、上対角値に対処し得る。その他の場合、チャンクは重複することなく、インデックスの全範囲をカバーしてもよい。

【０１０５】

相互運用性

［０１２８］ゲノムアノテーションデータの統一および標準化されたファイル形式は様々な追加のフィーチャを有し得る。１つの追加のフィーチャは、この形式で生成されたファイルに関連するリンケージおよび相互運用性に関連する。一実施形態では、システムプロセッサは、ソースファイル（例えば、図１のデータソース１４０によって提供されるデータ）の任意の符号化またはフォーマッティングから独立するように、ファイル情報を本明細書に記載されるように統一および標準化された形式に処理することができる。

【０１０６】

［０１２９］別の実施形態では、システムプロセッサは、ソースファイルの符号化および／またはフォーマッティングとリンクされるように、かつ相互運用可能になるように、ファイル情報を統一および標準化された形式に処理することができる。データソースファイルは、ＭＰＥＧ－Ｇ形式／符号化または別のタイプの形式または符号化であってもよい。

【０１０７】

［０１３０］リンクされた解釈可能な状態に処理されると、ファイルは、関連付けられた情報をデータセットに保存することにより、ソースファイル（ＭＰＥＧ－Ｇファイルなど）の一部として処理され得る。ＭＰＥＧ－Ｇファイルは、調査全体のデータを保存し、各データセットグループが異なる個人に対応する。各ＭＰＥＧ－Ｇデータセットグループは、さらに、複数の異なるシーケンシングランに対応する複数のデータセットに分割される。

【０１０８】

［０１３１］単一の個人に対応するデータを保存するために、複数の異なるアノテーションファイルが複数の別個のデータセットとして組み込まれ、各データセットは単一のアノテーションファイルまたはシーケンシングデータを含む。別個のデータセットの例を以下に示す。

データセットグループ（単一の個人）→
　データセット１（シーケンシングデータ）
　データセット２（シーケンシングデータ）
　データセット３（バリアントコールデータ）
　データセット４（遺伝子発現データ）
・・・

【０１０９】

［０１３２］大規模な調査からアノテーションデータを収集する場合、データセットは次のように編成されてもよい。

データセットグループ（大規模調査）→
　データセット１（バリアントコールデータ）→
　アノテーションファイル（サンプル１）
　アノテーションファイル（サンプル２）
　・・・
データセット２（遺伝子発現データ）→

アノテーションファイル（サンプル１）
アノテーションファイル（サンプル２）
・・・
・・・
【０１１０】
［０１３３］一実施形態では、例えば、以下に示すようにして、圧縮および分析パフォーマンスを向上させるために複数の異なるアノテーションファイルがマージされてもよい。
データセットグループ（大規模調査）→
データセット１（バリアントコールデータ）→
アノテーションファイル（すべてのサンプル）
データセット２（遺伝子発現データ）→
アノテーションファイル（すべてのサンプル）
・・・
【０１１１】
［０１３４］この実装形態を実行するために、システムプロセッサは、データ型（シーケンシング／バリアント／遺伝子発現／・・・）、データセット内のアノテーションファイルの数、およびこれらのファイルの各々のバイトオフセットをサポートするために、既存のデータセットヘッダー構造を追加フィールドで拡張し得る。コンプレッサがアノテーションファイル間またはデータセット間で共有される場合、コンプレッサのパラメータは、それぞれ、データセットレベルまたはデータセットグループレベルで保存され得る。一実施形態では、アノテーションファイルのうちの１つまたは複数は、コンプレッサ名「ＰＯＩＮＴＥＲ」を有するコンプレッサ構造を含み、位置を保存する圧縮パラメータ、例えば、｛“ＤａｔａｓｅｔＧｒｏｕｐＩｄ”：１，“ＤａｔａｓｅｔＩｄ”：２，“ＣｏｍｐｒｅｓｓｏｒＩｄ”：５｝は、コンプレッサが、データセットグループ１、データセット２の中の５番目のコンプレッサで指定されているとおりであることを示す。
【０１１２】
リンケージ
［０１３５］上記のフィーチャに加えて、システムプロセッサは、異なるタイプのアノテーションデータと対応するシーケンシングデータとの間のリンケージを含む形式で統一および標準化されたファイルを生成するように、データを処理してもよい。一実施形態では、このリンケージは、ファイルに保存された、またはファイルに関連付けて保存されたメタデータに基づいて提供され得る。
【０１１３】
［０１３６］これは、システムプロセッサが、データセットグループ、またはシーケンシングデータもしくは関連するアノテーションデータを保存するデータセットを、例えばＭＰＥＧ－Ｇ ｐａｒｔ ３で説明されているＵＲＩ（ｕｎｉｆｏｒｍ ｒｅｓｏｕｒｃｅ ｉｄｅｎｔｉｆｉｅｒ）表記を使用するか、またはＪＳＯＮを使用することによって図４のＦｉｌｅＭｅｔａｄａｔａフィールド、またはＴａｂｌｅＭｅｔａｄａｔａフィールド内で指定することによって達成され得る。例えば、シーケンシングデータセットへのリンケージを提供するために、次のＪＳＯＮがＦｉｌｅＭｅｔａｄａｔａ内で使用されてもよい。

```
“Ｌｉｎｋａｇｅｓ”：［｛
  “ＤａｔａＴｙｐｅ”　　：“Ｓｅｑｕｅｎｃｉｎｇ”，
  “ＤａｔａｓｅｔＧｒｏｕｐ”：５，
  “Ｄａｔａｓｅｔ”　　：２
｝］
```

上記の例は単一のリンケージのみを示すが、別の実施形態では複数のリンケージが提供されてもよい。
【０１１４】
［０１３７］追加で、または代わりに、システムプロセッサはテーブルレベルのリンケ

ージを生成してもよい。一実施形態では、システムプロセッサは、インデックスによってテーブルレベルのリンケージを生成することができる。この場合、例えば、あるテーブルのｎ番目の行（列）が別のテーブルのｍ番目の行（列）に対応し得る。このタイプのリンケージは、複数のアノテーションファイル／テーブルが同じ行／列を共有する場合、繰り返しを防ぐことができる（例えば、まだマージされておらず、同じバリアントからなる複数のＶＣＦ）。同様に、このタイプのリンクは、サンプルに関連する情報が単一のテーブルに保存されており、ＶＣＦテーブルおよび遺伝子発現テーブルの両方がこれにリンクしている場合に有用であり得る。

【０１１５】

［０１３８］別の実施形態では、システムプロセッサは、インデックスによってテーブルレベルのリンケージを生成することができる。この場合、値を別のテーブル内の属性とマッチさせることによって特定の属性がリンクされ得る。例えば、遺伝子発現データは、遺伝子に関する詳細な情報を伴わずに遺伝子名を含み得る。遺伝子に関する詳細な情報は別のファイルで入手可能である。このようなリンケージの使用例としては、自己免疫疾患に対応し、ヒト６番染色体内の座標範囲を指定するＭＨＣ（主要組織適合遺伝子複合体）のすべての遺伝子の遺伝子発現データを要求するクエリが考えられる。このクエリに対処するために、ゲノム座標インデックスに基づいて遺伝子情報ファイルから座標範囲のための複数の遺伝子名が取得され、これらの名前を遺伝子発現ファイル内で照会することで必要なデータが取得され得る。以下の例はこれらのフィーチャに関連しており、具体的には、行（次元１）を別のテーブル（例えば、同じアノテーションファイル内のテーブルＮｏ．３）の行にリンクし得る。

```
“Ｌｉｎｋａｇｅｓ”：［｛
  “Ｔｙｐｅ”　：　“ｂｙＩｎｄｅｘ”，
  “ＤｉｍｅｎｓｉｏｎＩｎＣｕｒｒｅｎｔＴａｂｌｅ”：１，
  “Ｔａｂｌｅ”　　：３，
  “ＤｉｍｅｎｓｉｏｎＩｎＬｉｎｋｅｄＴａｂｌｅ”　：１
｝］
```

