US 20230357842A1

(19) United States

(12) Patent Application Publication (10) Pub. No.: US 2023/0357842 A1

Locke et al. (43) Pub. Date: Nov. 9, 2023

(54) SYSTEMS AND METHODS FOR MITOCHONDRIAL ANALYSIS

(71) Applicant: **Seven Bridges Genomics Inc.**, Charlestown, MA (US)

(72) Inventors: **Devin Locke**, Medford, MA (US); **Piotr Szamel**, Cambridge, MA (US)

(73) Assignee: **Seven Bridges Genomics Inc.**, Charlestown, MA (US)

(21) Appl. No.: **18/132,353**

(22) Filed: **Apr. 7, 2023**

Related U.S. Application Data

(63) Continuation of application No. 16/798,759, filed on Feb. 24, 2020, now Pat. No. 11,649,495, which is a continuation of application No. 15/014,483, filed on Feb. 3, 2016, now Pat. No. 10,584,380.

(60) Provisional application No. 62/212,886, filed on Sep. 1, 2015.

Publication Classification

(51) **Int. Cl.**

C12Q 1/6874	(2006.01)
C12Q 1/6888	(2006.01)
G16B 20/00	(2006.01)
G16B 30/10	(2006.01)

(52) **U.S. Cl.**

CPC ***C12Q 1/6874*** (2013.01); ***C12Q 1/6888*** (2013.01); ***G16B 20/00*** (2019.02); ***G16B 30/10*** (2019.02); *C12Q 2600/156* (2013.01)

(57) **ABSTRACT**

The invention provides methods of analyzing an individual's mtDNA by transforming available reference sequences into a directed graph that compactly represents all the information without duplication and comparing sequence reads from the mtDNA to the graph to identify the individual or describe their mtDNA. A directed graph can represent all of the genetic variation found among the mitochondrial genomes across all of a number of reference organisms while providing a single article to which sequence reads can be aligned or compared. Thus any sequence read or other sequence fragment can be compared, in a single operation, to the article that represents all of the reference mitochondrial sequences.

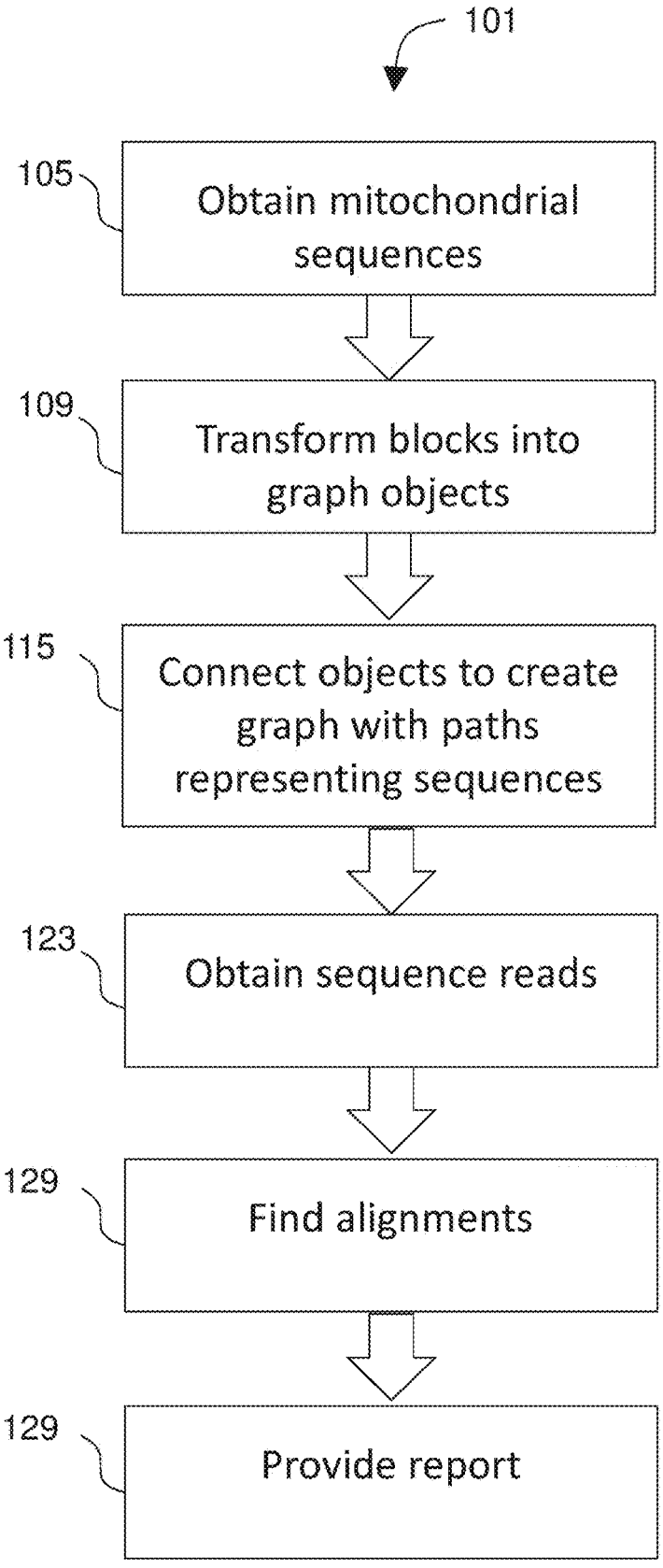
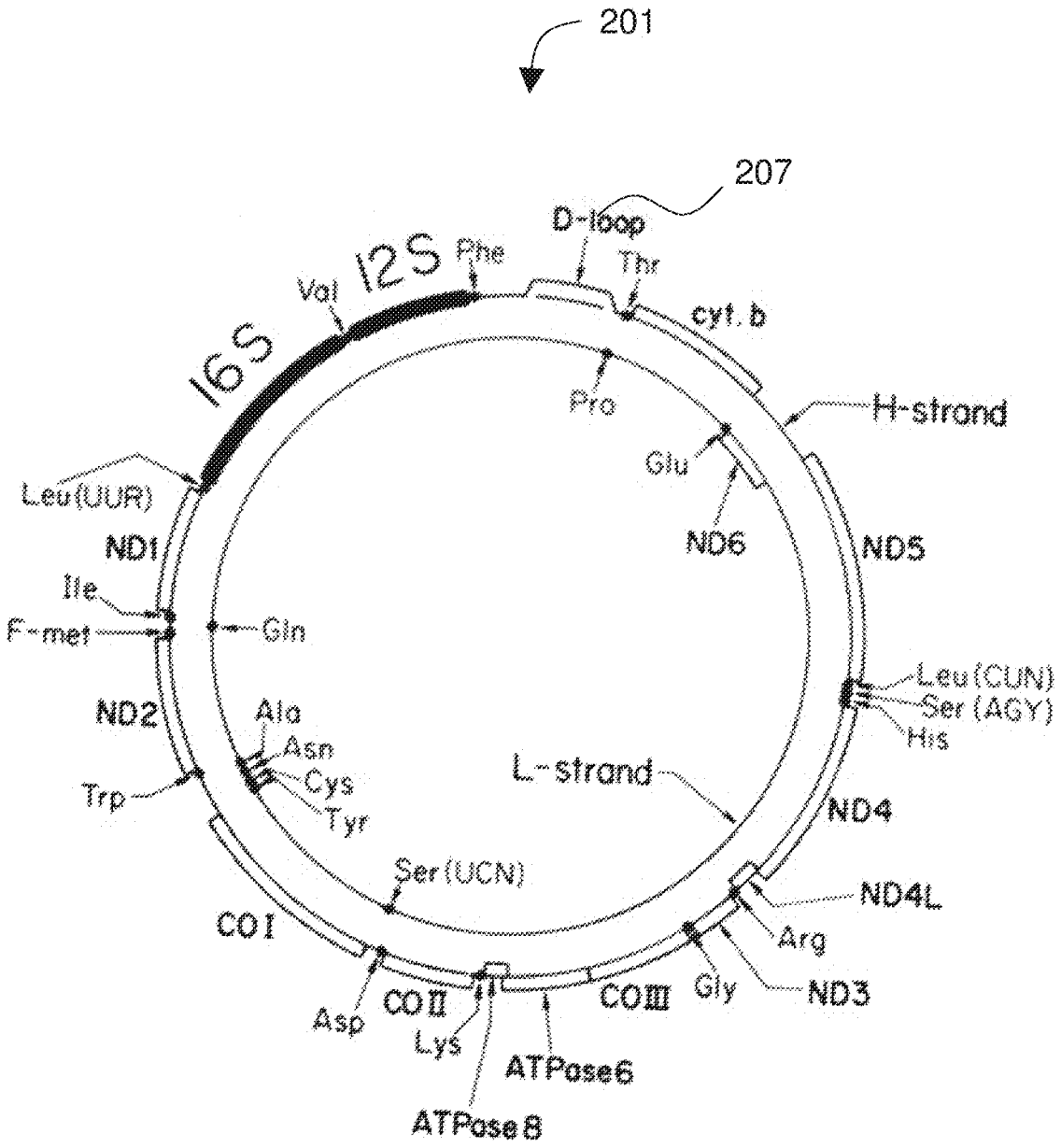

101
105
Obtain mitochondrial sequences
109
Transform blocks into graph objects
115
Connect objects to create graph with paths representing sequences
123
Obtain sequence reads
129
Find alignments
129
Provide report

FIG. 1



Reproduced from Ding et al, 2013 Mitochondrial DNA mutations and essential hypertension, Int J Mol Med 32(4):768-774

