

METHODS AND SYSTEMS FOR COPY NUMBER VARIANT DETECTION

CROSS REFERENCE TO RELATED PATENT APPLICATION

[0001] This application claims priority to U.S. Non-Provisional Application No. 14/714,949 filed May 18, 2015, here incorporated by reference in its entirety.

BACKGROUND

[0001] Genomic sequencing is an effective tool to discover the genetic basis of Mendelian disorders. Analysis of genomic sequences has revealed the existence of copy number variants (CNVs) (e.g., the number of copies of a particular gene in the genotype of an individual). CNVs may have important roles in human disease and/or drug response. However, calling CNVs from genomic sequence data (e.g., exome sequence data) is challenging. Current solutions detect CNVs from human sequencing read depth, but are not well-suited for large population studies on the order of tens or hundreds of thousands of exomes. Their limitations, among others, include being difficult to integrate into automated variant calling pipelines and being ill-suited for detecting common variants. These and other shortcomings are addressed in the present disclosure.

SUMMARY

[0002] It is to be understood that both the following general description and the following detailed description are exemplary and explanatory only and are not restrictive. Methods and systems for determining copy number variants are disclosed. An example method can comprise receiving, by a computing device, a sample coverage data set comprising a plurality of genomic sequences obtained from sequencing of nucleic acid samples of a subject, and sample sequencing quality control (SSQC) metrics; grouping, by the computing device, sets of sequencing quality control (SQC) metrics into a multidimensional tree data structure according to similarity, wherein each set of SQC metrics is associated with a respective reference coverage data set that comprises a plurality of genomic regions and read depths; selecting a reference panel of the reference coverage data sets using the multidimensional tree data

structure, wherein the selected reference coverage data sets have SQC metrics similar to the SSQC metrics; normalizing, by the computing device, the sample coverage data and the reference panel, and fitting, by the computing device, the normalized reference panel to a mixture model at each of the plurality of genomic regions to generate an expected coverage distribution at each of the plurality of genomic regions identifying one or more copy number variants (CNVs) by using the computing device to compare, according to a Hidden Markov Model (HMM), the normalized sample coverage data set to the expected coverage distribution at each of the plurality of genomic regions from the mixture models; and outputting, by the computing device, the one or more CNVs.

[0003] In an aspect, another example method can comprise providing, by computing device, sample coverage data comprising a plurality of genomic sequences obtained from sequencing of nucleic acid samples of a subject and sample sequencing quality control (SSQC) metrics; and receiving an indication of reference coverage data, wherein the reference coverage data is selected by: grouping, by the computing device, sets of sequencing quality control (SQC) metrics into a multidimensional tree data structure according to similarity, wherein each set of SQC metrics is associated with a respective reference coverage data set that comprises a plurality of genomic regions and read depths; selecting a reference panel of the reference coverage data sets using the multidimensional tree data structure, wherein the selected reference coverage data sets have SQC metrics similar to the SSQC metrics.

The method further comprises selecting one or more filters to apply to the sample coverage data and the reference panel to normalize the sample coverage data and the reference panel; requesting fitting of the normalized reference panel to a mixture model at each of the plurality of genomic regions to generate an expected coverage distribution at each of the plurality of genomic region. The method can comprise requesting identifying one or more copy number variants (CNVs) by using the computing device to compare, according to a Hidden Markov Model (HMM) the normalized sample coverage data set to the expected coverage distribution at each of the plurality of genomic region from the mixture models. The method can further comprise receiving an indication of the one or more CNVs.

[0004] There is provided another example method can comprise receiving sample coverage data comprising a plurality of genomic regions, retrieving one or more metrics for the sample coverage data, applying a sample grouping technique to the sample coverage data and reference coverage data to select a subset of the reference coverage data, normalizing the sample coverage data comprising the plurality of genomic regions, and fitting a mixture model to the normalized sample coverage data based on the subset of the reference coverage data. The method can comprise identifying one or more copy number variants according to a Hidden Markov Model (HMM) based on the normalized sample coverage data and the fitted mixture model. The method can comprise outputting the one or more copy number variants.

[0005] Additional advantages will be set forth in part in the description which follows or may be learned by practice. The advantages will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate embodiments and together with the description, serve to explain the principles of the methods and systems:

Figure 1 is flowchart illustrating an example CNV-calling pipeline;

Figure 2 is a flowchart illustrating an example method for determining copy number variants;

Figure 3 shows a graph illustrating the relationship of GC content and coverage;

Figure 4 is a graph illustrating normalized coverage of various exons;

Figure 5 is a flow chart illustrating another example method for estimating copy number variants;

Figure 6 is a flow chart illustrating yet another example method for estimating copy

number variants;

Figure 7 is a block diagram illustrating an exemplary operating environment for performing the disclosed methods;

Figure 8 compares the RAM usage of CLAMMS vs. other algorithms;

Figure 9 is a table illustrating performance metrics for CNV calls on the CEPH pedigree;

Figure 10 shows CLAMMS and XHMM CNV calls compared to PennCNV gold-standard;

Figure 11 shows a table illustrating Rare CNV TaqMan Validations;

Figure 12 shows a table illustrating common CNV TaqMan Validations;

Figure 13 is a graph comparing CLAMMS and TaqMan copy number predictions for the LILRA3 common variant locus;

Figure 14 is a graph comparing CLAMMS and TaqMan copy number predictions for the LILRA3 common variant locus; and

Figure 15 is an example output.