【０１１６】

［０１３９］属性値によって列（次元２）を別のテーブルの行にリンクする例（現在のテーブルの次元２の属性２が、ファイル２のデータセット４内のテーブル３の次元１の属性５にリンクされる）は、システムプロセッサによって次のようにして実行され得る。

```
“Ｌｉｎｋａｇｅｓ”：［｛
  “Ｔｙｐｅ”　　：“ｂｙＶａｌｕｅ”，
  “ＤｉｍｅｎｓｉｏｎＩｎＣｕｒｒｅｎｔＴａｂｌｅ”：２，
  “ＡｔｔｒｉｂｕｔｅＩｎＣｕｒｒｅｎｔＴａｂｌｅ”：４，
  “Ｄａｔａｓｅｔ”　　：４，
  “ＡｎｎｏｔａｔｉｏｎＦｉｌｅ”　　：２，
  “Ｔａｂｌｅ”　　：３，
  “ＤｉｍｅｎｓｉｏｎＩｎＬｉｎｋｅｄＴａｂｌｅ”　：１，
  “ＡｔｔｒｉｂｕｔｅＩｎＬｉｎｋｅｄＴａｂｌｅ”　：５，
｝］
```

【０１１７】

［０１４０］メタデータ構造は任意の情報ストレージをサポートするため、システムプロセッサは、例えば標準化された形式を使用して、さらに３つ以上のテーブルのリンクにリンケージをさらに拡張し得る（例えば、テーブル３が、テーブル１で使用されている遺伝子ＩＤをテーブル２内の遺伝子名に変換してもよい）。上記の例ではリンケージに特定のＪＳＯＮベースの形式が使用されているが、別の形式が使用されてもよく、例えば、限定はされないが、ＸＭＬが使用されてもよい。

【０１１８】

［０１４１］一実施形態では、属性ベースのリンケージが実施されてもよい。メタデー

タベースのリンケージは高レベルのリンケージに有用である可能性があるが、場合によっては、行／列ごとのリンケージが有益である可能性がある。例えば、複数のサンプルを有するＶＣＦファイルでは、属性ＳｅｑｕｅｎｃｉｎｇＤａｔａｓｅｔＧｒｏｕｐおよびＳｅｑｕｅｎｃｉｎｇＤａｔａｓｅｔを列属性に追加することによって、特定のサンプルに対応するシーケンシングデータがリンクされ得る。そのようなリンケージ属性では、デコンプレッサがリンケージ属性を通常の属性と区別できるよう、メタデータ内で「ＬｉｎｋａｇｅＡｔｔｒｉｂｕｔｅＦｌａｇ」が真に設定されていてもよい。

【０１１９】

［０１４２］場合によっては、システムプロセッサは、ゲノム領域に応じて、アノテーションデータセット間のマッピング動作を実行し得る。これは、例えば、各データセットに別々にインデックスを付けることで実現されてもよい。例えば、ＶＣＦファイル内のある領域に対応するシーケンシングデータを見つけるために、シーケンシングデータのマスターインデックステーブルにアクセスすることで、１つまたは複数の適切なアクセスユニットが決定されてもよい。異なるデータセットについて別々のインデックス付与を用いることにより、各データセットについて、所定の（例えば、最適な）チャンクサイズおよびその他のパラメータを選択できる。さらに、場合によっては、ＡＵクラスが異なるため、バリアントをＡＵに直接リンクできない可能性がある。同様に、複数のサンプルを有するＶＣＦファイルでは、バリアントは複数のデータセットにわたってアクセスユニットにマッピングされ、この情報の保存は大きなストレージを占める可能性がある。一実施形態では、シーケンシングデータのＡＵＩｄまたはバイトオフセットは、ＶＣＦファイル内に行属性として保存され、これは、現在のバリアントに対応するアクセスユニットの高速な検索を可能にする。また、遺伝子に対してリスト型属性を使用することによって、遺伝子がバリアントのリストにマッピングされてもよい。

【０１２０】

［０１４３］一実施形態では、アクセス制御ポリシーは、ＸＡＣＭＬなどの所定の形式を使用してい、ファイルレベル（例えば、図４のアノテーションファイル内）、テーブルレベル（例えば、図１０および図１１の情報内）、またはその両方で指定され得る。特定のユーザはすべてのデータにアクセスできる可能性があるが、他のユーザは粗い解像度のデータにしかアクセスできない可能性がある（異なる解像度が異なるテーブルに保存されていることを思い出されたい）。このタイプのポリシーはファイルレベルで指定され得る。テーブル内の属性に固有のポリシーが、例えばテーブルレベルで指定されてもよい。これは、属性のサブセットのみへのアクセス、または属性の値に基づく特定のチャンクのみへのアクセスを含み得る。別のタイプのポリシーでは、メタデータおよび情報へのアクセスは許可されるが、実際のデータへのアクセスは許可されない可能性がある。

【０１２１】

展開

［０１４４］本明細書に記載される統一および標準化された形式のファイルの選択された部分、またはファイル全体を展開するためのプロセスは、１つまたは複数のタイプのクエリを実行することをまず含み得る。クエリおよび／または展開アルゴリズムは相互に排他的ではない可能性があり、一部の実施形態では組み合わせられてもよい。例えば、メタデータおよび特定の属性の両方が展開されてもよいし、または選択されたチャンクからの選択された属性が展開されてもよい。展開はシステムプロセッサによって、またはシステムプロセッサに結合された別の処理エンティティによって実行され得る。場合によっては、アクセス制御ポリシーによってこれらのクエリの一部が制限されてもよい。これらをサポートするためにアプリケーション・プログラミング・インターフェース（ＡＰＩ）が使用されてもよい。そのようなＡＰＩはＭＰＥＧ－Ｇ　ｐａｒｔ　３と類似していてもよく、または異なるタイプのＡＰＩであってもよい。

【０１２２】

［０１４５］メタデータ／情報クエリ：これらのタイプのクエリは、要求されたテーブル（例えば、解像度レベル）、コンプレッサ、属性、および／またはチャンクに対応する

メタデータおよび情報のみを照会し得る。クエリを実行する際、まず、図４のアノテーションファイルの初めに記載されているようなトップレベルの情報に直接アクセスし得る。テーブル固有メタデータ／属性の詳細は、図４のテーブルのＢｙｔｅＯｆｆｓｅｔを使用して選択的にアクセスされ得る。

【０１２３】

［０１４６］完全データ展開：このタイプの展開は、すべてのテーブルおよび属性を含むデータ全体の展開を含み得る。これは、上記したように、まず、ファイルのトップレベルのメタデータおよびテーブル情報を読み取ることによって実行され得る。次に、圧縮パラメータが展開マネージャにロードされる。そして、テーブルごとに、テーブル情報、次元、および属性が読み取られた後、インデックスが読み取られて、各次元に沿ったチャンクの位置が決定される。次に、各チャンクおよび各属性についてデータペイロードが（シリアルまたはパラレルで）選択的に展開される。別の属性を依存関係／コンテキストとして使用して属性が圧縮されている場合、最初にその別の属性を展開することによって展開が実行されてもよい。属性がＣｏｍｐｒｅｓｓｏｒＣｏｍｍｏｎＤａｔａを使用する場合、チャンクを展開する前にこの情報がロードされてもよい。二次元対称配列の場合（例えば、上記のＳｙｍｍｅｔｒｙＦｌａｇを参照されたい）、対角および下三角行列のみが展開され、対称性を利用して上三角部分が埋められてもよい。

【０１２４】

［０１４７］１つのテーブルのみの展開：このタイプの展開は完全データ展開と類似しているが、要求されたテーブルにジャンプするために（例えば、図４の）アノテーションファイル内のＢｙｔｅＯｆｆｓｅｔフィールドが使用され、そのテーブルのみが展開される点で異なる。

【０１２５】

［０１４８］テーブルの選択された属性の照会：この方法は１つのテーブルのみの展開と類似しているが、要求された属性に対応する情報のみが読み取られる点で異なる。図１２の属性情報構造内のＡｔｔｒｉｂｕｔｅＩｎｆｏＳｉｚｅ変数に基づいて、他の属性はスキップされる。一実施形態では、図２２のＰａｙｌｏａｄ　Ｓｉｚｅ［ｉ］［ｊ］情報に基づいて他の属性をスキップすることによって、要求された属性のみが展開される。属性依存チャンクが使用される場合、所与の属性のすべてのチャンクが、例えば図２０のデータペイロード構造に示されるように、一緒に保存されてもよい。