FIG. 2

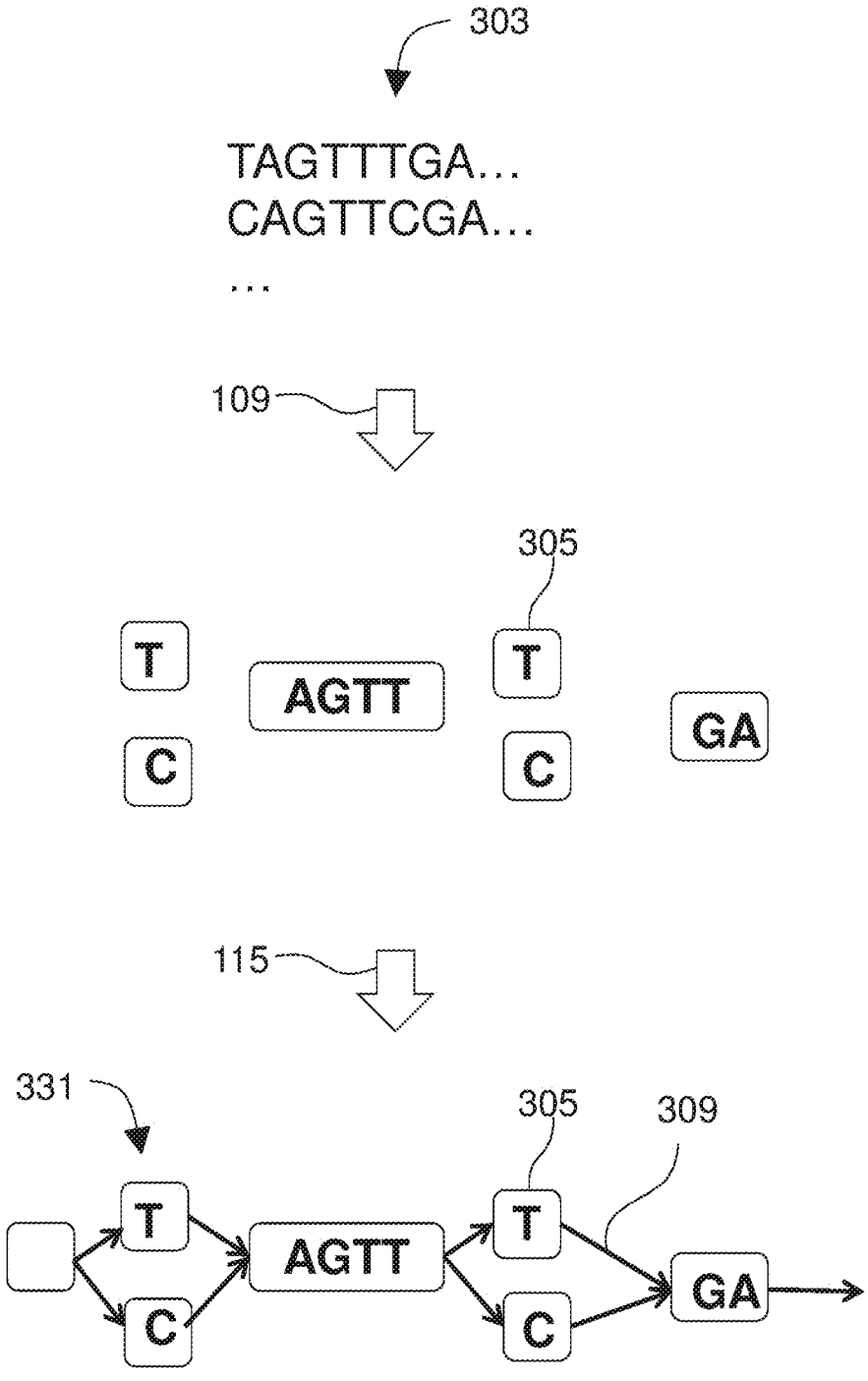

303
TAGTTTGA…
CAGTTCGA…
…
109
305
T
C
AGTT
T
C
GA
115
331
305
309
T
C
AGTT
T
C
GA

FIG. 3

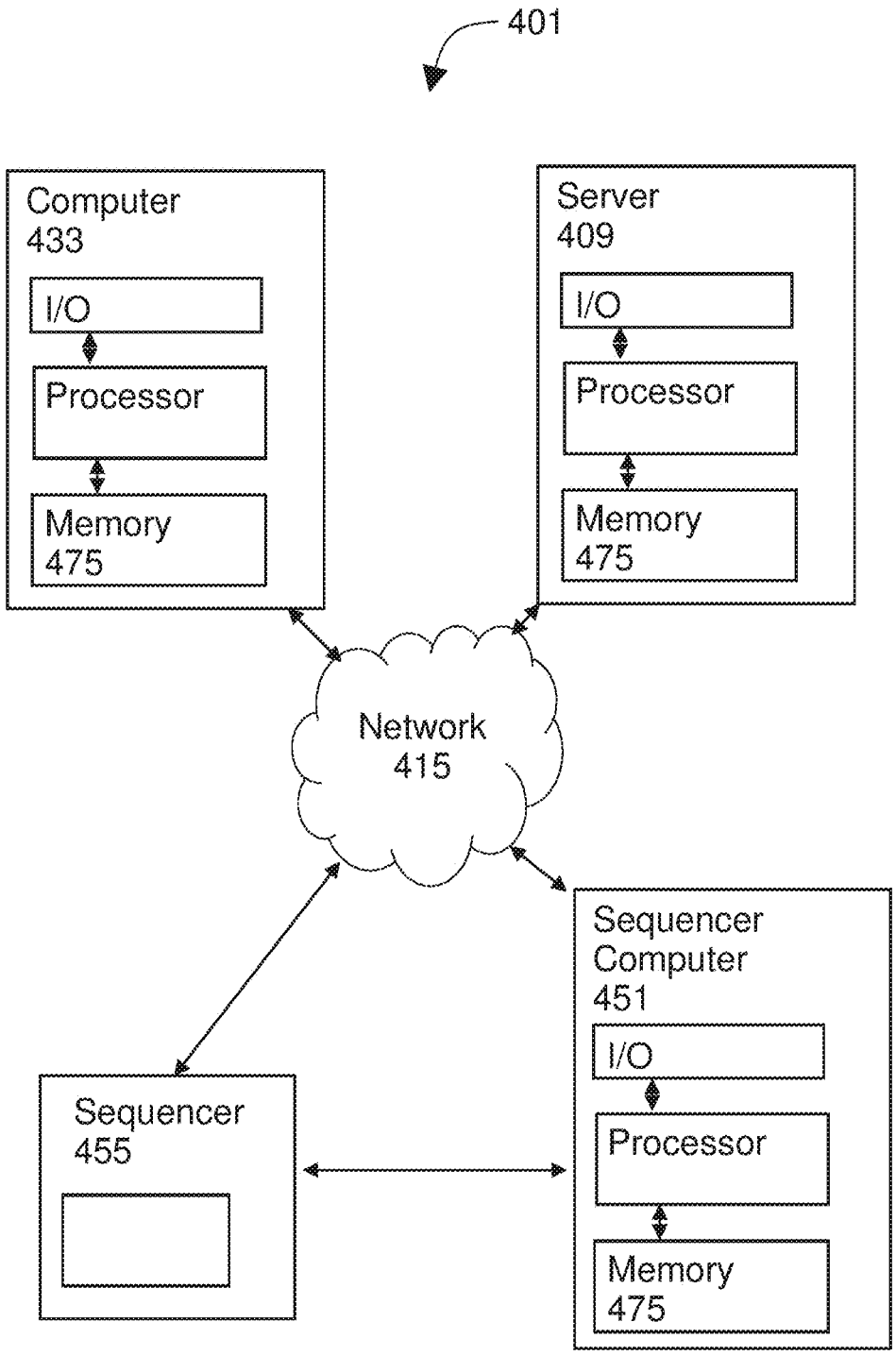

401
Computer
433
I/O
Processor
Memory
475
Server
409
I/O
Processor
Memory
475
Network
415
Sequencer
Computer
451
I/O
Processor
Memory
475
Sequencer
455