DETAILED DESCRIPTION

[0007] Before the present methods and systems are disclosed and described, it is to be understood that the methods and systems are not limited to specific methods, specific components, or to particular implementations. It is also to be understood that the terminology used herein is for the purpose of describing particular embodiments only and is not intended to be limiting.

[0008] As used in the specification and the appended claims, the singular forms “a,” “an” and “the” include plural referents unless the context clearly dictates otherwise. Ranges may be expressed herein as from “about” one particular value, and/or to “about” another particular value. When such a range is expressed, another embodiment includes from the one particular value and/or to the other particular value. Similarly, when values are expressed as approximations, by use of the antecedent “about,” it will be understood that the particular value forms another embodiment. It will be further understood that the endpoints of each of the ranges are significant both in relation to the other endpoint, and independently of the other endpoint.

[0009] “Optional” or “optionally” means that the subsequently described event or circumstance may or may not occur, and that the description includes instances

where said event or circumstance occurs and instances where it does not.

[0010] Throughout the description and claims of this specification, the word “comprise” and variations of the word, such as “comprising” and “comprises,” means “including but not limited to,” and is not intended to exclude, for example, other components, integers or steps. “Exemplary” means “an example of” and is not intended to convey an indication of a preferred or ideal embodiment. “Such as” is not used in a restrictive sense, but for explanatory purposes.

[0011] It is understood that the disclosed method and compositions are not limited to the particular methodology, protocols, and reagents described as these may vary. It is also to be understood that the terminology used herein is for the purpose of describing particular embodiments only, and is not intended to limit the scope of the present methods and system which will be limited only by the appended claims.

[0012] Unless defined otherwise, all technical and scientific terms used herein have the same meanings as commonly understood by one of skill in the art to which the disclosed method and compositions belong. Although any methods and materials similar or equivalent to those described herein can be used in the practice or testing of the present method and compositions, the particularly useful methods, devices, and materials are as described. Publications cited herein and the material for which they are cited are hereby specifically incorporated by reference. Nothing herein is to be construed as an admission that the present invention is not entitled to antedate such disclosure by virtue of prior invention. No admission is made that any reference constitutes prior art. The discussion of references states what their authors assert, and applicants reserve the right to challenge the accuracy and pertinency of the cited documents. It will be clearly understood that, although a number of publications are referred to herein, such reference does not constitute an admission that any of these documents forms part of the common general knowledge in the art.

[0013] Disclosed are components that can be used to perform the disclosed methods and systems. These and other components are disclosed herein, and it is understood that when combinations, subsets, interactions, groups, etc. of these components are disclosed that while specific reference of each various individual and collective combinations and permutation of these may not be explicitly disclosed, each is specifically contemplated and described herein, for all methods and systems. This applies to all aspects of this application including, but not limited to, steps in disclosed methods. Thus, if there are a variety of additional steps that can be

performed it is understood that each of these additional steps can be performed with any specific embodiment or combination of embodiments of the disclosed methods.

[0014] The present methods and systems may be understood more readily by reference to the following detailed description of preferred embodiments and the examples included therein and to the Figures and their previous and following description.

[0015] As will be appreciated by one skilled in the art, the methods and systems may take the form of an entirely hardware embodiment, an entirely software embodiment, or an embodiment combining software and hardware aspects. Furthermore, the methods and systems may take the form of a computer program product on a computer-readable storage medium having computer-readable program instructions (e.g., computer software) embodied in the storage medium. More particularly, the present methods and systems may take the form of web-implemented computer software. Any suitable computer-readable storage medium may be utilized including hard disks, CD-ROMs, optical storage devices, or magnetic storage devices.

[0016] Embodiments of the methods and systems are described below with reference to block diagrams and flowchart illustrations of methods, systems, apparatuses and computer program products. It will be understood that each block of the block diagrams and flowchart illustrations, and combinations of blocks in the block diagrams and flowchart illustrations, respectively, can be implemented by computer program instructions. These computer program instructions may be loaded onto a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions which execute on the computer or other programmable data processing apparatus create a means for implementing the functions specified in the flowchart block or blocks.

[0017] These computer program instructions may also be stored in a computer-readable memory that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including computer-readable instructions for implementing the function specified in the flowchart block or blocks. The computer program instructions may also be loaded onto a computer or other programmable data processing apparatus to cause a series of operational steps to be performed on the computer or other programmable apparatus to produce

a computer-implemented process such that the instructions that execute on the computer or other programmable apparatus provide steps for implementing the functions specified in the flowchart block or blocks.

[0018] Accordingly, blocks of the block diagrams and flowchart illustrations support combinations of means for performing the specified functions, combinations of steps for performing the specified functions and program instruction means for performing the specified functions. It will also be understood that each block of the block diagrams and flowchart illustrations, and combinations of blocks in the block diagrams and flowchart illustrations, can be implemented by special purpose hardware-based computer systems that perform the specified functions or steps, or combinations of special purpose hardware and computer instructions.