【０１２６】

［０１４９］配列内の選択されたインデックス範囲のみの照会：この方法は１つのテーブルのみの展開と類似しているが、インデックスがロードされ、チャンク化のタイプ（固定サイズ／可変サイズ）に依存して、要求された範囲と重複するチャンクが決定される点で異なる。また、図１６のチャンク構造テーブル内のＢｙｔｅＯｆｆｓｅｔ情報を使用して、上記の決定されたチャンクのペイロードにジャンプしてもよい。場合によっては、属性依存チャンクが使用されておらず、所与のチャンクのすべての属性が一緒に保存されている場合、方法はより効率的に実施され得る。また、要求されたチャンクが展開され、重複するインデックスのみが返される。ある属性のコンプレッサがチャンク内で効率的なランダムアクセスを可能にする場合、このランダムアクセスは、展開速度をさらに高めるための基礎として利用され得る。これが起こる可能性のあるシナリオの例として、スパース配列またはＧＴＣなどの遺伝子型のための特化コンプレッサが挙げられる。

【０１２７】

［０１５０］特定の属性の値／範囲に基づく照会：この手法は、配列内の選択されたインデックス範囲のみの照会と類似しているが、（例えば、図１５のインデックス構造内に示されている）追加の属性固有インデックスが対象のチャンクについて利用可能である場合、このインデックスを使用して該１つまたは複数のチャンクが決定されてもよい。そのようなインデックスが利用可能でない場合、すべてのチャンクについて対象の属性が展開された後、関連するチャンクが決定されてもよい。追加のインデックスを使用しない場合であっても、すべてのチャンクについて一部の属性のみが展開されるため、展開速度が向

上する可能性がある。一実施形態では、残りの属性は、関連するチャンクについてのみ展開されてもよい。

【０１２８】

フォルダ構造および編集

［０１５１］上記のような、システムプロセッサによるゲノムアノテーションデータの統一および標準化されたファイル形式への処理は、効率、利便性、システム要件の低減、高速照会、およびデータアクセサビリティに関して多くの利点を提供する。特定の用途へのより良い適合のために、実施形態に追加の変形および調整が加えられてもよい。

【０１２９】

［０１５２］例えば、データが単一のマシンに保存され、頻繁に編集される場合、データはファイルマネージャを使用してディレクトリ／フォルダ階層で保存されてもよい。この階層は、ファイル全体を上書きするのではなく、単一のチャンクおよび属性に対応するファイルのみを変更することによって、データの一部を操作することを容易にし得る。編集が完了し、データが送信されるとき、ファイルは単一のファイル形式に変換し直されてもよい。これは、データペイロードサイズに基づいてインデックスを再計算し、フォルダ階層をまとめて１つのファイルに戻すことで実行できる。

【０１３０】

［０１５３］ファイル形式（例えば、図２３Ａ）をフォルダ階層（例えば、図２３Ｂ）に変換し、そして単一のテーブルに戻す方法の例が示されている。この例では、変換は、ファイル形式のテーブルセクション２３１０内の各テーブルをフォルダ階層のフォルダとして扱い、ファイル形式のデータセクション２３２０内のデータの各チャンクをフォルダ階層のフォルダとして扱うことによって実行される（例えば、ＡｔｔｒｉｂｕｔｅＤｅｐｅｎｄｅｎｔＣｈｕｎｋｓフィールドに偽値が保存されていると仮定して）。フォルダ階層において、ブロック２３３１～２３４３はファイルに対応し、ブロック２３２１～２３２６はフォルダに対応する。フォルダ階層において、チャンクは既に個別のフォルダに保存されているため、インデックスは属性固有インデックスを保存するだけでよい。単純な例を図９に示す。

【０１３１】

［０１５４］フォルダ階層として考えると、提案される方式は、用語を次のように変更することによって、上記の関連特許出願で提案されている方式と大まかに関連付けることができる。「属性」：「１つまたは複数の行／列を有する領域」および「チャンク」：「ブロック」。

【０１３２】

［０１５５］上記実施形態のうちの１つまたは複数がいかにして実装され得るかを説明するために、関連する機能を提供しつつ、様々なアノテーションデータを保存するための以下の２つの例、（１）バリアントコールデータ、および（２）ゲノム機能的アノテーションデータについて議論する。

【表１】

バリアントコールデータ

```
##fileformat=VCFv4.0
##fileDate=20090805
##source=myImputationProgramV3.1
##reference=1000GenomesPilot-NCBI36
##phasing=partial
##INFO=<ID=NS,Number=1,Type=Integer,Description="Number of Samples With Data">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Total Depth">
##INFO=<ID=AF,Number=.,Type=Float,Description="Allele Frequency">
##INFO=<ID=AA,Number=1,Type=String,Description="Ancestral Allele">
##INFO=<ID=DB,Number=0,Type=Flag,Description="dbSNP membership, build 129">
##INFO=<ID=H2,Number=0,Type=Flag,Description="HapMap2 membership">
##FILTER=<ID=q10,Description="Quality below 10">
##FILTER=<ID=s50,Description="Less than 50% of samples have data">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=HQ,Number=2,Type=Integer,Description="Haplotype Quality">
#CHROM POS     ID        REF ALT    QUAL FILTER INFO
FORMAT      NA00001        NA00002        NA00003
20     14370   rs6054257 G      A       29   PASS   NS=3;DP=14;AF=0.5;DB;H2
GT:GQ:DP:HQ 0|0:48:1:51,51 1|0:48:8:51,51 1/1:43:5:.,.
20     17330   .         T      A       3    q10    NS=3;DP=11;AF=0.017
GT:GQ:DP:HQ 0|0:49:3:58,50 0|1:3:5:65,3   0/0:41:3
20     1110696 rs6040355 A      G,T     67   PASS   NS=2;DP=10;AF=0.333,0.667;AA=T;DB
GT:GQ:DP:HQ 1|2:21:6:23,27 2|1:2:0:18,2   2/2:35:4
20     1230237 .         T      .       47   PASS   NS=3;DP=13;AA=T
GT:GQ:DP:HQ 0|0:54:7:56,60 0|0:48:4:51,51 0/0:61:2
20     1234567 microsat1 GTCT   G,GTACT 50   PASS   NS=3;DP=9;AA=G
GT:GQ:DP    0/1:35:4       0/2:17:2       1/1:40:3
```

表１：ＶＣＦファイルの例（ＩＧＳＲより）

【０１３３】

［０１５６］上記の表１は、５つのバリアントおよび３つのサンプルを含むＶＣＦファイルのセクションを示す。ＶＣＦファイルはシステムプロセッサ１１０によって、データを保持するとともに、関連付けられた追加の機能を提供しつつ、統一および標準化されたファイル形式に処理され得る。そのようなファイルは以下のフィーチャを有し得る。

【０１３４】

［０１５７］メタデータ：（＃＃で始まる）コメント行はＦｉｌｅＭｅｔａｄａｔａの一部として保持されてもよい。これが、シーケンシングデータとともにＭＰＥＧ－Ｇファイルの一部として保存される場合、メタデータも、このバリアントコールデータに対応するシーケンシングデータを含む、対応するデータセットグループを含み得る。

【０１３５】

［０１５８］トレーサビリティ：トレーサビリティ情報が、シーケンシングデータとともにＭＰＥＧ－Ｇファイルの一部として保存される場合、トレーサビリティ情報は、ＲＡＷシーケンシングデータと、使用されるツールおよびそれらのバージョンとのＵＲＩから開始する、１つまたは複数のバリアントコールの生成のための１つまたは複数のコマンドを含み得る。トレーサビリティ情報は、再現可能な方法でファイルを検証するために使用されてもよい。

【０１３６】

［０１５９］テーブル：バリアントデータが単一の解像度で保存される場合、バリアントデータは、統合されたファイル形式で単一のテーブルに保存されてもよい（ｎＤｉｍｅｎｓｉｏｎｓ＝２）。