FIG. 4

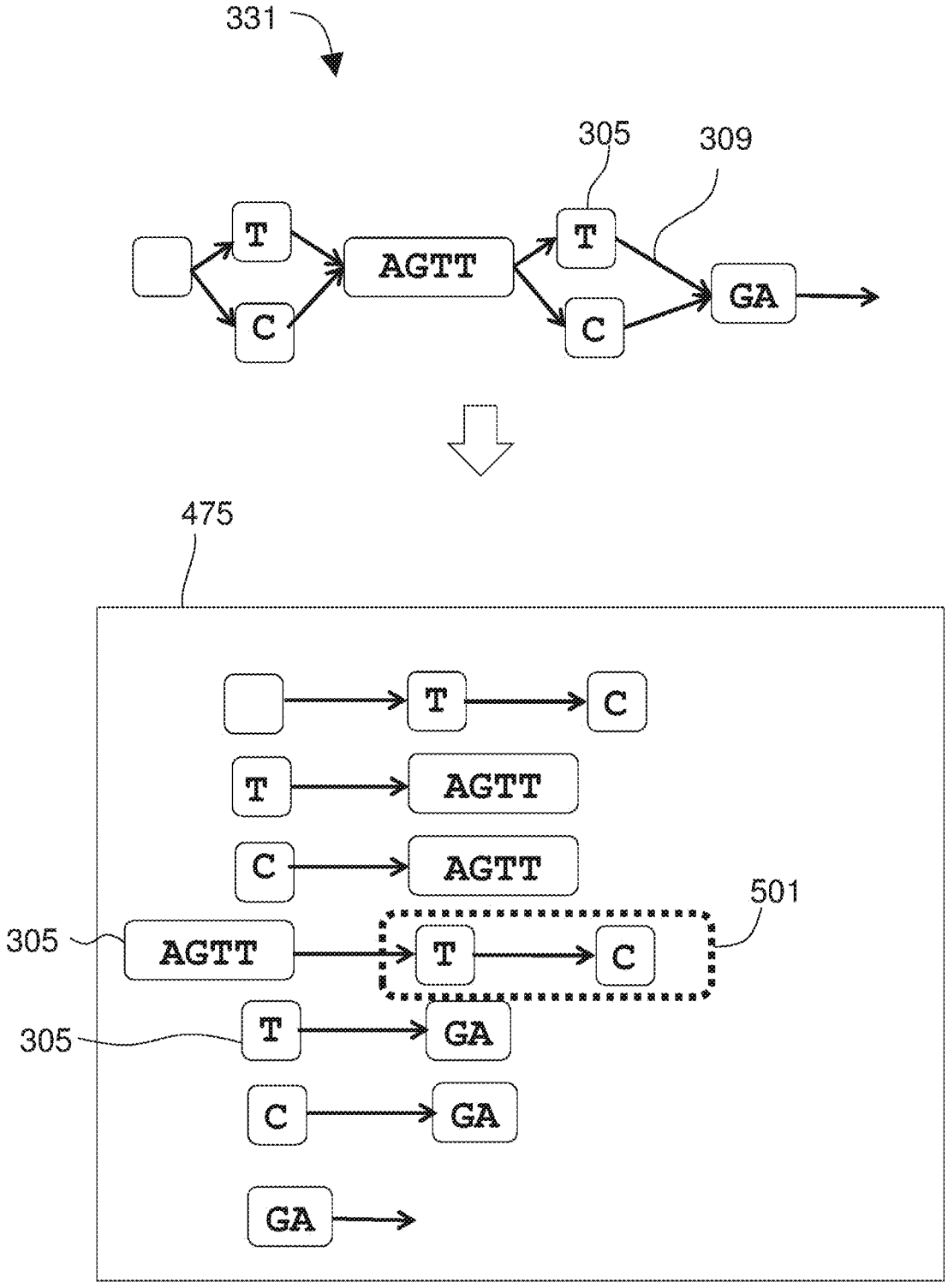

331
305
309
T
C
AGTT
T
C
GA
475
T
C
T
AGTT
C
AGTT
501
305
AGTT
T
C
305
T
GA
C
GA
GA

FIG. 5

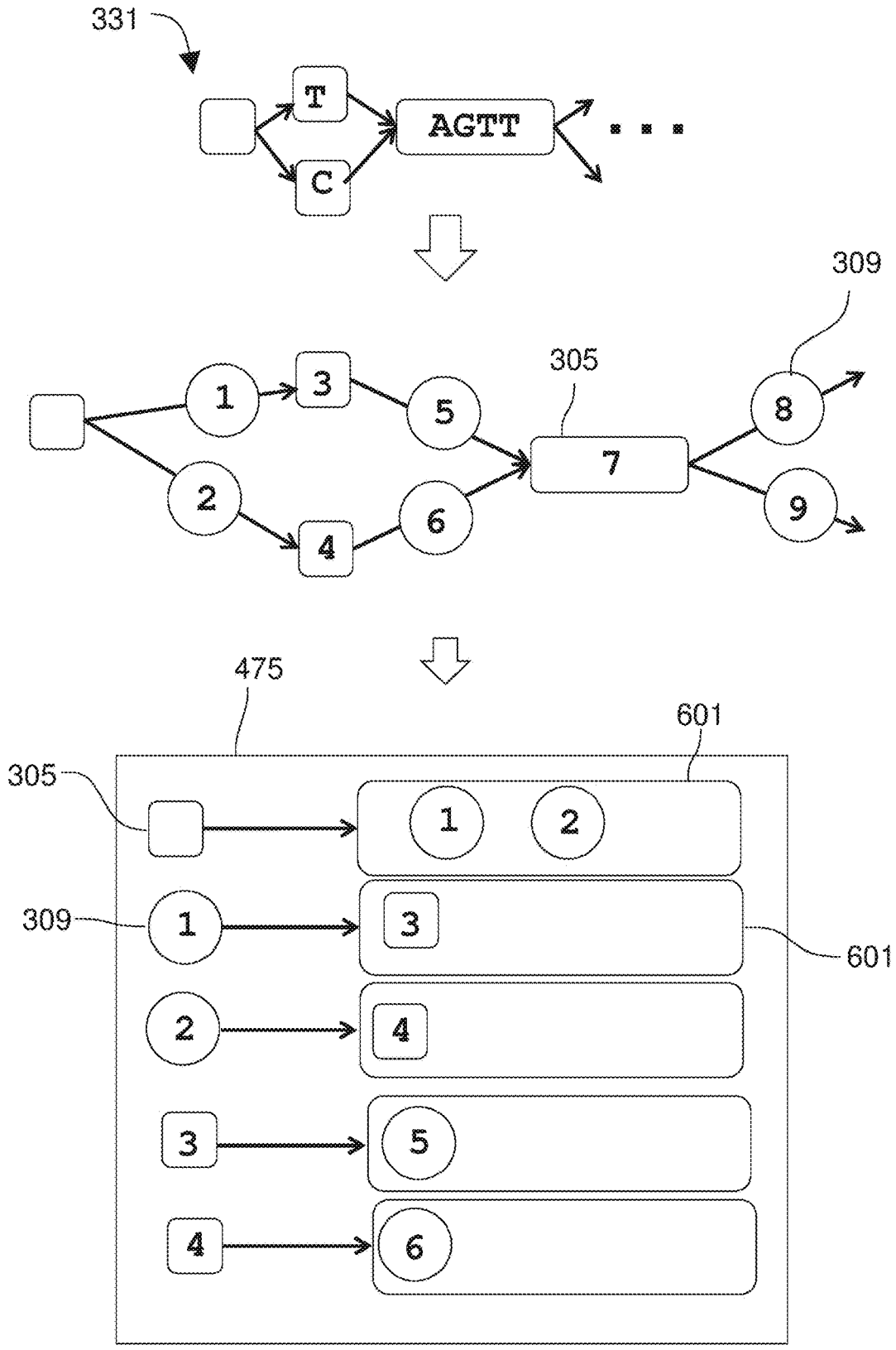


FIG. 6

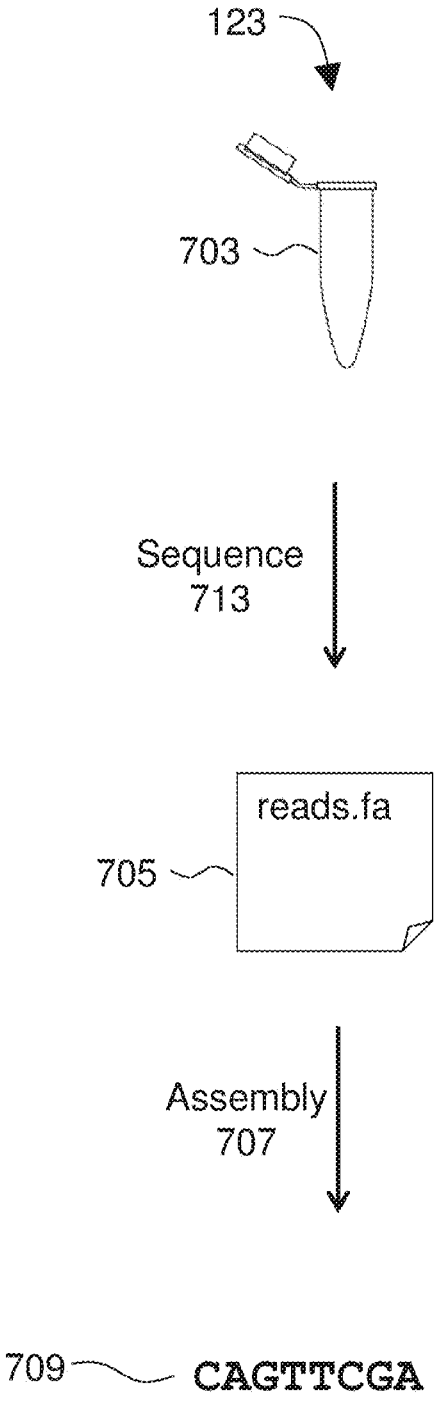


FIG. 7

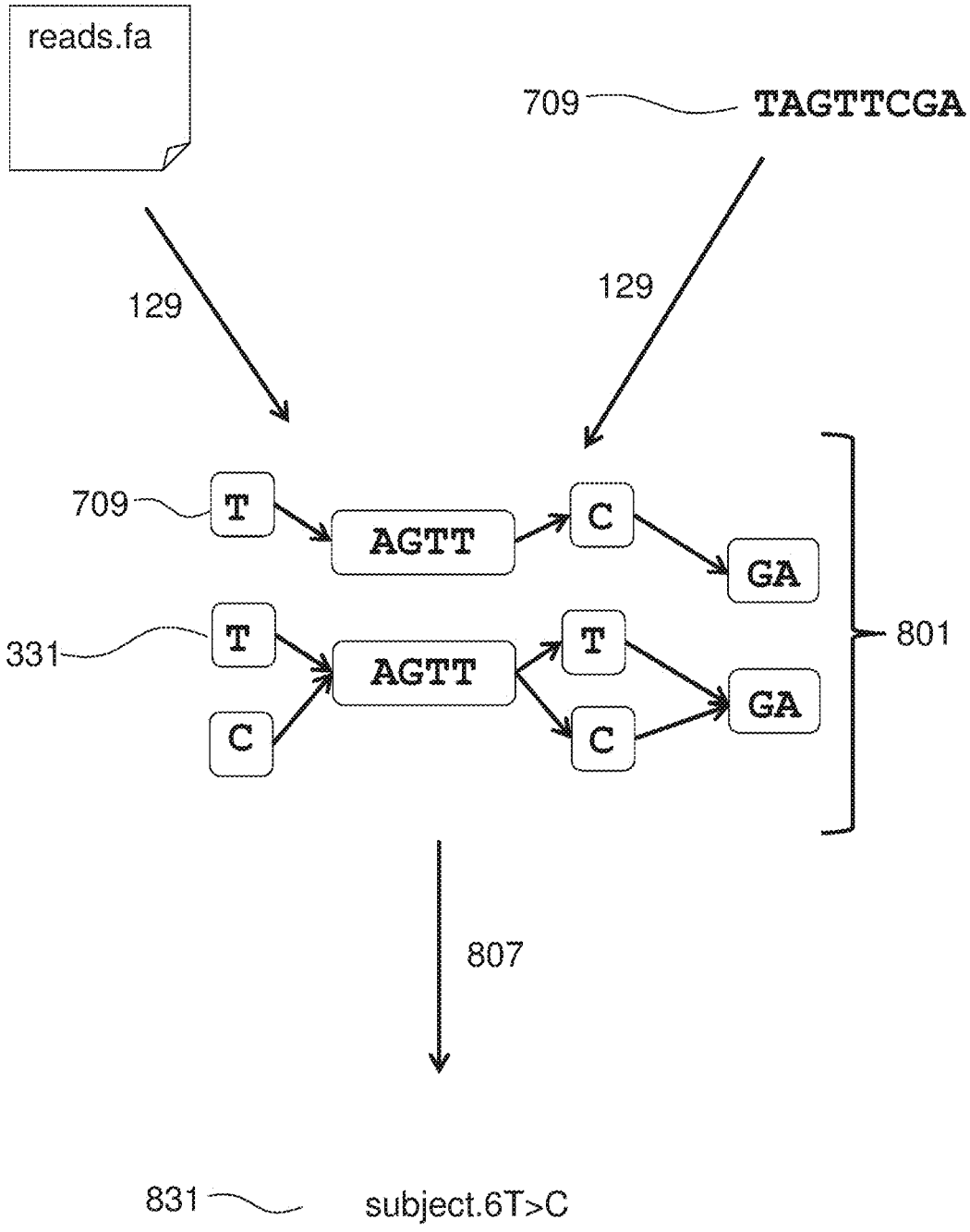

reads.fa
709
TAGTTCGA
129
129
709
T
AGTT
C
GA
331
T
C
AGTT
T
C
GA
801
807
831
subject.6T>C

FIG. 8

		T	T	G	G	A	T
	0	**0**	**0**	**0**	**0**	**0**	**0**
A	**0**	0	0	0	0	**10**	0
T	**0**	10	10	0	0	0	**20**
C	**0**	0	0	0	0	0	10
G	**0**	0	0	10	10	0	0
A	**0**	0	0	0	0	20	10
A	**0**	0	0	0	0	10	10

		C	G	A	A	T	T
	0	**0**	**0**	**0**	**0**	**0**	**0**
A	**0**	0	0	10	10	0	0
T	**20**	10	0	0	0	20	10
C	**10**	**30**	20	10	0	10	10
G	**0**	20	**40**	30	20	10	0
A	**10**	10	30	**50**	40	30	20
A	**10**	0	20	40	**60**	50	40

		A	T	G	G	G
	0	**0**	**0**	**0**	**0**	**0**
A	**0**	10	0	0	0	0
T	**20**	10	20	10	0	0
C	**10**	10	10	10	0	0
G	**0**	0	0	20	20	10
A	**20**	10	0	10	10	10
A	**40**	30	20	10	0	0

FIG. 9

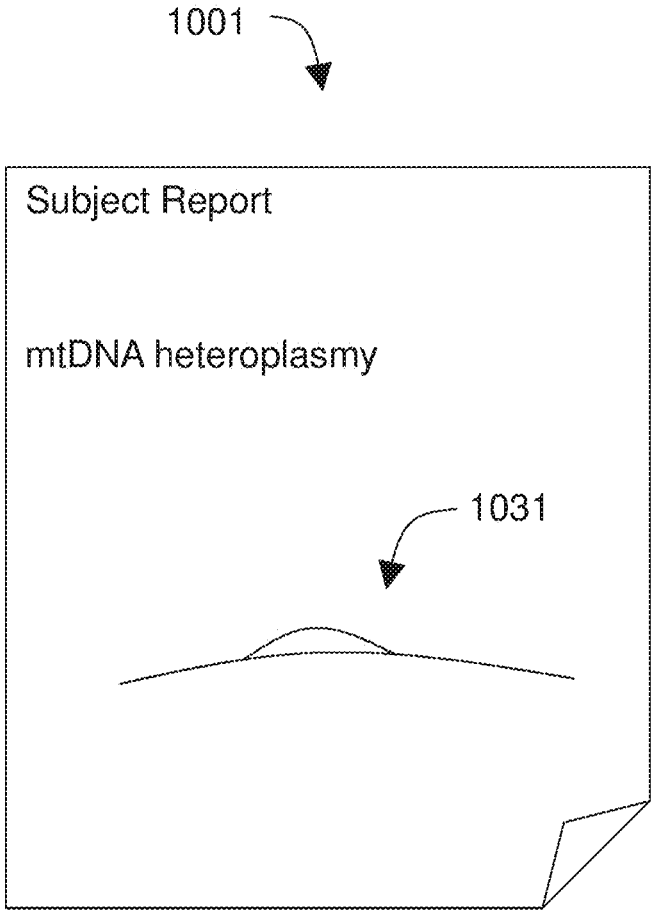


FIG. 10

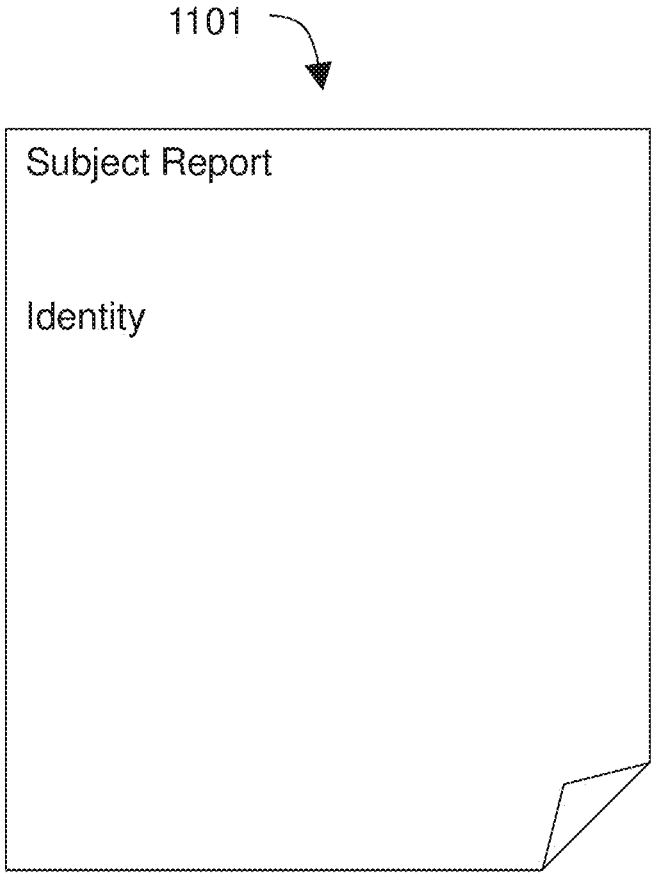


FIG. 11

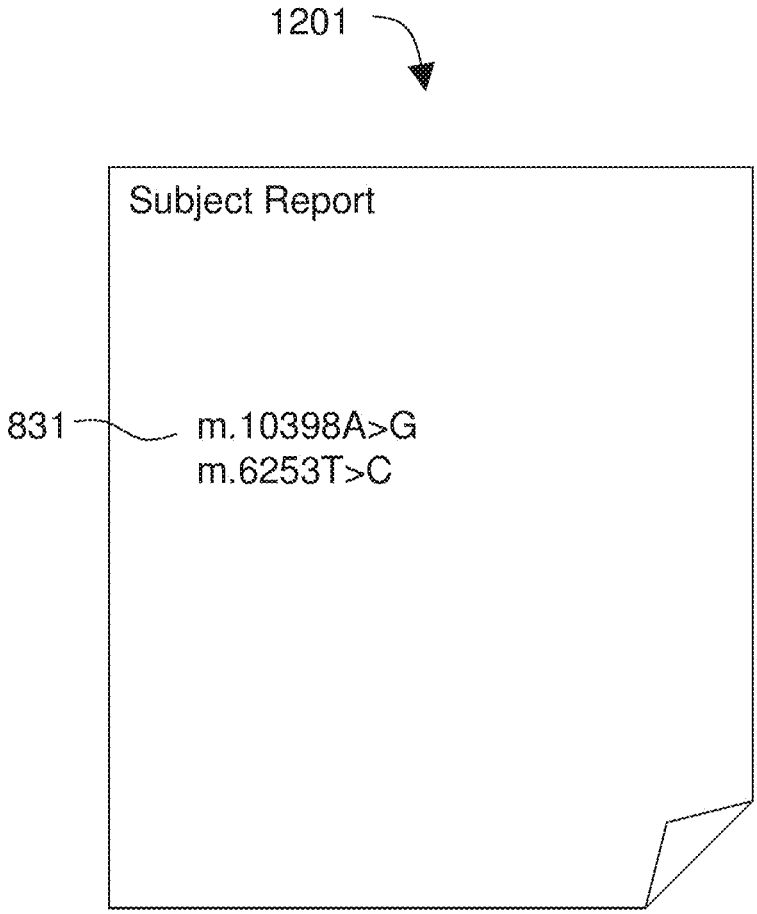


FIG. 12

SYSTEMS AND METHODS FOR MITOCHONDRIAL ANALYSIS

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application claims priority to, and the benefit of, U.S. Provisional Patent Application Ser. No. 62/212,886, filed Sep. 1, 2015, the contents of which are incorporated by reference.

TECHNICAL FIELD

[0002] The invention relates to the analysis of mitochondrial genetic material.

BACKGROUND

[0003] People inherit diploid nuclear chromosomes from both parents, but also inherit a significant amount of maternal mitochondrial DNA. The mitochondrial DNA encodes, among other things, proteins of the electron transport chain that are used in oxidative metabolism.

[0004] Because mitochondrial genomes contain both conserved and hypervariable regions, are maternally inherited, and do not recombine, they have been used extensively to identify, and determine the evolutionary history of, various organisms.

[0005] Mutations within the mitochondrial genome have been associated with diseases such as cancer, cardiovascular disease, diabetes, hearing loss, and neurodegenerative disease. Analyzing mitochondrial genes can potentially reveal important medical information. Also, since most human cells contain hundreds of copies of the mitochondrial genome, a useful amount of mitochondrial DNA (mtDNA) can sometimes be recovered from samples that are too small or degraded to reliably yield nuclear DNA. Thus mtDNA has the potential to be an important tool in forensics, e.g., in a missing persons case or in studying a natural disaster.

[0006] Unfortunately, analyzing a subject's mtDNA does not always produce a useful result. Next-generation sequencing (NGS) technologies give very deep coverage-even millions of reads. But the very quantity of information that can be obtained by NGS and the quantity of reference material potentially available for comparisons can mean that a full multiple-sequence alignment and comparison of all inputs may not be computationally possible. Moreover, it is now understood that a person does not have one mitochondrial genome. Instead, in a phenomenon known as heteroplasmy, each cell in a person can include as many as 10 differing genomes within the hundreds of copies of the mitochondrial DNA. See Sosa et al., Next-generation sequencing of human mitochondrial reference genomes uncovers high heteroplasmy frequency, PLoS Comp Biol 8(10):e1002737. Thus the paradigm of sequencing a gene and finding a match in a database may not even be strictly applicable with mtDNA.

SUMMARY

[0007] The invention provides methods of analyzing an individual's mtDNA by transforming available reference sequences into a directed graph that compactly represents all the information without duplication and comparing sequence reads from the mtDNA to the graph to identify the individual or describe their mtDNA. A directed graph can represent all of the genetic variation found among the mitochondrial genomes across all of a number of reference organisms while providing a single article to which sequence reads can be aligned or compared. Thus any sequence read or other sequence fragment can be compared, in a single operation, to the article that represents all of the reference mitochondrial sequences. Since conserved portions across the reference sequences are stored as singular paths through the directed graph, without duplication, very large amounts of original reference genome information can be analyzed without exceeding computational or data storage limits. Since the reference article can be stored as a directed graph that is characterized by adjacency relationships between the node and edge objects that constitute the graph, the objects and their adjacency relationships can be implemented using pointers that address specific physical locations in a storage system. Since the reference sequence data can be accessed by reading and de-referencing pointers, reading across the graph is very rapid. That is, comparison operations are much more rapid than they would be using indices in a relational database. Thus, transforming reference mitochondrial sequences into a directed graph for use as a reference article for the analysis of mtDNA sequence reads greatly increases the speed and capacity of a computer system used for mitochondrial analysis in comparison to using a computer without the structures such as pointers to specific physical locations in memory that are provided by the invention. Since structures of the invention allow the reference article to represent a potentially complete set of all potential reference sequences, mtDNA sequence reads can be analyzed against a great amount of genetic variation among mitochondrial genomes. Since systems and methods of the invention allow mitochondrial analysis to be fast and comprehensive, they provide for very effective tools for medical genetics and forensic investigations.

[0008] In certain aspects, the invention provides a method for analyzing a mitochondrial genome from an organism. The method includes representing a plurality of mitochondrial sequences as a directed graph comprising objects stored in a tangible memory device, wherein portions of the sequences that match each other when aligned are each represented by a single object and wherein each of the sequences is represented by a path through the directed graph. Sequence reads are obtained from a sample from a subject and the method includes finding alignments between the sequence reads and paths through the directed graph using a processor coupled to the tangible memory device. A report is provided that may identify one or more of the mitochondrial sequences that aligned to the sequence reads. The report can identify, for example, heteroplasmy in the subject, the identity of the subject, or mutations in the subject's mitochondrial genome. The plurality of mitochondrial sequences may be obtained from relatives of the subject.

[0009] Preferably, the directed graph comprises vertex objects connected by edge objects and an adjacency list for each vertex object and edge object, wherein the adjacency list for a vertex object or edge object lists the edge objects or vertex objects to which that vertex object or edge object is adjacent. Each entry in the adjacency list is a pointer to the adjacent vertex object or edge object. In some embodiments, each pointer identifies a physical location in the memory subsystem at which the adjacent object is stored. In certain embodiments, finding alignments between the sequence reads and paths through the directed graph is done using a

multi-dimensional look-back operation to find a highest-scoring trace through a multi-dimensional matrix.

[0010] Representing the plurality of mitochondrial sequences as the directed graph may be done by obtaining each of the plurality of mitochondrial sequences, using the processor to find the portions of the sequences that match each other when aligned, creating—using the processor—the objects to represent the portions, storing each of the objects in the tangible memory device, and connecting the objects to create paths through the directed graph such that each of the sequences is represented by one of the paths. The method may be used with whole mitochondrial genomes. In a preferred embodiment, each of the plurality of mitochondrial sequences represents at least 80% of a mitochondrial genome.

[0011] Aspects of the invention provide a system for analyzing a mitochondrial genome from an organism. The system includes a tangible memory device having stored therein a directed graph representing a plurality of mitochondrial sequences. Portions of the sequences that match each other when aligned are each represented in the graph by a single object and each of the sequences is represented by a path through the directed graph. The system also includes a processor coupled to the memory device. The system is operable to obtain sequence reads from a sample from a subject, find alignments between the sequence reads and paths through the directed graph, and provide a report via an output device that identifies one or more of the mitochondrial sequences that aligned to the sequence reads. Within the system, the directed graph may be implemented using vertex objects connected by edge objects and an adjacency list for each vertex object and edge object, wherein the adjacency list for a vertex object or edge object lists the edge objects or vertex objects to which that vertex object or edge object is adjacent. Each entry in an adjacency list may be a pointer to the adjacent vertex object or edge object. Preferably, each pointer identifies a physical location in the memory subsystem at which the adjacent object is stored. In some embodiments, the system finds alignments between the sequence reads and paths through the directed graph by performing a multi-dimensional look-back operation to find a highest-scoring trace through a multi-dimensional matrix. In certain embodiments, the system is operable to obtain each of the plurality of mitochondrial sequences, use the processor to find the portions of the sequences that match each other when aligned, create—using the processor—the objects to represent the portions; store each of the objects in the tangible memory device, and connect the objects to create paths through the directed graph such that each of the sequences is represented by one of the paths.

[0012] Aspects of the invention further comprise a method of detecting mitochondrial heteroplasmy in a subject. The method comprises representing a plurality of known variations in the mitochondrial genome as a directed graph. Each of the known variations is associated with a path through the directed graph, and the directed graph comprises objects stored in a tangible memory device. Nucleotide sequence information is associated with the objects. A plurality of sequence reads from a sample from a subject are obtained, and each sequence read is aligned to the directed graph. The aligning can comprise finding the most likely position on the directed graph for the sequence read based on the sequence read and the nucleotide sequence information associated with each object. In certain embodiments, the aligning can comprise a multi-dimensional look-back operation to find a highest-scoring trace through a multi-dimensional matrix. Based on the aligned sequence reads, at least one position in the directed graph is identified in which sequence reads align to alternate paths. A report may be provided identifying mitochondrial heteroplasmy in the subject based on the identified at least one position. A second position may also be identified in the directed graph in which sequence reads align to alternate paths.

[0013] Aspects of the invention can further comprise a method of identifying an unknown individual. The method can comprise representing a plurality of known variations in the mitochondrial genome as a directed graph. Each of the known variations is associated with a path through the directed graph, and the directed graph comprises objects stored in a tangible memory device. Nucleotide sequence information is associated with the objects. A plurality of sequence reads from a sample from an unknown subject are obtained, and each sequence read is aligned to the directed graph. The aligning can comprise finding the most likely position on the directed graph for the sequence read based on the sequence read and the nucleotide sequence information associated with each object. The identity of the unknown subject may then be determined based on the aligned sequence reads. In certain examples, at least one of the known variations comprise variations in the mitochondrial genome of a maternal-line individual related to the unknown subject. In certain examples, the method may further comprise obtaining a plurality of maternal-line sequence reads from a sample from a maternal-line relative of the unknown subject. Each maternal-line sequence read is similarly aligned to the directed graph. The alignment of the maternal-line sequence reads and the alignment of the sequence reads from the unknown subject are then compared. The identity of the unknown subject may then be determined based on the comparison. In certain examples, the known variations can comprise a hyper-variable region of the mitochondrial genome.

BRIEF DESCRIPTION OF THE DRAWINGS

[0014] FIG. **1** diagrams a method for analyzing an organism's mtDNA.

[0015] FIG. **2** diagrams a typical human mitochondrial genome.

[0016] FIG. **3** illustrates transforming mitochondrial reference sequences into a graph.

[0017] FIG. **4** illustrates a computer system for performing methods of the invention.

[0018] FIG. **5** shows the use of an adjacency list for each vertex.

[0019] FIG. **6** shows the use of an adjacency list for each vertex and edge.

[0020] FIG. **7** illustrates obtaining sequence reads from a sample.

[0021] FIG. **8** illustrates finding alignments between the sequence and the graph.

[0022] FIG. **9** shows the matrices that represent a comparison operation.

[0023] FIG. **10** shows a report that identifies heteroplasmy within an organism.

[0024] FIG. **11** shows a report that provides the identify of a subject.

[0025] FIG. **12** shows a report that describes variants in a subject's mitochondrial genome.

DETAILED DESCRIPTION

[0026] The invention provides systems and methods for analyzing DNA that is present in the cellular organelles known as mitochondria found in the cells of most Eukaryotic organisms. Those organelles are responsible for cellular energy production and they also each contain their own genomic DNA, thought to be a relic of these organelles' origin as independent organisms absorbed and repurposed by eukaryotes in eons past. Many NGS projects focus on the nuclear genome. However in the last few years, researchers have sought to apply NGS to the sequencing of mitochondrial DNA (mtDNA). The present invention provides systems and methods in which graph-based analysis may be applied to sequencing projects involving mtDNA. Systems and methods of the invention may be used for analysis of the mtDNA and may find particular application for the better detection of heteroplasmy and the better identification of unknown individuals.

[0027] FIG. **1** diagrams a method **101** for analyzing a mitochondrial genome from an organism. At step **105** a plurality of mitochondrial sequences are obtained. Portions of the sequences that match each other when aligned are identified as blocks that are transformed **109** into objects that are stored in a tangible memory device. The objects are connected **115** to create paths such that there is a path for each of the original mitochondrial sequences. This creates a new article, a directed graph comprising objects stored in the tangible memory device.

[0028] In certain embodiments, the directed graph may be created by associating an initial object with a mitochondrial reference genome, such as the Revised Cambridge Reference Sequence of the human mitochondrial genome (rCRS). Known variations from the reference genome previously observed across a population, such as single nucleotide polymorphisms, small insertions and deletions (indels), and larger structural variants, may be associated with additional objects. The object representing the mitochondrial reference genome may then be divided into multiple objects at positions in which the known variations occur, and the plurality of objects are then connected to create paths such that there is a path for each known variation. Both a mitochondrial reference genome and known variations may be accessed via MITOMAP (www.mitomap.org), a human mitochondrial genome database that includes a compendium of polymorphism and mutations in human mitochondrial DNA, for example.

[0029] Method **101** preferably further includes obtaining **123** sequence reads from a sample from a subject. Sequence reads can be obtained from a nucleic acid sequencing instrument. A processor coupled to the tangible memory device is used to find 129 alignments between the sequence reads and the paths through the directed graph. A report is provided **129** that identifies a one or more of the mitochondrial sequences that aligned to the sequence reads. Specifically, the report may characterize heteroplasmy in the organism, provide the identity of the organism, identify significant mutations or other genetic features (e.g., gene rearrangements or truncated tRNAs) in the organism, or otherwise describe the organism or its mitochondrial genome.