【０１３７】
［０１６０］次元属性：第１の次元（バリアント）については、ＣＨＲＯＭ、ＰＯＳ、ＩＤ、ＲＥＦ、ＡＬＴ、ＱＵＡＬ、ＦＩＬＴＥＲ、およびＩＮＦＯフィールドなどの複数の次元属性が存在し得る。コメントに記述されているように、ＩＮＦＯフィールドはＮＳ、ＤＰ、ＡＦなどの複数の属性に分割され得る。これらの属性のタイプもコメントフィールド内で言及されていてもよい。これらをグループ化するために属性メタデータが使用されてもよい（例えば、ＮＳ、ＤＰ、ＡＦはグループＩＮＦＯに属し得る）。デフォルト値は属性に依存し、例えば、ＦＩＬＴＥＲ属性の場合、デフォルト値は「ＰＡＳＳ」に設定され得る。
【０１３８】
［０１６１］第２の次元（サンプル）については、サンプル名（例えば、ＮＡ００００１）がオリジナルのＶＣＦファイル内に存在する唯一の属性である可能性がある。シーケンシングデータへのリンケージをサポートするために、追加の属性、例えば、このサンプルに対応するシーケンシングデータを含むデータセットグループおよびデータセットが追加されてもよい。特定の数量、例えば、特定のバリアントに対応するカウントや平均数量などへの高速アクセスをサポートするために、追加の次元属性が追加されてもよい。コメント内のＩＮＦＯ属性の記述がＡｔｔｒｉｂｕｔｅＭｅｔａｄａｔａの一部として保存されてもよい。
【０１３９】
［０１６２］２Ｄテーブル属性：これらの属性は、それぞれが二次元配列であるＧＴ、ＧＱ、ＤＰなどのＦＯＲＭＡＴフィールド内に記述されている。これらの属性のタイプもコメント内に記述されていてもよい。ほとんどのバリアントが表現されていない場合、ＧＴ属性のデフォルト値は例えば０／０に設定されてもよい。コメント内のこれらの属性の記述がＡｔｔｒｉｂｕｔｅＭｅｔａｄａｔａの一部として保存されてもよい。
【０１４０】
［０１６３］コンプレッサ：属性のためのコンプレッサは、属性のタイプおよび特性に基づいて選択されてもよい。例えば、ＣＨＲＯＭは列挙ベースのスキームの後にｇｚｉｐを使用して圧縮され、ＰＯＳは差分符号化の後にｇｚｉｐを使用して圧縮され得る。サンプル名（ＮＡ０００００１など）は、例えば、トークン化ベースの文字列コンプレッサを使用して効率的に圧縮され得る。一部のＩＮＦＯフィールドは少数のバリアントのためにのみ存在する可能性があるため、スパース表現で符号化されてもよい。同様に、遺伝子型（ＧＴ）は、スパース表現、または遺伝子型専用のコンプレッサ（例えば、ＧＴＣ）を用いて符号化されてもよい。
【０１４１】
［０１６４］特定の可変長属性の長さは、１つまたは複数の他の属性に依存し得る。例えば、ＡＦ（アレル頻度）属性の長さは、変異アレルの数に等しい可能性がある。そのような場合、コンプレッサのｎＤｅｐｅｎｄｅｎｃｉｅｓは１に設定され、この依存関係を利用して圧縮が強化され得る。同様に、両者間の依存関係を利用して、他のＦＯＲＭＡＴフィールドの圧縮のための副次的情報としてＧＴフィールドの値が使用されてもよい。
【０１４２】
［０１６５］チャンク化およびインデックス付与：メイン２Ｄ配列のチャンク化はアクセスパターンに依存して実行されてもよい。ほとんどのアクセスが特定の領域内のバリアントに対するものである場合、各チャンクはすべてのサンプルと少数のバリアントとを含み得る（例えば、水平チャンク）。ほとんどのアクセスが特定のサンプルのすべてのバリアントに対するものである場合、チャンクはすべてのバリアントと少数のサンプルとを含み得る（例えば、垂直チャンク）。両方のタイプのクエリが頻繁に生じる場合、場合によっては、少数のバリアントとサンプルとを含む矩形チャンクを使用する方が良い可能性がある。チャンクのサイズを大きくすることにより、ランダムアクセスのパフォーマンスが圧縮率とトレードオフされる可能性がある。
【０１４３】

［０１６６］ゲノム領域に基づくランダムアクセスの場合、表２に示されるように追加のインデックスが使用されてもよい（例えば、ＣＳＩインデックス付与に基づく）。

【表２】

ＡｔｔｒｉｂｕｔｅＩＤｓＩｎｄｅｘｅｄ	ＣＨＲＯＭ，ＰＯＳ
ＩｎｄｅｘＴｙｐｅ	ＣＳＩ
ＩｎｄｅｘＳｉｚｅ	インデックスのサイズ
ＩｎｄｅｘＤａｔａ	ＣＳＩインデックス構造

【０１４４】

［０１６７］ＣＳＩで行われるように実際のファイル位置を指定する代わりに、対象のゲノム領域と重複するチャンクＩＤのリストが返されてもよい（または示されてもよい）。そして、デフォルトインデックス構造から、ファイル内のこれらのチャンクの位置が決定され得る。Ｉｎｄｅｌバリアントまたは構造バリアントが一般的である場合、バリアントのＳＴＡＲＴ位置およびＥＮＤ位置の両方に基づいてＣＳＩインデックス付与が行われ得る。高速ランダムアクセス照会を可能にするために、より多くの属性にインデックスが付けられてもよい。例えば、ＦＩＬＴＥＲ属性にインデックスを付けることで、ＦＩＬＴＥＲ＝ＰＡＳＳであるか否かに基づくバリアントのより高速なフィルタリングが可能にされてもよい。

【０１４５】

［０１６８］保護：アクセス制御ポリシーは、例えばユースケースに応じて、様々な形態をとることができる。例えば、特定のユーザはすべてのデータにアクセスできるが、他のユーザは特定のゲノム領域内のバリアントにのみアクセスできる場合がある（例えば、ＣＨＲＯＭおよびＰＯＳによって指定される）。同様に、アクセスは特定のサンプルのみに制限されてもよい。あるケースでは、これは、チャンクがそれに従って選択されることを要する可能性がある。アクセス制御は属性レベルで課されてもよく、例えば、ＩＮＦＯフィールドへのアクセスは許可されるが、個別のサンプルデータへのアクセスは許可されない。

【表３】

ゲノム機能的アノテーションデータ（ＢＥＤ）

```
browser position chr7:127471196-127495720
browser hide all
track name="ItemRGBDemo" description="Item RGB demonstration" visibility=2
itemRgb="On"
chr7    127471196  127472363  Pos1  0  -  127471196  127472363  255,0,0
chr7    127472363  127473530  Pos2  0  -  127472363  127473530  255,0,0
chr7    127473530  127474697  Pos3  0  -  127473530  127474697  255,0,0
chr7    127474697  127475864  Pos4  0  -  127474697  127475864  255,0,0
chr7    127475864  127477031  Neg1  0  -  127475864  127477031  0,0,255
chr7    127477031  127478198  Neg2  0  -  127477031  127478198  0,0,255
chr7    127478198  127479365  Neg3  0  -  127478198  127479365  0,0,255
chr7    127479365  127480532  Pos5  0  -  127479365  127480532  255,0,0
chr7    127480532  127481699  Neg4  0  -  127480532  127481699  0,0,255
```

表３：簡単なＢＥＤファイルの例（ＵＣＳＣゲノムブラウザＦＡＱより）

【０１４６】

［０１６９］上記の表３は、アノテーションデータとともにＢＥＤファイルのセクショ

ンを示す。システムプロセッサは、後述されるように、データを保持するとともに、追加の機能を提供しつつ、統一および標準化されたファイルに準拠するようにこのファイル内の情報を処理し得る。

【０１４７】

［０１７０］メタデータ：コメント行（例えば、最初の３つの行）はＦｉｌｅＭｅｔａｄａｔａの一部として保持されてもよい。これが、シーケンシングデータとともにＭＰＥＧ－Ｇファイルの一部として保存される場合、メタデータも、このアノテーションデータに対応するシーケンシングデータを含む、対応するデータセットグループを含み得る。

【０１４８】

［０１７１］テーブル：データを異なるスケールや解像度で表示できるようにするために、システムプロセッサによって、異なる解像度のための事前に計算された値とともに、複数のテーブルが生成されてもよい。例えば、ＴａｂｌｅＩｎｆｏフィールドは、解像度を示すパラメータおよび他の情報を事前定義された形式で保存してもよい。これにより、ユーザはファイル全体を読み取ることなく、利用可能な解像度のリストを照会できる可能性がある。また、テーブルごとのＢｙｔｅＯｆｆｓｅｔ変数を使用することにより、所望の解像度に直接アクセスできる可能性がある。複数のテーブルのうちの１つまたは複数は、例えば、単一の次元を有し得る。

【０１４９】

［０１７２］属性：一実施形態では、各列は次のような属性として機能することができる：ｃｈｒｏｍ（文字列）、ｃｈｒｏｍＳｔａｒｔ（整数）、ｃｈｒｏｍＥｎｄ（整数）、ｎａｍｅ（文字列）、ｓｃｏｒｅ（整数）、ｓｔｒａｎｄ（文字）、ｔｈｉｃｋＳｔａｒｔ（整数）、ｔｈｉｃｋＥｎｄ（整数）、ｉｔｅｍＲＧＢ（長さ３の８ビット整数配列）。