[0030] FIG. **2** diagrams a typical human mitochondrial genome **201**. The mitochondrial DNA molecule is usually a covalently closed circle of about 14,000 to 40,000 nucleotide base pairs. See Wolstenholme 1992, Animal mitochondrial DNA: structure and evolution. Int Rev Cytol 141:173-216. For mammals, approximately 16,000 base-pairs is standard. There are typically 37 genes encoded on the mitochondrial genome. The gene content of the mitochondrial genome is highly conserved and the genes are compactly arranged with few non-coding nucleotides. Protein coding genes are often separated by tRNAs-which are thought to signal RNA primary transcript processing. It is thought that this arrangement is achieved by selection for small genome size and, by implication, speed of replication. See Moritz et al. 1987, Evolution of animal mitochondrial DNA: Relevance for population biology and systematics. Annual review of ecology and systematics 18:269-292.

[0031] Most mitochondrial genomes have one major non-coding region—the D-loop **207**—that has no open reading frames of significant length. This region is on the order of 2000 base pairs, and is slightly more AT-rich than the rest of the molecule. This region often contains hairpin structures, some of them GC rich. This region is thought to be involved in initiation of replication and transcription of one of the two strands. Transcription termination is highly conserved in animal mitochondria. It is signaled by a second conserved non-coding sequence downstream of the 16S rRNA. That AT rich region is often a source of variation in size of mitochondrial genomes. See Boore, 1999, Animal Mitochondrial Genomes, Nucl Ac Res 27:1767-80. Replication is asymmetrical; in some organisms it begins on one strand, and after much of one strand has been replicated, replication of the other strand begins. One strand is deemed the light strand, the other heavy, based on their separation by CsCl centrifugation. Over 1,100 different metazoan mitochondrial genomes have been sequenced and published. Because mitochondrial genomes do not recombine, and genomic arrangements are likely unique, and convergent evolution of arrangements is unlikely, mitochondrial genomes are often used in phylogenetic inference. See e.g., Sankoff et al., 1992, Gene order comparisons for phylogenetic inference: evolution of the mitochondrial genome, PNAS 89:6575-6579. Of the 37 mitochondrial genes, there are 13 protein-coding genes whose products participate in the coupled pathways of electron transport and ATP synthesis. The mitochondrial genome usually contains 22 genes coding for tRNAs and 2 genes coding for rRNAs, which work with proteins imported from the cytoplasm in protein synthesis. The mitochondria phosphorylate ADP to ATP, which is then used throughout the host cell for energy.

[0032] In humans, the heavy strand of mtDNA carries 28 genes and the light strand of mtDNA carries only 9 genes. Eight of the 9 genes on the light strand code for mitochondrial tRNA molecules. Human mtDNA consists of typically about 16,569 nucleotide pairs. The entire molecule is regulated by only one regulatory region which contains the origins of replication of both heavy and light strands. The entire human mitochondrial DNA molecule has been mapped. In mammals, all but six of the 37 genes are on the heavy strand. The gene for ND6, as well as five of the 22 tRNA genes are on the light strand. In the nematodes *C. elegans* and *Meloidogyne javanica* as well as in *Drosophila yakuba* all of the genes are on the same strand, oriented in the same direction, and—it is thought—transcribed in one polycistronic unit. In other metazoans, there is much variation in how the genes are distributed between the strands.

[0033] Protein coding or ribosomal genes in mitochondria are often separated by a single tRNA gene with few to no other nucleotides. It is thought that the base pairing neces-

sary for functional tRNAs also plays a role in signaling processing of the RNA transcript. This has been dubbed the "tRNA Punctuation Model" in Ojala et al., 1980, The tRNA genes punctuate the reading of genetic information in human mitochondrial DNA, Cell 22:393-403, incorporated by reference.

[0034] One of the challenges in studying mtDNA is that mitochondria each contain multiple copies of the mitochondrial genome, and because of the way in which mtDNA is inherited an individual mitochondrion or the organism as a whole can contain several different versions of the genome. This within-organism variation in mtDNA is known as heteroplasmy. See Sosa et al., Next-generation sequencing of human mitochondrial reference genomes uncovers high heteroplasmy frequency, PLoS Comp Biol 8(10):e1002737. In turn, one of the verified advantages of NGS methods over Sanger sequencing in the mtDNA is that the high throughput and deep coverage allows for better detection of heteroplasmy.

[0035] Heteroplasmy is just the sort of problem well-suited for analysis using graph-based methods, which are able to represent potential heteroplasmy events in a reference graph. This can be accomplished by obtaining a plurality of mitochondrial sequences and/or known variations and transforming them into a directed graph that can be used to describe heteroplasmy in a subject.

[0036] Heteroplasmy can be described by the following strategy. First, a reference graph (which may be a directed acyclic graph or DAG) is built from all known mitochondrial genome references and known variants. Next, reads from the mtDNA of an unknown sample are aligned to the graph, which may be done using the modified Smith-Waterman algorithm described below. Third, reads aligned to different branches at a given position can be interpreted as heteroplasmy. (Optionally, a threshold may be imposed, e.g. at least 0.5% of reads aligning to a branch other than the majority/"consensus" branch.) There are a number of ways in which results can be reported that preserve heteroplasmy data, including for example as a DAG with a majority/plurality/"consensus" sequence along with branches at any positions at which the heteroplasmy was detected.

[0037] In other embodiments, the invention provides systems and methods that may be used for the identification of unknown individuals. For example, systems and methods of the invention may be used in forensics, or to identify missing persons.

[0038] Because each cell typically contains hundreds of copies of the mitochondrial genome and these are usually more easily recovered from severely degraded or limited samples than nuclear DNA, analysis of mtDNA is frequently conducted in forensics and the identification of missing persons. Such analysis also benefits from the fact that mtDNA is inherited solely down the maternal line, meaning that even persons generations apart or in distant branches of a family will share a similar set of mitochondrial genomes as long as they have a common maternal-line ancestor. See e.g., Budowle et al., 2003, Forensics and mitochondrial DNA, Ann Rev Genom Hum Genet 4:119-41, incorporated by reference.

[0039] In one embodiment, identification of unknown individuals is performed against a background of known candidates. The embodiment may be preferred when it is clear that the unknown individual is one of the known candidates.

[0040] For this embodiment, first, a reference graph is built from the sequences of mitochondrial genomes and/or known variations from maternal-line relatives of each candidate, including those reflecting known heteroplasmy. A record is kept of which branches or paths of the graph correspond to which candidate (with some branches corresponding to multiple candidates). Reads from the mtDNA of an unknown individual are aligned to the graph. The branches to which the reads have been aligned are identified, along with the corresponding candidates.

[0041] Results can be reported as a list of the candidates corresponding to the branches to which the reads aligned, along with the percentage of reads aligned (or percentage of nucleotides matched) to each. In some embodiments, branches (and thus candidates) are weighted according to a delta between the branch and the next-best alignment for a given read. Paths through the graph that include multiple variants may be scored to develop matching probabilities (cumulative probability).

[0042] A second embodiment provides an alternative approach to the identification of unknown individuals from among known candidates. In this embodiment, a reference graph is built from all known mitochondrial genome references and known variants (e.g., a reference graph built to perform heteroplasmy detection as described herein). Reference mitochondrial genome information and known variations may be retrieved from reference databases such as GenBank and dbSNP, by sequencing subject organisms, by retrieving sequence files, others, or combinations thereof. Reads from the mtDNA of an unknown individual are aligned to the graph, e.g., using the modified Smith-Waterman operation described below. Reads from the mtDNA of maternal-line relatives of each candidate are similarly aligned to the graph. The alignment of the unknown individual and the maternal-line candidate relatives are compared to determine whether or not the unknown individual is likely to be one of the candidates.

[0043] Embodiments of the invention provide for the identification of unknown individuals where there are not particular candidates for the identity of the unknown individual. Even when there are no known candidates for the identity of a sample being sequenced, graph-based methods can help minimize reference bias by representing in the reference all known variations in hypervariable regions. Information on mtDNA sequencing targeting hyper-variable regions for the purpose of identification may be found in Morovvati et al., 2007, Sequence analysis of mitochondrial DNA hypervariable regions; an approach to personal identification, Arch Med Res 38(3):345-9, incorporated by reference. In this embodiment, a reference graph is built of the hypervariable D-loop **207** region of the mitochondrial genome, including all known variants.

[0044] Reads from the D-loop **207** of the mitochondrial genome of an unknown person are generated (e.g., by using targeted sequencing methods) and aligned to the directed graph using methods described herein. A sequence describing a particular D-loop for the unknown person is reported. Optionally, branches to which the sequence aligned can be reported as a sort of summary "bar code" of the sample. For example, the "bar code" can be the most likely path through the graph representing the unknown person's' D-loop sequence. Additional information may be found in Yang et al., 2014, Species identification through mitochondrial rRNA genetic analysis, Scientific Reports 4:4089, incorpo-

rated by reference. The different applications and embodiments described above each include the use of a reference directed graph that is created using a plurality of mitochondrial reference sequences.

[0045] FIG. **3** illustrates obtaining mitochondrial reference sequences **303** and transforming **109** the reference sequences **303** into a graph **331** that includes vertex objects **305** and edge objects **309**. Each of the sequences **303** are aligned to another and in some embodiments, a multiple sequence alignment is performed. Portions of the sequences that match each other when aligned are identified as blocks and those blocks are transformed **109** into objects **205** that are stored in a tangible memory device.

[0046] In the fragments of sequence represented in FIG. **3**, it can be seen that bases 2-5 of first sequence align to, and match, bases 2-5 of the second sequence. Thus those segments of those two sequences are identified as a block and systems of the invention create an object **305** to represent that AGTT string. It is noted that this object could potentially be stored using one byte of information. For example, if A=00, C=01, G=10, and T=11, then this block contains 00101111 (one byte). Where the original sequences **303** contain thousands of mitochondrial genomes, the described methods provide a considerable improvement to the operation of the computer system in comparison to a prior art method that stores an entire multiple sequence alignment.

[0047] The objects **305** are connected **115** to create paths such that there is a path for each of the original mitochondrial sequences. The paths are directed and preferably in the sense that the direction of each path corresponds to the 5' to 3' directionality of the mtDNA. The connections creating the paths can themselves be implemented as objects so that the blocks are represented by vertex objects **305** and the connections are represented by edge objects **309**. Thus the directed graph comprises vertex and edge objects stored in the tangible memory device. The directed graph **331** represents the plurality of mitochondrial sequences **303** in that each one of the original sequences can be retrieved by reading a path in the direction of that path. However, the directed graph **331** is a different article than the original sequences **303**, at least in that portions of the sequences that match each other when aligned have been transformed into single objects **303**. Thus if the original article includes 90,000 full mitochondrial genomes in which the NADH dehydrogenase 4 gene is perfectly conserved for a span of 12,000 bp across all of the genomes, then over 1 billion characters of information from the original article are transformed into a single object that can use less than 3 KB on disk.

[0048] In some embodiments, the directed graph **331** is a directed acyclic graph (DAG). Any arbitrary point within the mitochondrial genome may be selected to correspond to the source of the graph. For example, the genome may be imaginarily split at the origin of replication of the heavy strand. The 5'-most base of the heavy strand may be represented by the source node object **305** of graph **331** and the 3'-most base of the heavy strand may be represented by the sink node object **305** of graph **331**. Where a DAG is used to represent homologous genomic sequences, it may be convenient to have one or more source nodes correspond to the 5' end of those sequences and one or more sink nodes correspond to the 3' end.

[0049] In certain embodiments, a DAG is used and the sequence data is stored in the edge objects **309**. This may be useful where the 5'-most nucleotide is not conserved across the plurality of linear sequences. Thus the source node object **305** does not store any sequence data and each edge object **309** includes relevant sequence data.

[0050] In other embodiments, a directed graph is constructed. This may be appealing where it is wished to represent the circular nature of the mitochondrial genome. Either the heavy strand or the light strand may be fully represented using a cyclic, directed graph. Additionally, it may be desirable to represent a mitochondrial genome using two directed cyclic graphs, one for each strand, to represent complex structures that can occur during replication. As noted it may be possible to store the sequence strings within either the vertex objects **305** or the edge objects **309** (node and vertex are used synonymously). As used herein, node object **305** and edge object **309** refers to an object created using a computer system.

[0051] FIG. **4** illustrates a computer system **401** suitable for performing methods of the invention. The system **401** includes at least one computer **433**. Optionally, the system **401** may further include one or more of a server computer **409** and a sequencer **455**, which may be coupled to a sequencer computer **451**. Each computer in the system **401** includes a processor coupled to a memory device and at least one input/output device. Thus the system **401** includes at least one processor coupled to a memory subsystem (e.g., a memory device or collection of memory devices **475**). Using those mechanical components, the system **401** is operable to obtain a sequence generated by sequencing nucleic acid from a genome of a patient. The system uses the processor to transform the sequences **303** into the graph **331**.

[0052] Processor refers to any device or system of devices that performs processing operations. A processor will generally include a chip, such as a single core or multi-core chip, to provide a central processing unit (CPU). A processor may be provided by a chip from Intel or AMD. A processor may be any suitable processor such as the microprocessor sold under the trademark XEON E7 by Intel (Santa Clara, CA) or the microprocessor sold under the trademark OPTERON 6200 by AMD (Sunnyvale, CA).

[0053] The memory subsystem **475** contains one or any combination of memory devices. A memory device is a mechanical device that stores data or instructions in a machine-readable format. Memory may include one or more sets of instructions (e.g., software) which, when executed by one or more of the processors of the disclosed computers can accomplish some or all of the methods or functions described herein. Preferably, each computer includes a non-transitory memory device such as a solid state drive, flash drive, disk drive, hard drive, subscriber identity module (SIM) card, secure digital card (SD card), micro SD card, or solid-state drive (SSD), optical and magnetic media, others, or a combination thereof.

[0054] Using the described components, the system **401** is operable to produce a report and provide the report to a user via an input/output device. An input/output device is a mechanism or system for transferring data into or out of a computer. Exemplary input/output devices include a video display unit (e.g., a liquid crystal display (LCD) or a cathode ray tube (CRT)), a printer, an alphanumeric input device (e.g., a keyboard), a cursor control device (e.g., a mouse), a disk drive unit, a speaker, a touchscreen, an accelerometer, a microphone, a cellular radio frequency antenna, and a

network interface device, which can be, for example, a network interface card (NIC), Wi-Fi card, or cellular modem.

[0055] Preferably the graph is stored in the memory subsystem using adjacency lists, which may include pointers to identify a physical location in the memory subsystem **475** where each vertex is stored. In a preferred embodiment, the graph is stored in the memory subsystem **475** using adjacency lists. In some embodiments, there is an adjacency list for each vertex. For discussion of implementations see 'Chapter 4, Graphs' at pages 515-693 of Sedgewick and Wayne, 2011, Algorithms, 4th Ed., Pearson Education, Inc., Upper Saddle River NJ, 955 pages, the contents of which are incorporated by reference and within which pages 524-527 illustrate adjacency lists.

[0056] FIG. **5** shows the use of an adjacency list **501** for each vertex **305**. The system **401** uses a processor to create a graph (such as the graph **331** of FIG. **3**) that includes vertex objects **305** and edge objects **309** through the use of adjacency, i.e., adjacency lists or index free adjacency. Thus, the processor may create the graph **331** using index-free adjacency wherein a vertex **305** includes a pointer to another vertex **305** to which it is connected and the pointer identifies a physical location in on a memory device **475** where the connected vertex is stored. The graph **331** may be implemented using adjacency lists such that each vertex or edge stores a list of such objects that it is adjacent to. Each adjacency list comprises pointers to specific physical locations within a memory device for the adjacent objects.

[0057] In the top part of FIG. **5**, the graph **331** is illustrated in a cartoon-like visual-friendly format. The graph **331** will typically be stored on a physical device of memory subsystem **475** in a fashion that provide for very rapid traversals. In that sense, the bottom portion of FIG. **5** is not cartoon-like and represents that objects are stored at specific physical locations on a tangible part of the memory subsystem **475**. Each node **305** is stored at a physical location, the location of which is referenced by a pointer in any adjacency list **501** that references that node. Each node **305** has an adjacency list **501** that includes every adjacent node in the graph **331**. The entries in the list **501** are pointers to the adjacent nodes.

[0058] In certain embodiments, there is an adjacency list for each vertex and edge and the adjacency list for a vertex or edge lists the edges or vertices to which that vertex or edge is adjacent.

[0059] FIG. **6** shows the use of an adjacency list **601** for each vertex **305** and edge **309**. As shown in FIG. **6**, system **401** creates the graph **331** using an adjacency list **601** for each vertex and edge, wherein the adjacency list **601** for a vertex **305** or edge **309** lists the edges or vertices to which that vertex or edge is adjacent. Each entry in an adjacency list **601** is a pointer to the adjacent vertex or edge.

[0060] Preferably, each pointer identifies a physical location in the memory subsystem at which the adjacent object is stored. In the preferred embodiments, the pointer or native pointer is manipulatable as a memory address in that it points to a physical location on the memory but also dereferencing the pointer accesses intended data. That is, a pointer is a reference to a datum stored somewhere in memory; to obtain that datum is to dereference the pointer. The feature that separates pointers from other kinds of reference is that a pointer's value is interpreted as a memory address, at a low-level or hardware level. The speed and efficiency of the described graph genome engine allows a sequence to be queried against a large-scale genomic reference graph **331** representing millions or billions of bases, using a computer system **401**. Such a graph representation provides means for fast random access, modification, and data retrieval.

[0061] In some embodiments, fast random access is supported and graph object storage are implemented with index-free adjacency in that every element contains a direct pointer to its adjacent elements (e.g., as described in U.S. Pub. 2014/0280360 and U.S. Pub. 2014/0278590, incorporated by reference), which obviates the need for index look-ups, allowing traversals (e.g., as done in the modified SW alignment algorithm described herein) to be very rapid. Index-free adjacency is another example of low-level, or hardware-level, memory referencing for data retrieval (as required in alignment and as particularly pays off in terms of speed gains in the modified, multi-dimensional Smith-Waterman alignment described below). Specifically, index-free adjacency can be implemented such that the pointers contained within elements are in-fact references to a physical location in memory.

[0062] Since a technological implementation that uses physical memory addressing such as native pointers can access and use data in such a lightweight fashion without the requirement of separate index tables or other intervening lookup steps, the capabilities of a given computer, e.g., any modern consumer-grade desktop computer, are extended to allow for full operation of a genomic-scale graph (i.e., a graph **331** that represents all loci in a substantial portion of the subject's genome). Thus storing graph elements (e.g., nodes and edges) using a library of objects with native pointers or other implementation that provides index-free adjacency—i.e., embodiments in which data is retrieved by dereferencing a pointer to a physical location in memory-actually improves the ability of the technology to provide storage, retrieval, and alignment for genomic information since it uses the physical memory of a computer in a particular way.

[0063] While no specific format is required for storage of a graph, FIGS. **5** and **6** are presented to illustrate useful formats. With reference back to FIG. **1**, it is noted that methods of the invention use the stored graph with sequence reads that are obtained from a subject. In some embodiments, sequence reads are obtained as an electronic article, e.g., uploaded, emailed, or FTP transferred from a lab to a system, such as the system **401** of FIG. **4**. In certain embodiments, sequence reads are obtained by sequencing.

[0064] FIG. **7** illustrates obtaining sequence reads **705** from a sample **703**. In certain embodiments, sequence reads are obtained by performing sequencing **713** on a sample **703** from a subject. Sequencing may be by any method known in the art. See, generally, Quail, et al., 2012, A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers, BMC Genomics 13:341. DNA sequencing techniques include classic dideoxy sequencing reactions (Sanger method) using labeled terminators or primers and gel separation in slab or capillary, sequencing by synthesis using reversibly terminated labeled nucleotides, pyrosequencing, 454 sequencing, Illumina/Solexa sequencing, allele specific hybridization to a library of labeled oligonucleotide probes, sequencing by synthesis using allele specific hybridization to a library of labeled clones that is followed by ligation, real

time monitoring of the incorporation of labeled nucleotides during a polymerization step, polony sequencing, and SOLiD sequencing.

[0065] A sequencing technique that can be used includes, for example, use of sequencing-by-synthesis systems sold under the trademarks GS JUNIOR, GS FLX+ and 454 SEQUENCING by 454 Life Sciences, a Roche company (Branford, CT), and described by Margulies, M. et al., Genome sequencing in micro-fabricated high-density picotiter reactors, Nature, 437:376-380 (2005); U.S. Pat. Nos. 5,583,024; 5,674,713: and 5,700,673, the contents of which are incorporated by reference herein in their entirety. 454 sequencing involves two steps. In the first step of those systems, DNA is sheared into blunt-end fragments attached to DNA capture beads and then amplified in droplets. In the second step, pyrosequencing is performed on each DNA fragment in parallel. Addition of one or more nucleotides generates a light signal that is recorded by a CCD camera in a sequencing instrument.

[0066] Another example of a DNA sequencing technique that can be used is SOLiD technology by Applied Biosystems from Life Technologies Corporation (Carlsbad, CA). In SOLiD sequencing, genomic DNA is sheared into fragments, and adaptors are attached to generate a fragment library. Clonal bead populations are prepared in microreactors containing beads, primers, template, and PCR components. Following PCR, the templates are denatured and enriched and the sequence is determined by a process that includes sequential hybridization and ligation of fluorescently labeled oligonucleotides.

[0067] Another example of a DNA sequencing technique that can be used is ion semiconductor sequencing using, for example, a system sold under the trademark ION TORRENT by Ion Torrent by Life Technologies (South San Francisco, CA). Ion semiconductor sequencing is described, for example, in Rothberg, et al., An integrated semiconductor device enabling non-optical genome sequencing, Nature 475:348-352 (2011); U.S. Pubs. 2009/0026082, 2009/0127589, 2010/0035252, 2010/0137143, 2010/0188073, 2010/0197507, 2010/0282617, 2010/0300559, 2010/0300895, 2010/0301398, and 2010/0304982, each incorporated by reference. DNA is fragmented and given amplification and sequencing adapter oligos. The fragments can be attached to a surface. Addition of one or more nucleotides releases a proton (H+), which signal is detected and recorded in a sequencing instrument.

[0068] Another example of a sequencing technology that can be used is Illumina sequencing. Illumina sequencing is based on the amplification of DNA on a solid surface using fold-back PCR and anchored primers. Genomic DNA is fragmented and attached to the surface of flow cell channels. Four fluorophore-labeled, reversibly terminating nucleotides are used to perform sequential sequencing. After nucleotide incorporation, a laser is used to excite the fluorophores, and an image is captured and the identity of the first base is recorded. Sequencing according to this technology is described in U.S. Pub. 2011/0009278, U.S. Pub. 2007/0114362, U.S. Pub. 2006/0024681, U.S. Pub. 2006/0292611, U.S. Pat. Nos. 7,960,120, 7,835,871, 7,232,656, 7,598,035, 6,306,597, 6,210,891, 6,828,100, 6,833,246, and 6,911,345, each incorporated by reference.

[0069] Other examples of a sequencing technology that can be used include the single molecule, real-time (SMRT) technology of Pacific Biosciences (Menlo Park, CA) and nanopore sequencing as described in Soni and Meller, 2007 Clin Chem 53:1996-2001.

[0070] As shown in FIG. **7**, sequencing **713** generates a plurality of reads **705**. Reads according to the invention generally include sequences of nucleotide data anywhere from tens to thousands of bases in length. Reads may be stored in any suitable format such as, for example, FASTA or FASTQ format. FASTA is originally a computer program for searching sequence databases and the name FASTA has come to also refer to a standard file format. See Pearson & Lipman, 1988, Improved tools for biological sequence comparison, PNAS 85:2444-2448. A sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater-than (">") symbol in the first column. FASTQ files are similar to FASTA but further include a line of quality scores. Typically, sequence reads will be obtained **105** in a format such as FASTA, FASTQ, or similar.

[0071] Sequence reads **705** may be directly aligned to a graph, such as the graph **331** of FIG. **3**.

[0072] In some embodiments, the sequence reads **705** are assembled **707** to provide a contig or consensus sequence **709**, which contig or consensus sequence may be used in finding alignments to the graph **331**. Sequence assembly **707** may include any suitable methods known in the art including de novo assembly, reference-guided assembly, others, or combinations thereof. In a preferred embodiment, sequence reads are assembled **707** using graph-based alignment methods. See, e.g., U.S. Pub. 2015/0057946 and U.S. Pub. 2015/0056613, both incorporated by reference. Embodiments of a graph and its use are discussed in greater detail below. The result of assembly **707** is a sequence **709** representing the corresponding portion of the subject's mitochondrial genome. The contig or consensus sequence **709** or one or more of the sequence reads **705** are then mapped to the graph **331** to find an alignment with an optimal score.

[0073] FIG. **8** illustrates finding 129 alignments **801** between the sequence **709** (which may be one or more of the reads **705** or may be a consensus sequence from assembling the reads **705**) and the graph **331**. FIG. **8** also illustrates an optional variant calling **801** step to identify a subject genotype **831**. Using alignment operations of the invention, reads can be rapidly mapped to a graph despite their large numbers or short lengths. Numerous benefits obtain by using a graph as a reference. For example, aligning against a graph is more accurate than aligning against a linear reference and then attempting to adjust one's results in light of other extrinsic information (published mitochondrial genomes are linearized in nature, subject to arbitrary beginning and end). This is primarily because the latter approach enforces an unnatural asymmetry between the sequence used in the initial alignment and other information. Aligning against an object that potentially represents all the relevant physical possibilities is much more computationally efficient than attempting to align against a linear sequence for each physical possibility (the number of such possibilities will generally be exponential in the number of junctions). A modified Smith-Waterman operation for comparing a sequence to a reference graph is provided here as an extension of pairwise alignment methods.