【０１５０】

［０１７３］コンプレッサ：属性のためのコンプレッサは、属性のタイプおよび特性に基づいて選択されてもよい。例えば、ｃｈｒｏｍは列挙ベースのスキームの後にｇｚｉｐまたはランレングス圧縮を使用して圧縮されてもよく、ｃｈｒｏｍＳｔａｒｔおよびｃｈｒｏｍＥｎｄは、差分符号化の後にｇｚｉｐを使用して圧縮されてもよい。ｔｈｉｃｋＳｔａｒｔおよびｔｈｉｃｋＥｎｄの値がｃｈｒｏｍＳｔａｒｔおよびｃｈｒｏｍＥｎｄに近い場合、例えば、これらの値を副次的情報として使用することによって圧縮が改善される可能性がある。

【０１５１】

［０１７４］この例では、ｃｈｒｏｍＳｔａｒｔの値は前の行のｃｈｒｏｍＥｎｄの値と合致する。これを利用する１つの方法は、ｃｈｒｏｍＳｔａｒｔおよびｃｈｒｏｍＥｎｄを「長さ２の整数配列」タイプの単一の属性と見なすことである。これは、例えば、視覚化ツールがこの代替表現を理解する場合に実行され得る。

【０１５２】

［０１７５］チャンク化およびインデックス付与：ゲノム領域に基づくランダムアクセスの場合、表４に示されるように追加のインデックスが使用されてもよい（ＣＳＩインデックス付与に基づき）。ＣＳＩで行われるように実際のファイル位置を指定する代わりに対象のゲノム領域と重複するチャンクＩＤのリストが示されてもよい。そして、デフォルトインデックス構造から、ファイル内のこれらのチャンクの位置が決定され得る。

【表４】

ＡｔｔｒｉｂｕｔｅＩＤｓＩｎｄｅｘｅｄ	ｃｈｒｏｍ，ｃｈｒｏｍＳｔａｒｔ，ｃｈｒｏｍＥｎｄ
ＩｎｄｅｘＴｙｐｅ	ＣＳＩ
ＩｎｄｅｘＳｉｚｅ	インデックスのサイズ
ＩｎｄｅｘＤａｔａ	ＣＳＩインデックス構造

【０１５３】
［０１７６］ＣＳＩで行われるように実際のファイル位置を指定する代わりに対象のゲノム領域と重複するチャンクＩＤのリストが示されてもよい。そして、デフォルトインデックス構造から、ファイル内のこれらのチャンクの位置が決定され得る。
【０１５４】
［０１７７］保護：また、アクセス制御ポリシーはユースケースに応じて様々な形態をとることができる。例えば、特定のユーザはすべてのデータにアクセスできる可能性があるが、他のユーザは粗い解像度のデータにしかアクセスできない可能性がある（例えば、異なる解像度が異なるテーブルに保存され得ることを思い出されたい）。同様に、アクセスは特定のゲノム領域のみに制限されてもよい。この場合、チャンクがそれに従って選択されてもよい。
【０１５５】
シングルセルＲＮＡｓｅｑ発現データ
［０１７８］シングルセルＲＮＡｓｅｑ発現データは、整数／ｆｌｏａｔ発現値のスパース二次元行列を含み得る。各行は遺伝子に対応し、各列は細胞を表すバーコードに対応する。この場合、発現値は二次元のスパース属性配列として保存され、一方、遺伝子に関連する情報は次元固有の行属性になり、バーコードに関連する情報は次元固有の列属性になる。スパース配列は座標ストリームと値ストリームに分割されてもよい。これらは別々に圧縮され、行座標および各行の列座標は圧縮前に差分符号化される。高速ランダムアクセスを可能にするために、チャンクあたり固定数の遺伝子（行）にデータがチャンク化されてもよい。最後に、遺伝子ｉｄに基づく選択的照会を実行するために追加のＢツリーインデックスが使用されてもよい。このインデックスは、遺伝子ｉｄを、遺伝子ｉｄおよびチャンク内の遺伝子ｉｄの位置を含むチャンクにマッピングする。
【０１５６】
［０１７９］この手法をＥ１８マウスの１万個の脳細胞からなるデータセットに適用した。約３１，０００個の遺伝子、６，８００，０００個のバーコード、および４，０００万個の整数エントリがスパース配列に含まれていた。最終圧縮レイヤーとしてＢＳＣを用いた提案される手法は、７５０ＭＢ（未圧縮）から６７ＭＢにサイズを縮小する。これは、列を属性に分割したり、座標の差分符号化を行うことなく、オリジナルのファイルにｇｚｉｐまたはＢＳＣを直接適用した場合の圧縮後サイズの２分の１未満である。
【０１５７】
［０１８０］本明細書に記載の方法、プロセス、および／または動作は、コンピュータ、プロセッサ、コントローラ、または他の信号処理デバイスによって実行されるコードもしくは命令によって実行され得る。コードまたは命令は、１つまたは複数の実施形態に従って、非一時的なコンピュータ可読媒体に保存されてもよい。方法（または、コンピュータ、プロセッサ、コントローラ、もしくは他の信号処理デバイスの動作）の基礎をなすアルゴリズムが詳細に説明されているので、方法の実施形態の動作を実装するためのコードまたは命令は、コンピュータ、プロセッサ、コントローラ、または他の信号処理デバイスを、本明細書に記載の方法を実行するための専用プロセッサに変換することができる。
【０１５８】

［０１８１］本明細書に開示される実施形態のプロセッサ、コンプレッサ、デコンプレッサ、マネージャ、セレクタ、パーサ、ならびに他の情報生成、処理、および計算フィーチャは、例えば、ハードウェア、ソフトウェア、または両方を含み得る論理で実装され得る。少なくとも部分的にハードウェアで実装される場合、プロセッサ、コンプレッサ、デコンプレッサ、マネージャ、セレクタ、パーサ、およびその他の情報生成、処理、および計算フィーチャは、例えば、様々な集積回路、例えば、限定はされないが、特定用途向け集積回路、フィールドプログラマブルゲートアレイ、論理ゲートの組み合わせ、システムオンチップ、マイクロプロセッサ、または別のタイプの処理もしくは制御回路のうちのいずれかであり得る。

【０１５９】

［０１８２］少なくとも部分的にソフトウェアで実装される場合、プロセッサ、コンプレッサ、デコンプレッサ、マネージャ、セレクタ、パーサ、およびその他の情報生成、処理、および計算フィーチャは、例えば、コンピュータ、プロセッサ、マイクロプロセッサ、コントローラ、または他の信号処理デバイスなどによって実行されるコードまたは命令を保存するためのメモリまたは他のストレージデバイスを含み得る。方法（または、コンピュータ、プロセッサ、マイクロプロセッサ、コントローラ、もしくは他の信号処理デバイスの動作）の基礎をなすアルゴリズムが詳細に説明されているので、方法の実施形態の動作を実装するためのコードまたは命令は、コンピュータ、プロセッサ、コントローラ、または他の信号処理デバイスを、本明細書に記載の方法を実行するための専用プロセッサに変換することができる。

【０１６０】

［０１８３］上記の説明から、本発明の様々な例示的実施形態がハードウェアまたはファームウェアで実装され得ることは明らかであろう。また、様々な例示的実施形態は、機械可読記憶媒体上に記憶された命令として実装されてもよい。これらの命令は、本明細書において詳細に説明される動作を実行するために少なくとも１つのプロセッサによって読み取られ、実行され得る。機械可読記憶媒体は、パーソナルもしくはラップトップコンピュータ、サーバ、または他のコンピューティングデバイスなどの機械によって読み取り可能な形態で情報を保存するための任意のメカニズムを含み得る。したがって、機械可読記憶媒体は、リードオンリーメモリ（ＲＯＭ）、ランダムアクセスメモリ（ＲＡＭ）、磁気ディスク記憶媒体、光学記憶媒体、フラッシュメモリデバイス、および同様な記憶媒体を含むことができる。

【０１６１】

［０１８４］当業者は、本明細書に示されるブロック図が、本発明の原理を具現化する例示的回路の概念図であることを理解するであろう。同様に、あらゆるフローチャート、フロー図、状態遷移図および疑似コードなどは、機械可読媒体内に実体的に表され、よって（コンピュータまたはプロセッサが明示的に示されているか否かに関わらず）コンピュータまたはプロセッサによって実行され得る様々なプロセスを表す。