[0074] Pairwise alignment generally involves placing one sequence along part of target, introducing gaps according to

an algorithm, scoring how well the two sequences match, and preferably repeating for various positions along the reference. The best-scoring match is deemed to be the alignment and represents an inference of homology between alignment portions of the sequences. In some embodiments, scoring an alignment of a pair of nucleic acid sequences involves setting values for the scores of substitutions and indels. When individual bases are aligned, a match or mismatch contributes to the alignment score by a substitution probability, which could be, for example, 1 for a match and −0.33 for a mismatch. An indel deducts from an alignment score by a gap penalty, which could be, for example, −1. Gap penalties and substitution probabilities can be based on empirical knowledge or a priori assumptions about how sequences evolve. Their values affect the resulting alignment. Particularly, the relationship between the gap penalties and substitution probabilities influences whether substitutions or indels will be favored in the resulting alignment.

[0075] Stated formally, an alignment represents an inferred relationship between two sequences, x and y. For example, in some embodiments, an alignment A of sequences x and y maps x and y respectively to another two strings x' and y' that may contain spaces such that: (i)|x'|=|y'|; (ii) removing spaces from x' and y' should get back x and y, respectively; and (iii) for any i, x'[i] and y'[i] cannot be both spaces.

[0076] A gap is a maximal substring of contiguous spaces in either x' or y'. An alignment A can include the following three kinds of regions: (i) matched pair (e.g., x'[i]=y'[i]; (ii) mismatched pair, (e.g., x'[i]≠y'[i] and both are not spaces); or (iii) gap (e.g., either x'[i..j] or y'[i..j] is a gap). In certain embodiments, only a matched pair has a high positive score a. In some embodiments, a mismatched pair generally has a negative score b and a gap of length r also has a negative score g+rs where g, s<0. For DNA, one common scoring scheme (e.g. used by BLAST) makes score a=1, score b=−3, g=−5 and s=−2. The score of the alignment A is the sum of the scores for all matched pairs, mismatched pairs and gaps. The alignment score of x and y can be defined as the maximum score among all possible alignments of x and y.

[0077] Any pair may have a score a defined by a 4×4 matrix B of substitution probabilities. For example, B(i,i)=1 and 0<B(i,j)[for i≠j]<1 is one possible scoring system. For instance, where a transition is thought to be more biologically probable than a transversion, matrix B could include B(C,T)=0.7 and B(A,T)=0.3, or other values desired or determined by methods known in the art.

[0078] A pairwise alignment, generally, involves—for sequence Q(query) having m characters and a reference genome T(target) of n characters—finding and evaluating possible local alignments between Q and T. For any 1≤i≤n and 1≤j≤m, the largest possible alignment score of T[h..i] and Q[k..j], where h≤i and k≤j, is computed (i.e. the best alignment score of any substring of T ending at position i and any substring of Q ending at position j). This can include examining all substrings with cm characters, where c is a constant depending on a similarity model, and aligning each substring separately with Q. Each alignment is scored, and the alignment with the preferred score is accepted as the alignment. One of skill in the art will appreciate that there are exact and approximate algorithms for sequence alignment. Exact algorithms will find the highest scoring alignment, but can be computationally expensive. Two well-known exact algorithms are Needleman-Wunsch (J Mol Biol, 48(3):443-453, 1970) and Smith-Waterman (J Mol Biol, 147(1):195-197, 1981: Adv. in Math. 20(3), 367-387, 1976). A further improvement to Smith-Waterman by Gotoh (J Mol Biol, 162(3), 705-708, 1982) reduces the calculation time from O(m^2n) to O(mn) where m and n are the sequence sizes being compared and is more amendable to parallel processing. In the field of bioinformatics, it is Gotoh's modified algorithm that is often referred to as the Smith-Waterman algorithm. Smith-Waterman approaches are being used to align larger sequence sets against larger reference sequences as parallel computing resources become more widely and cheaply available. See, e.g., Amazon's cloud computing resources. All of the journal articles referenced herein are incorporated by reference in their entireties.

[0079] The original Smith-Waterman (SW) algorithm aligns linear sequences by rewarding overlap between bases in the sequences, and penalizing gaps between the sequences. Smith-Waterman also differs from Needleman-Wunsch, in that SW does not require the shorter sequence to span the string of letters describing the longer sequence. That is, SW does not assume that one sequence is a read of the entirety of the other sequence. Furthermore, because SW is not obligated to find an alignment that stretches across the entire length of the strings, a local alignment can begin and end anywhere within the two sequences.

[0080] The original SW algorithm is expressed for an n×m matrix H, representing the two strings of length n and m, in terms of equation (1):

$$H\_k0=H\_01=0(\text{for } 0\le k\le n \text{ and } 0\le 1\le m)$$

$$H\_ij=\max\{H\_(i-1,j-1)+s(a\_i,b\_j),H\_(i-1,j)-W\_\text{in},\ H\_(i,j-1)-W\_del,0\}$$

$$(\text{for } 1\le i\le n \text{ and } 1\le j\le m) \quad (1)$$

[0081] In the equations above, s(ai,bj) represents either a match bonus (when ai=bj) or a mismatch penalty (when ai≠bj), and insertions and deletions are given the penalties Win and Wdel, respectively. In most instances, the resulting matrix has many elements that are zero. This representation makes it easier to backtrace from high-to-low, right-to-left in the matrix, thus identifying the alignment.

[0082] Once the matrix has been fully populated with scores, the SW algorithm performs a backtrack to determine the alignment. Starting with the maximum value in the matrix, the algorithm will backtrack based on which of the three values (Hi−1,j−1, Hi−1,j, or Hij−1) was used to compute the final maximum value for each cell. The backtracking stops when a zero is reached. The optimal-scoring alignment may contain greater than the minimum possible number of insertions and deletions, while containing far fewer than the maximum possible number of substitutions.

[0083] SW or SW-Gotoh may be implemented using dynamic programming to perform local sequence alignment of the two strings, S and A, of sizes m and n, respectively. This dynamic programming employs tables or matrices to preserve match scores and avoid re-computation for successive cells. Each element of the string can be indexed with respect to a letter of the sequence, that is, if S is the string ATCGAA, S[1]=A.

[0084] Instead of representing the optimum alignment as Hi,j(above), the optimum alignment can be represented as B[j,k] in equation (2) below:

$$B[j,k]=\max(p[j,k],i[j,k],d[j,k],0)(\text{for } 0<j\le m,0<k\le n) \quad (2)$$

The arguments of the maximum function, B[j,k], are outlined in equations (3)-(5) below, wherein MISMATCH_PEN, MATCH_BONUS, INSERTION_PEN, DELETION_PEN, and OPENING_PEN are all constants, and all negative except for MATCH_BONUS (PEN is short for PENALTY). The match argument, p[j,k], is given by equation (3), below:

$$p[j,k] = \max(p[j-1,k-1], i[j-1,k-1], d[j-1,k-1]) + \text{MISMATCH\_PEN, if } S[j] \neq A[k] = \max(p[j-1,k-1], i[j-1,k-1], d[j-1,k-1]) + \text{MATCH\_BONUS, if } S[j] = A[k] \quad (3)$$

[0085] the insertion argument i[j,k], is given by equation (4), below:

$$i[j,k]=\max(p[j-1,k]+\text{OPENING\_PEN}, i[j-1,k], d[j-1,k]+\text{OPENING\_PEN})+\text{INSERTION\_PEN} \quad (4)$$

and the deletion argument d[j,k], is given by equation (5), below:

$$d[j,k]=\max(p[j,k-1]+\text{OPENING\_PEN}, i[j,k-1]+\text{OPENING\_PEN}, d[j,k-1])+\text{DELETION\_PEN} \quad (5)$$

[0086] For all three arguments, the [0,0] element is set to zero to assure that the backtrack goes to completion, i.e., p[0,0]=i[0,0]=d[0,0]=0.

[0087] The scoring parameters are somewhat arbitrary, and can be adjusted to achieve the behavior of the computations. One example of the scoring parameter settings (Huang, Chapter 3: Bio-Sequence Comparison and Alignment, ser. Curr Top Comp Mol Biol. Cambridge, Mass.: The MIT Press, 2002) for DNA would be:

[0088] MATCH_BONUS: 10

[0089] MISMATCH_PEN: −20

[0090] INSERTION_PEN: −40

[0091] OPENING_PEN: −10

[0092] DELETION_PEN: −5

[0093] The relationship between the gap penalties (INSERTION_PEN, OPENING_PEN) above help limit the number of gap openings, i.e., favor grouping gaps together, by setting the gap insertion penalty higher than the gap opening cost. Of course, alternative relationships between MISMATCH_PEN, MATCH_BONUS, INSERTION_PEN, OPENING_PEN and DELETION_PEN are possible.

[0094] In some embodiments, the methods and systems of the invention use a modified Smith-Waterman operation that involves a multi-dimensional look-back through a graph, such as the graph **331** of FIG. **3**. Multi-dimensional operations of the invention provide for a "look-back" type analysis of sequence information (as in Smith-Waterman), wherein the look back is conducted through a multi-dimensional space that includes multiple pathways and multiple nodes. The multi-dimensional algorithm can be used to align sequence reads against the graph-type reference. That alignment algorithm identifies the maximum value for Ci,j by identifying the maximum score with respect to each sequence contained at a position on the graph. In fact, by looking "backwards" at the preceding positions, it is possible to identify the optimum alignment across a plurality of possible paths.

[0095] The modified Smith-Waterman operation described here, aka the multi-dimensional alignment, provides exceptional speed when performed in a genomic graph system that employs physical memory addressing (e.g., through the use of native pointers or index free adjacency as discussed above). The combination of multi-dimensional alignment to a graph **331** with the use of spatial memory addresses (e.g., native pointers or index-free adjacency) improves what the computer system is capable of, facilitating whole genomic scale analysis and epigenetic profiling to be performed using the methods described herein.

[0096] The operation includes aligning a sequence, or string, to a graph. For the purpose of defining the algorithm, let S be the string being aligned, and let D be the directed graph to which S is being aligned. The elements of the string, S, are bracketed with indices beginning at 1. Thus, if S is the string ATCGAA, S[1]=A, S[4]=G, etc.

[0097] In certain embodiments, for the graph, each letter of the sequence of a node will be represented as a separate element, d. In a preferred embodiment, node or edge objects contain the sequences and the sequences are stored as the longest-possible string in each object. A predecessor of d is defined as:

[0098] (i) If d is not the first letter of the sequence of its node, the letter preceding d in its node is its (only) predecessor;

[0099] (ii) If d is the first letter of the sequence of its node, the last letter of the sequence of any node (e.g., all exons upstream in the genome) that is a parent of d's node is a predecessor of d.

[0100] The set of all predecessors is, in turn, represented as P[d].

[0101] In order to find the "best" alignment, the algorithm seeks the value of M[j,d], the score of the optimal alignment of the first j elements of S with the portion of the graph preceding (and including) d. This step is similar to finding Hi,j in equation 1 above. Specifically, determining M[j,d] involves finding the maximum of a, i, e, and 0, as defined below:

$$M[j,d]=\max\{a,i,e,0\}$$

where

$$e=\max\{M[j,p^*]+\text{DELETE\_PEN}\} \text{ for } p^* \text{ in } P[d]$$

$$i=M[j-1,d]+\text{INSERT\_PEN}$$

$$a=\max\{M[j-1,p^*]+\text{MATCH\_SCORE}) \text{ for } p^* \text{ in } P[d], \text{ if } S[j]=d;$$

$$\max(M[j-1,p^*]+\text{MISMATCH\_PEN}) \text{ for } p^* \text{ in } P[d], \text{ if } S[j]\neq d \quad (6)$$

[0102] As described above, e is the highest of the alignments of the first j characters of S with the portions of the graph up to, but not including, d, plus an additional DELETE_PEN. Accordingly, if d is not the first letter of the sequence of the node, then there is only one predecessor, p, and the alignment score of the first j characters of S with the graph (up-to-and-including p) is equivalent to M[j,p]+DELETE_PEN. In the instance where d is the first letter of the sequence of its node, there can be multiple possible predecessors, and because the DELETE_PEN is constant, maximizing [M[j, p*]+DELETE_PEN] is the same as choosing the predecessor with the highest alignment score with the first j characters of S.

[0103] In equation (6), i is the alignment of the first j−1 characters of the string S with the graph up-to-and-including d, plus an INSERT_PEN, which is similar to the definition of the insertion argument in SW (see equation 1).