【０１６２】

［０１８５］様々な例示的実施形態が、それらの例示的実施形態の特定の例示的態様に特に言及して詳細に説明されているが、本発明は他の実施形態も可能であり、その細部は様々な明白な点において変更可能であることを理解されたい。１つまたは複数の実施形態を１つまたは複数の他の実施形態と組み合わせることで新たな実施形態が形成されてもよい。当業者には容易に明らかであるように、本発明の趣旨および範囲から逸脱することなく変形および変更を加えることが可能である。したがって、上記の開示、記載、および図面は説明のみを目的としており、請求項によってのみ定義される本発明をいかようにも制限しない。

【０１６３】

［０１８６］１つまたは複数の実施形態によれば、ゲノムアノテーションデータの選択的圧縮および展開を制御するためのシステムおよび方法は、複数の互換性のないファイル形式のうちの１つのファイル形式の情報を処理して、選択的圧縮および展開を制御する統

合されたファイルにすることを含む。高速クエリ、ランダムアクセス、複数の解像度（ズーム）、選択的暗号化、認証、アクセス制御、およびトレーサビリティなどの機能をサポートするように、アノテーションデータが処理され、情報が抽出されおよびさらに処理される。また、処理は、データの複数の異なる属性を分離し、これらに特化したコンプレッサを使用できるようにすることによって、大幅な圧縮ゲインを可能にするように実行されてもよい。アノテーションに関連するシーケンシングデータへのメタデータおよびリンケージ、ならびに同じ調査からの他のアノテーションデータへのリンケージをサポートするために追加の処理が実行されてもよい。これは、シーケンシングデータのための既存のＭＰＥＧ－Ｇファイル形式とのシームレスな統合を可能にする。

【０１６４】

［０１８７］１つまたは複数の実施形態では、きめ細かいセキュリティ設定を可能にする、階層内の複数のレベルにおける保護（アクセス制御）情報を生成することために追加の処理が実行されてもよい。同様に、メタデータおよび属性は、複数の異なるタイプのアノテーションデータやシーケンシングデータセットを効果的にリンクすることを可能にする。処理によって生成されたファイルは、スタンドアロンファイルとして、またはＭＰＥＧ－Ｇファイルの一部として使用され得る。また、そのようなファイル、特にゲノムアノテーションデータのためのそのようなファイルの生成は、対象の属性に適した圧縮技術を組み込むことにより、様々なデータ型において最高水準の圧縮パフォーマンスを達成するのに十分な柔軟性を提供する。

【図面】

【図１】

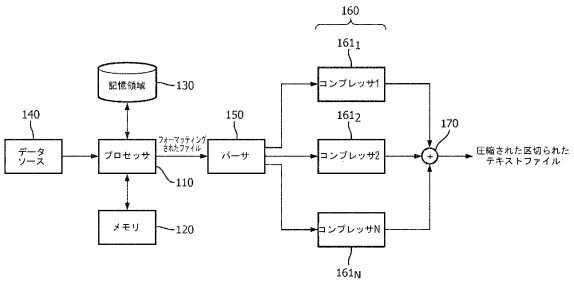

図１

【図２】

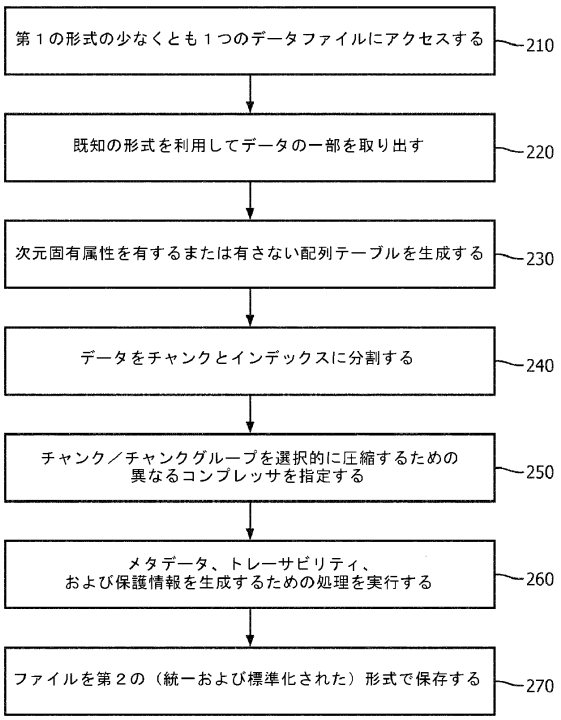

図２

【図３】

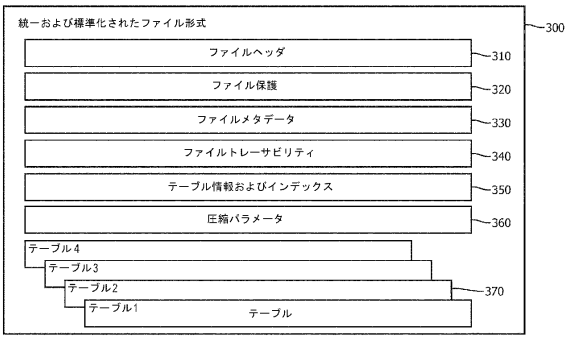

図３

【図４】

アノテーションファイル

	フィールド	簡単な説明	タイプ
310	File header		ファイルヘッダ(§T2)
320	File protection info	アクセス制御ポリシー	ゲノム情報(§T3)
330	File metadata	メタデータ／リンケージ	ゲノム情報(§T3)
340	File traceability info	データの生成に使用されるコマンド	ゲノム情報(§T3)
351	nTables	ファイルに保存されているテーブルの数（例えば複数の解像度）	
	For i in 1 ...nTables:		
352	Table ID [i]	一意のテーブル識別子	整数
353	TableInfo [i]	テーブル情報（例えば解像度）	ゲノム情報(§T3)
354	ByteOffset [i]	ファイル内のテーブル i のバイトオフセット	整数
361	nCompressors	後に保存される様々な属性において使用される異なるコンプレッサの数	整数
	For i in nCompressors:		
362	Compressor [i]		コンプレッサ情報(§T4)
	For i in 1 ...nTables [i]		
	Table [i]		テーブル(§T5)

図４

【図５】

ファイルヘッダ 310

	フィールド	簡単な説明	タイプ
311	File name		文字列
312	File type	例えば「バリアント」、「遺伝子発現」など	文字列
313	File version	更新を記録するため	文字列

図５

【図６】

一般的情報構造 320

	フィールド	簡単な説明	タイプ
321	Payload size	これをスキップすることを可能にするため	整数
322	Payload	所定のコンプレッサ（例えば７ｚ）で圧縮	バイト

図６

【図７】

コンプレッサ情報構造 360

	フィールド	簡単な説明	タイプ
363	CompressorID	一意のコンプレッサ識別子	文字列
364	nDependencies	他の属性の値に基づいて属性を圧縮することを可能にするため	整数
365	CompressorNameList	コンプレッサ名のリスト（または「埋め込み」）	リスト（文字列）
366	CompressorParametersList	展開に必要なパラメータ	リスト（ゲノム情報）