[0104] Additionally, a is the highest of the alignments of the first j characters of S with the portions of the graph up to, but not including d, plus either a MATCH_SCORE (if the jth character of S is the same as the character d) or a MISMATCH_PEN (if the jth character of S is not the same as the character d). As with e, this means that if d is not the first letter of the sequence of its node, then there is only one predecessor, i.e., p. That means a is the alignment score of the first j−1 characters of S with the graph (up-to-and-including p), i.e., M[j−1,p], with either a MISMATCH_PEN or MATCH_SCORE added, depending upon whether d and the jth character of S match. In the instance where d is the first letter of the sequence of its node, there can be multiple possible predecessors. In this case, maximizing {M[j, p*]+ MISMATCH_PEN or MATCH_SCORE} is the same as choosing the predecessor with the highest alignment score with the first j−1 characters of S (i.e., the highest of the candidate M[j−1,p*] arguments) and adding either a MISMATCH_PEN or a MATCH_SCORE depending on whether d and the jth character of S match.

[0105] Again, as in the SW algorithm, the penalties, e.g., DELETE_PEN, INSERT_PEN, MATCH_SCORE and MISMATCH_PEN, can be adjusted to encourage alignment with fewer gaps, etc.

[0106] As described in the equations above, the operation finds the optimal (e.g., maximum) value for the sequence **709** by calculating not only the insertion, deletion, and match scores for that element, but looking backward (against the direction of the graph) to any prior nodes on the graph to find a maximum score.

[0107] FIG. **9** shows the matrices that represent the comparison. The modified Smith-Waterman operation of the invention identifies the highest score and performs a backtrack to identify the proper alignment of the sequence. See, e.g., U.S. Pub. 2015/0057946 and U.S. Pub. 2015/0056613, both incorporated by reference. Systems and methods of the invention can be used to provide a report that identifies a modified base at the position within the genome of the subject. Other information may be found in Kehr et al., 2014, Genome alignment with graph data structures: a comparison, BMC Bioinformatics 15:99, incorporated by reference.

[0108] FIG. **10** illustrates a report **1001** that identifies heteroplasmy within an organism. Each successful match between a sequence **709** and a path in the graph **331** can increment a count for that path. Thus if all of the sequences **709** from a subject match to the same path in the graph **331**, that path would be reported as representing the mitochondrial genome of the subject and the subject would be reported as having no heteroplasmy detected. However, if 70% of the sequences **709** mapped to a first one of the paths, and 25% of the sequences mapped to a second one of the paths, the subject's two distinct mitochondrial forms could be reported and quantified.

[0109] A report **1001** that identifies heteroplasmy in the subject may do so, wholly or in part, by including a graph **1031** that illustrates the heteroplasmy. The graph **1031** may be labelled in such a way as to give the entire sequence or the graph **1031** may simply diagram the divergences presented by the heteroplasmy in the subject. The graph may have branch labels showing the relative amounts of each form as present in the sample. Where the graph **1031** is presented on-screen, the system can allow a user to see a simplified form and alternatively to retrieve complete sequences, e.g., via a click or a zoom operation.

[0110] Additionally or alternatively, methods of the invention may be used to identify a subject.

[0111] FIG. **11** shows a report **1101** that provides the identity of a subject. This may be provided in different ways. For example, where there are known candidates, the graph **331** may be built from the sequences of mitochondrial genomes from maternal-line relatives of each candidate, including reflecting known heteroplasmy. A record is kept of which branches or paths of the graph correspond to which candidate (with some branches corresponding to multiple candidates). Reads from the mtDNA of an unknown individual are aligned to the graph. The report **1101** may identify the branches to which the reads have been aligned and the corresponding candidates.

[0112] The report **1101** may include a list of the candidates corresponding to the branches to which the reads aligned, along with the percentage of reads aligned (or percentage of nucleotides matched) to each. In some embodiments, the branches are weighted (and thus candidates) according to the "delta" between the branch and the next-best alignment for a given read.

[0113] In yet other aspects and embodiments, systems and methods of the invention may be used to provide a report that includes a description of mutations/variants or other significant features in a subject's mitochondrial genome.

[0114] FIG. **12** shows a report **1201** that includes a description of variants in a subject's mitochondrial genome. Optionally, systems and methods of the invention may be used for variant calling **807** to produce genotype information **831** about the subject's genome. The variant calling can include aligning sequence reads to the graph and reporting SNP alleles in a format such as a Sequence Alignment Map (SAM) or a Variant Call Format (VCF) file. Some background may be found in Li & Durbin, 2009, Fast and accurate short read alignment with Burrows-Wheeler Transform. Bioinformatics 25:1754-60 and McKenna et al., 2010, The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data, Genome Res 20(9):1297-1303, the contents of each of which are incorporated by reference. Variant calling **831** produces results ("variant calls") that may be stored as a sequence alignment map (SAM) or binary alignment map (BAM) file-comprising an alignment string (the SAM format is described, e.g., in Li, et al., The Sequence Alignment/Map format and SAMtools, Bioinformatics, 2009, 25(16):2078-9). Additionally or alternatively, output from the variant calling may be provided in a variant call format (VCF) file, e.g., in report **1201**. A typical VCF file will include a header section and a data section. The header contains an arbitrary number of meta-information lines, each starting with characters '##', and a TAB delimited field definition line starting with a single '#' character. The field definition line names eight mandatory columns and the body section contains lines of data populating the columns defined by the field definition line. The VCF format is described in Danecek et al., 2011, The variant call format and VCFtools, Bioinformatics 27(15):2156-2158. Further discussion may be found in U.S. Pub. 2013/0073214; U.S. Pub. 2013/0345066; U.S. Pub. 2013/0311106; U.S. Pub. 2013/0059740; U.S. Pub. 2012/0157322; U.S. Pub. 2015/0057946 and U.S. Pub. 2015/0056613, each incorporated by reference. Systems and methods of the invention may be used to describe and report

particular mutations that have significant influences in human diseases such as cancer, such as those variants described in van Gisbergen et al., 2015, How do changes in the mtDNA and mitochondrial dysfunction influence cancer and cancer therapy?, Mutat Res 764:16-30, incorporated by reference. Thus use of systems and methods of the invention provide a product that facilitates medical genetics and patient counseling. A physician may use a report **1201** provided by the system to determine a medical course of action or counsel a patient on health and wellness issues.

INCORPORATION BY REFERENCE

[0115] References and citations to other documents, such as patents, patent applications, patent publications, journals, books, papers, web contents, have been made throughout this disclosure. All such documents are hereby incorporated herein by reference in their entirety for all purposes.

EQUIVALENTS

[0116] Various modifications of the invention and many further embodiments thereof, in addition to those shown and described herein, will become apparent to those skilled in the art from the full contents of this document, including references to the scientific and patent literature cited herein. The subject matter herein contains important information, exemplification and guidance that can be adapted to the practice of this invention in its various embodiments and equivalents thereof.

1-20. (canceled)

21. A system, comprising:

at least one processor, and

at least one non-transitory memory storing processor-executable instructions that, when executed by the at least one processor, cause the at least one processor to perform a method for analyzing a mitochondrial genome comprising a plurality of mitochondrial sequences, the method comprising:

generating, in at least one non-transitory memory, a mitochondrial DNA (mtDNA) reference graph representing at least some of the plurality of mitochondrial sequences, the mtDNA reference graph comprising nodes and edges connecting the nodes, the nodes including a first node and a second node, wherein:

the first node is stored as a first object in the at least one non-transitory memory,

the second node is stored as a second object in the at least one non-transitory memory, and

a first edge of the edges is stored as a first pointer from the first object to the second object in the at least one non-transitory memory,

obtaining a plurality of sequence reads from a biological sample previously obtained from a subject;

aligning one or more sequence reads of the plurality of sequence reads to the mtDNA reference graph in the at least one non-transitory memory at least in part by determining alignment scores between the one or more sequence reads and symbol strings associated with the nodes in the mtDNA reference graph, the symbol strings representing sequences of one or more nucleotides of the mitochondrial genome; and

identifying, based on results of the aligning, one or more mitochondrial sequences of the plurality of mitochondrial sequences to which the one or more sequence reads align.

22. The system of claim **21**, further comprising identifying mitochondrial heteroplasmy in the subject based on the identified one or more mitochondrial sequences to which the one or more sequence reads align.

23. The system of claim **21**, wherein the plurality of sequence reads correspond to at least a portion of a D-loop of mitochondria of the subject.

24. The system of claim **21**, wherein the plurality of mitochondrial sequences is obtained from relatives of the subject.

25. The system of claim **21**, wherein the subject is an unknown subject, and wherein the method further comprises:

determining an identity of the unknown subject based on the identified one or more mitochondrial sequences to which the one or more sequence reads align.

26. The system of claim **25**, wherein the mtDNA reference graph represents variations in a mitochondrial genome of a maternal-line individual related to the unknown subject, variations in a hyper-variable region of the mitochondrial genome, or a combination thereof.

27. The system of claim **21**,

wherein the first object comprises a first symbol string representing a first sequence of one or more nucleotides of the mitochondrial genome,

wherein the second object comprises a second symbol string representing a second sequence of one or more nucleotides of the mitochondrial genome, and

wherein determining an alignment score for the second node comprises, for a first symbol in the second symbol string associated with the second node, determining the alignment score for the second node based on an alignment score associated with the first node.

28. The system of claim **27**, wherein aligning a second read of the one or more sequence reads against the mtDNA reference graph comprises:

determining a first alignment score between a portion of the sequence read and a portion of the mtDNA reference graph preceding and including a symbol in the second symbol string.

29. The system of claim **28**, wherein determining the first alignment score between the portion of the sequence read and the portion of the mtDNA reference graph preceding and including the symbol in the second symbol string comprises:

determining the first alignment score based on an alignment score associated with the first node, if and only if the symbol comprises the first symbol of the second symbol string.

30. The system of claim **21**,

wherein the nodes of the mtDNA reference graph further comprise a third node,

wherein the third node is stored as a third object in the at least one non-transitory memory, and

wherein a second edge of the edges is stored as a second pointer from the third object to the second object in the at least one non-transitory memory.

31. The system of claim **21**, wherein the mtDNA reference graph is a directed acyclic graph (DAG).

32. At least one non-transitory memory storing processor-executable instructions that, when executed by at least one

processor, cause the at least one processor to perform a method for analyzing a mitochondrial genome comprising a plurality of mitochondrial sequences, the method comprising:

generating, in at least one non-transitory memory, a mitochondrial DNA (mtDNA) reference graph representing at least some of the plurality of mitochondrial sequences, the mtDNA reference graph comprising nodes and edges connecting the nodes, the nodes including a first node and a second node, wherein:

the first node is stored as a first object in the at least one non-transitory memory,

the second node is stored as a second object in the at least one non-transitory memory, and

a first edge of the edges is stored as a first pointer from the first object to the second object in the at least one non-transitory memory,

obtaining a plurality of sequence reads from a biological sample previously obtained from a subject;

aligning one or more sequence reads of the plurality of sequence reads to the mtDNA reference graph in the at least one non-transitory memory at least in part by determining alignment scores between the one or more sequence reads and symbol strings associated with the nodes in the mtDNA reference graph, the symbol strings representing sequences of one or more nucleotides of the mitochondrial genome; and

identifying, based on results of the aligning, one or more mitochondrial sequences of the plurality of mitochondrial sequences to which the one or more sequence reads align.

33. The at least one non-transitory memory of claim **32**, further comprising identifying mitochondrial heteroplasmy in the subject based on the identified one or more mitochondrial sequences to which the one or more sequence reads align.

34. The at least one non-transitory memory of claim **32**, wherein the plurality of sequence reads correspond to at least a portion of a D-loop of mitochondria of the subject.

35. The at least one non-transitory memory of claim **32**, wherein the plurality of mitochondrial sequences is obtained from relatives of the subject.

36. The at least one non-transitory memory of claim **32**, wherein the subject is an unknown subject, and wherein the method further comprises:

determining an identity of the unknown subject based on the identified one or more mitochondrial sequences to which the one or more sequence reads align.

37. The at least one non-transitory memory of claim **36**, wherein the mtDNA reference graph represents variations in a mitochondrial genome of a maternal-line individual related to the unknown subject, variations in a hyper-variable region of the mitochondrial genome, or a combination thereof.

38. The at least one non-transitory memory of claim **32**,

wherein the first object comprises a first symbol string representing a first sequence of one or more nucleotides of the mitochondrial genome,

wherein the second object comprises a second symbol string representing a second sequence of one or more nucleotides of the mitochondrial genome, and

wherein determining an alignment score for the second node comprises, for a first symbol in the second symbol string associated with the second node, determining the alignment score for the second node based on an alignment score associated with the first node.

39. The at least one non-transitory memory of claim **38**, wherein aligning a second read of the one or more sequence reads against the mtDNA reference graph comprises:

determining a first alignment score between a portion of the sequence read and a portion of the mtDNA reference graph preceding and including a symbol in the second symbol string.

40. The at least one non-transitory memory of claim **39**, wherein determining the first alignment score between the portion of the sequence read and the portion of the mtDNA reference graph preceding and including the symbol in the second symbol string comprises:

determining the first alignment score based on an alignment score associated with the first node, if and only if the symbol comprises the first symbol of the second symbol string.

* * * * *