図７

【図８】

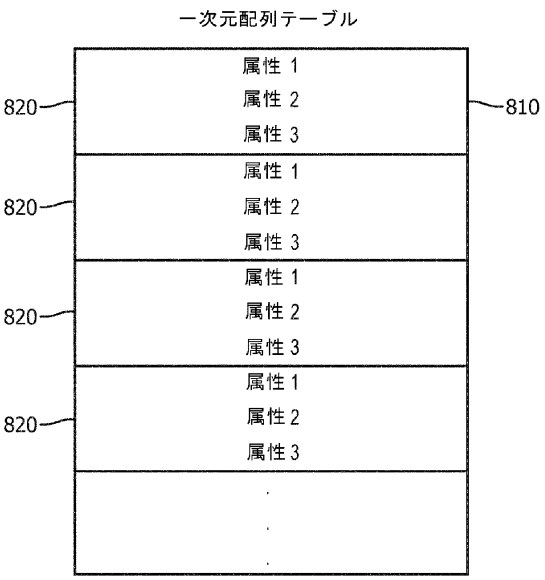

図８

【図９】

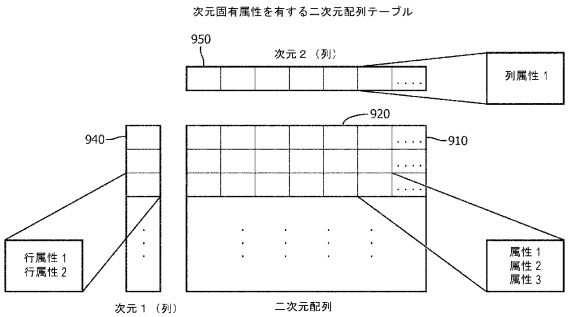

図９

【図１０】

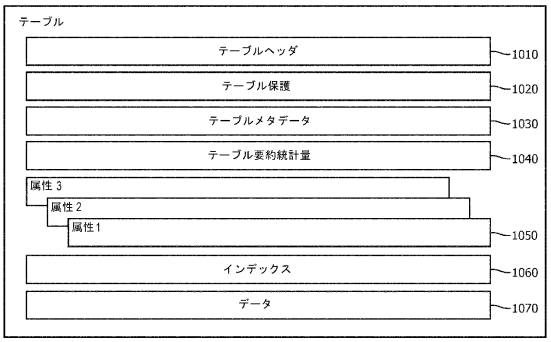

図１０

【図１１】

フィールド		簡単な説明	タイプ
TableID	1105	§Ｔ１と同じ	ゲノム情報(§T3)
TableInfo	1110	§Ｔ１と同じ	ゲノム情報(§T3)
Table protection	1020	アクセス制御ポリシー	ゲノム情報(§T3)
Table metadata	1030	メタデータ／リンケージ	ゲノム情報(§T3)
Summary statistics	1040	例えばカウント、平均値 1140	リスト(キー値)
nDimensions	1150	次元の数	整数
For i in 1 ... nDimensions:			
Size [i]		次元ｉのサイズ	整数
Dimension name [i]			
Dimension metadata [i]		メタデータ／リンケージ	ゲノム情報(§T3)
If nDimensions == 2:			
Symmetry flag		二次元配列が対称な場合は真	ブール
n AttributesMain	1160		
For i in 1 ... nAttributesMain:			
Attribute info main [i]			属性情報(§T6)
Byte offset main		Ｉｎｄｅｘ ｍａｉｎのバイトオフセット	整数
If n dimensions > 1:		次元固有属性	
For i in 1 ... nDimensions:			
nAttributes dim [i]		次元固有属性の数	整数
For j in 1 ... nAttributes dim [i]			
Attribute info dim [i][j]			属性情報(§T6)
Byte offset dim [i]		Ｉｎｄｅｘ ｄｉｍ［ｉ］のバイトオフセット	整数
Index main			インデックス(§T7)
Data payloads main			データ(§T9)

図１１

【図１２】

属性情報構造

フィールド		簡単な説明	タイプ
AttributeInfoSize	1205	構造をスキップすることを可能にするため	整数
Attribute ID	1210	一意の属性識別子	整数
AttributeName	1215		文字列
AttributeMetadata	1220	属性のメタデータ／リンケージ／グループ化	ゲノム情報(§T3)
Attribute type	1225	基本型（例えば整数、文字、ｆｌｏａｔ、文字列）または派生型（例えば固定長、可変長配列）	文字列
DefaultValue	1230	ほとんどの値がデフォルトと合致する場合のスパース符号化のため	属性タイプ
Summary statistics	1235	例えばカウント、平均値	リスト(キー値)
CompressorID	1240	この属性に使用されるコンプレッサ	整数
For i in 1 ... nDependencies:		§Ｔ４で定義されているｎＤｅｐｅｎｄｅｎｃｉｅｓ	
If nDimensions > 1:			
Dimension		これがメインｎ次元テーブルの属性である場合、これは、どの次元が依存関係属性を含むかを示す（依存関係属性がメインｎＤｉｍｅｎｓｉｏｎａｌテーブルにもある場合はｎＤｉｍｅｎｓｉｏｎｓ＋１に設定）	整数
AttributeID		依存関係を含むＡｔｔｒｉｂｕｔｅＩＤ	整数
CompressorCommonData size			整数
CompressorCommonData		すべてのチャンクに共通するコンプレッサのコードブック／統計モデルを保存するため	バイト

図１２

【図１３】

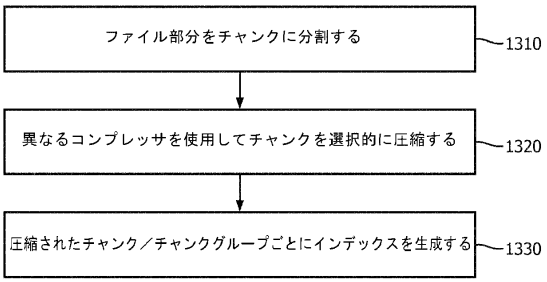

図１３

【図１４】

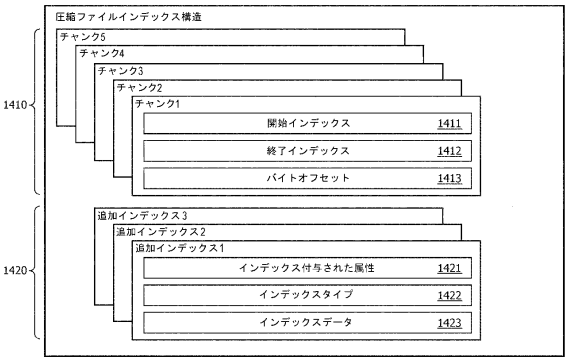

図１４

【図１５】

インデックス構造

フィールド	簡単な説明	タイプ
AttributeDependentChunks 1510	チャンクサイズが属性に依存しているか、またはすべての属性に同じチャンク化が使用されているかを示すフラグ	ブール
If not AttributeDependentChunks:		
ChunksStructure 1515		
Else:		チャンク(§T8)
For i in 1 ... nAttributes:		
Chunks structure [i] 1520		
// Additional attribute specific indexes 1525		
n Additional indexes	特定の属性（例えば染色体、位置）に基づくより高速な照会のための追加インデックスの数 これらは、所望の照会結果を含むチャンク番号を返す	整数
For i in 1 ... nAdditionalIndexes:		
AttributeIDsIndexed [i] 1530	インデックス付与された属性のリスト	リスト(整数)
IndexType [i] 1535	インデックスタイプ（例えば、染色体およびゲノム位置のためのＣＳＩインデックス、またはデータベース型クエリのためのＢツリー）	文字列
IndexSize [i] 1540	インデックスをスキップすることを可能にするため	整数
IndexData [i] 1545	実際のインデックスデータ、詳細はＩｎｄｅｘＴｙｐｅ［ｉ］に依存する	バイト

図１５

【図１６】

チャンク構造

フィールド	簡単な説明	タイプ
nChunks 1605	チャンクの数	整数
VariableSizeChunks 1610	チャンクサイズが可変か固定かを示すフラグ （各次元の境界を除く）	ブール
If VariableSizeChunks:		
For j in 1 ... nChunks:		
For k in 1 ... nDimensions:		
StartIndex [j][k] 1615	次元ｋ沿いのチャンクの開始位置	整数
EndIndex [j][k] 1620	次元ｋ沿いのチャンクの終了位置	整数
ByteOffset [j] 1625	ファイル内のチャンクｊのバイトオフセット	整数
Else:		
For k in 1 ... nDimensions:		
ChunkSize [k]	固定サイズのチャンクの場合、各次元でチャンクのサイズを格納するのに十分	整数
For j in 1 ... nChunks:		
ByteOffset [j]	ファイル内のチャンクｊのバイトオフセット	整数

図１６

【図１７】

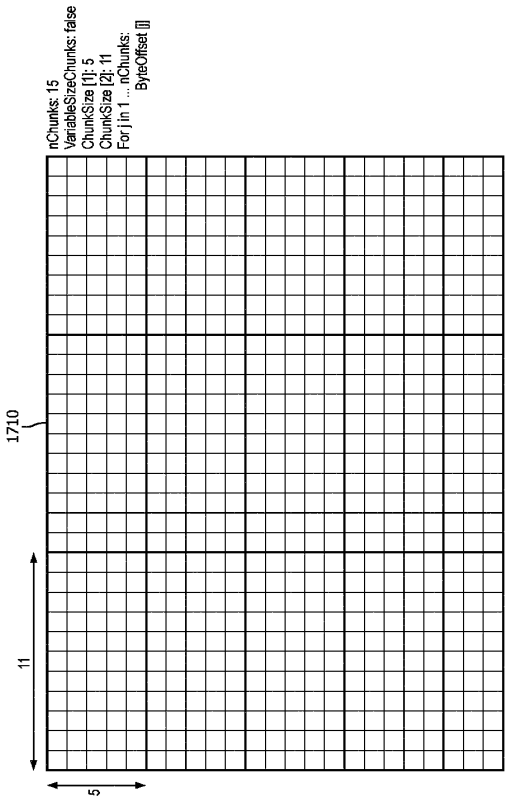

FIG. 17

【図１８】

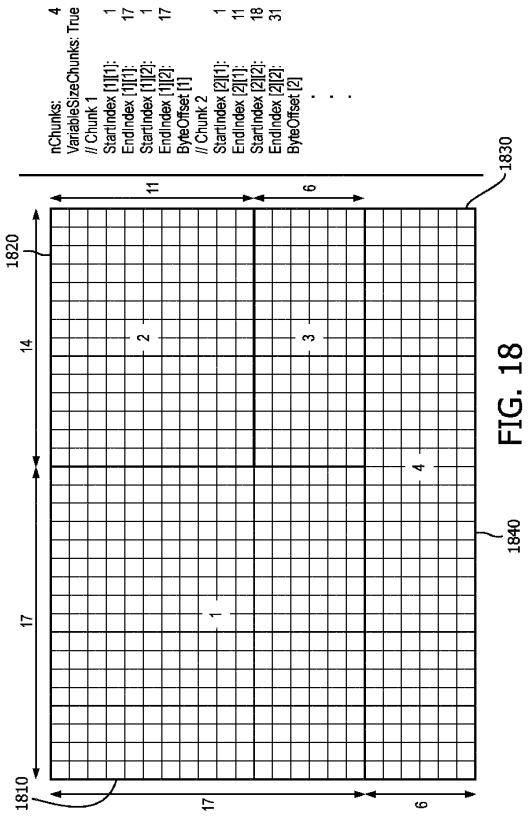

FIG. 18

【図１９】

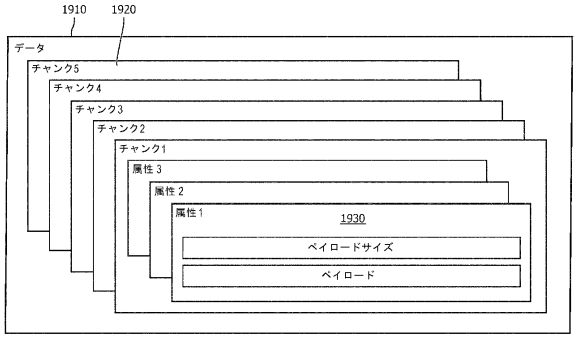

図１９

【図２０】

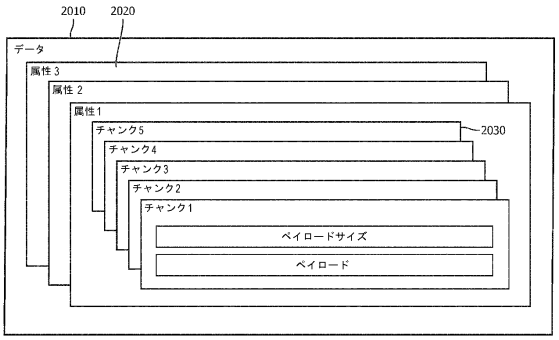

図２０

【図２１】

データペイロード

フィールド	簡単な説明	タイプ
If not AttributeDependentChunks:		
For i in 1... nChunks:		
For j in 1 ... nAttributes:		
Payload size [i][j]	特定の属性をスキップすることを可能にするため	整数
Payload [i][j]	圧縮されたペイロード	バイト
Else:		
For j in 1 ... nAttributes:		
For i in 1... nChunks:		
Payload size [j][i]		整数
Payload [i][j]	圧縮されたペイロード	バイト

図２１

【図２２】

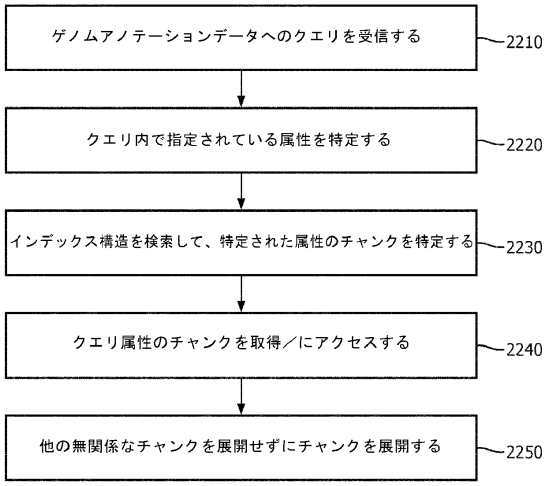

図２２

【図２３Ａ】

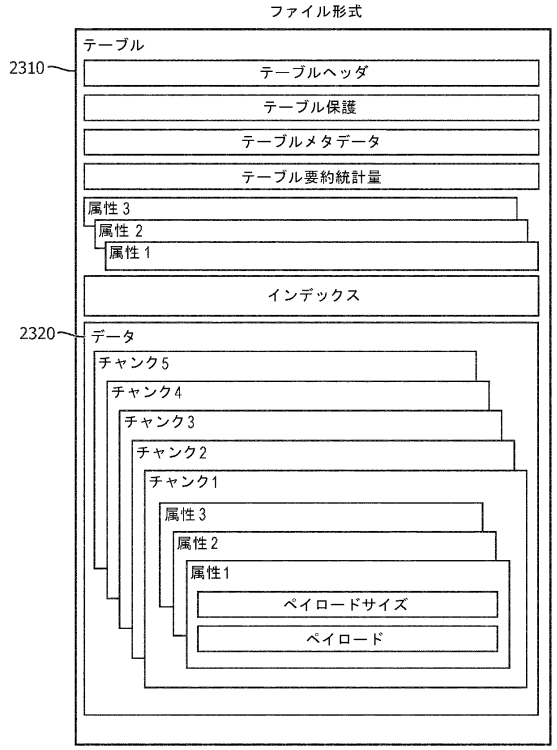

図２３Ａ

【図２３Ｂ】

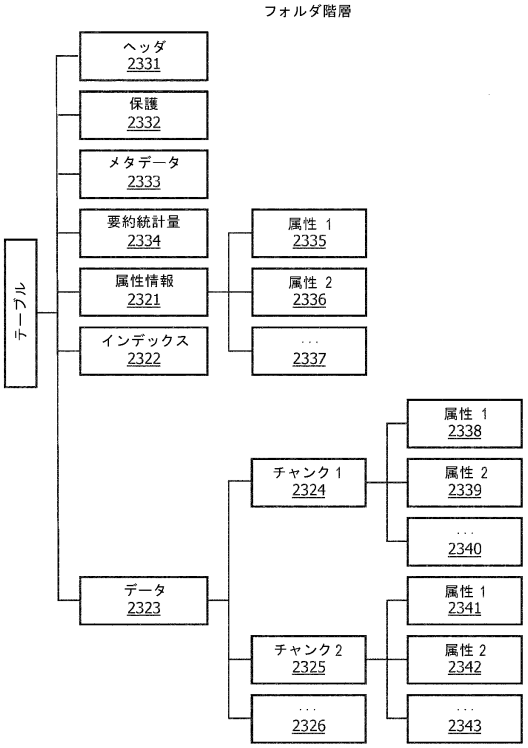

図２３Ｂ

フロントページの続き

(33)優先権主張国・地域又は機関
米国(US)
フィリップス インターナショナル ビー． ヴィ． インテレクチュアル プロパティー アンド スタンダーズ
(72)発明者 チャン イー ヒム
オランダ国 ５６５６ アーエー アインドーフェン ハイ テック キャンパス ５ フィリップス インターナショナル ビー． ヴィ． インテレクチュアル プロパティー アンド スタンダーズ
審査官 渡邉 加寿磨
(56)参考文献 米国特許出願公開第２０１３／００３１０９２（ＵＳ，Ａ１）
特表２０１９－５３７７８１（ＪＰ，Ａ）
米国特許出願公開第２０１８／００８９３６９（ＵＳ，Ａ１）
国際公開第２０１８／０７１０５５（WO，Ａ１）
(58)調査した分野 (Int.Cl.，ＤＢ名)
Ｇ１６Ｂ ５／００－９９／００
Ｇ０６Ｆ １２／００
Ｇ０６Ｆ １６／００－１６／９５８
Ｈ０３Ｍ ７／３０