[0019] The present methods and systems are directed to CNV detection (e.g., identification, prediction, estimation). Some aspects of the present methods and systems can be referred to as “Copy number estimation using Lattice-aligned Mixture Models (CLAMMS).” Detecting copy number variants with whole exome sequencing (WES) data can be challenging because CNV breakpoints are likely to fall outside of the exome. The present methods and systems can utilize read depths within the CNV. Such read depths can be linearly correlated to copy number state. However, depth-of-coverage can be subject to both systematic biases (e.g., often related to sequence GC-content) and stochastic volatility (e.g., which is exacerbated by variation in input DNA quality). The present methods and systems can normalize coverage data to correct for systematic biases and characterize the expected coverage profile given diploid copy number so that true CNVs can be distinguished from noise. Such normalization can comprise, for example, comparing each sample’s coverage data to data from a “reference panel” (e.g., reference coverage data) of similarly-sequenced samples. Variability in sample preparation and sequencing procedures can result in additional coverage biases that are commonly referred to as “batch effects.”

[0020] In an aspect, the present method and systems can identify CNVs based on the use of both mixture models and Hidden Markov Models (HMM). For example, mixture models can be fit based on reference coverage data determined using a sample grouping algorithm, such as k-nearest neighbor algorithm. Information from the mixture models can be input into an HMM for identification of CNVs.

[0021] **FIG. 1** is flowchart illustrating an example CNV-calling pipeline. A reference panel of coverage data (e.g., reference coverage data comprising one or more genomic capture regions) can be selected for each sample (e.g., sample coverage data comprising one or more genomic capture regions) based on a plurality of metrics (e.g., sequencing Quality Control (QC) metrics) using a sample grouping technique. The sample grouping technique can comprise a technique (e.g., algorithm) for grouping samples by similarity. Examples of sample grouping techniques that can be used include, but are not limited to, a decision tree, a support-vector machine, a k -nearest neighbors (knn) algorithm, a Naïve Bayes algorithm, a CART (Classification and Regression Trees) algorithm, and/or the like. For example, a kNN algorithm can comprise generating a k -d tree data structure. The reference coverage data can be selected by inserting the sample coverage data (e.g., or metrics associated with the sample coverage data) into the k -d tree structure and identifying a predetermined number of nearest neighbors (e.g., 10, 100, 1000, 10000, etc...). After selecting reference coverage data, samples can be processed in parallel. Sample-level analysis (right panel) includes normalizing coverage, fitting coverage distributions with a mixture model, and generating calls from an HMM

[0022] In an aspect, an example implementation of the present methods and systems is disclosed in **FIG. 1**. As shown in the left panel, reference coverage data (e.g., pulled from a sample set) can be used as part of the sample grouping technique. Though a k -nearest neighbor algorithm that utilizes a k -d tree is used to illustrate the sample grouping technique, it should be appreciated that other sample grouping techniques can be applied (e.g., any appropriate clustering, grouping, and/or classification algorithm). The k -d tree can comprise a multidimensional search tree for points in k dimensional space. For example, a plurality of metrics of the reference coverage data can be used by the sample grouping technique. For example, the plurality of metrics of the reference coverage data can be used to build the k -d tree. The plurality of metrics can comprise, for example, sequencing quality control (QC) metrics, sample metadata, ancestry-based values, sequence-similarity scores, and/or any metric that captures sample-level variability. For example, in the case of sequencing QC metrics, seven QC metrics can be used. As an illustration, the sequencing QC metrics can comprise GCDROPOUT, ATDROPOUT, MEANINSERTSIZE, ONBAITVSSELECTED, PCTPFUQREADS, PCTTARGETBASES10X, PCTTARGETBASES50X, and/or the like. The

sequencing QC metrics can be scaled (e.g., by applying a linear transform) and processed to build the k -d tree.

[0023] A plurality of metrics (e.g., sequencing QC metrics) for sample coverage data can also be scaled and inserted into the k -d tree. The k -d tree can then be used to perform a nearest neighbor search to identify the nearest neighbors to the sample coverage data. Any number of nearest neighbors in the reference coverage data can be identified (e.g., 10, 100, 1000, 10000, etc...). The desired number of nearest neighbors can be used to form selected reference coverage data (e.g., a subset of the reference coverage data). The present methods and systems can address data heterogeneity by selecting custom reference coverage data for each sample. By way of example, a distance metric between samples (e.g., reference coverage data) can be defined based the seven sequencing QC metrics described above. For example, the sequencing QC metrics can be determined, selected, received, and/or the like from a sequencing tool, such as Picard. Each newly sequenced sample can be added to a k -d tree in this metric space. CNVs can be called using selected reference coverage data comprising the individual sample's k (e.g., 100) nearest neighbors. The k -nearest neighbors can be found using any nearest neighbors algorithm, such as a k -d tree algorithm or other sample grouping technique.

[0024] As shown in the right panel, sample coverage data (e.g., sample i) can be selected from a sample set. The sample coverage data can be normalized to correct for GC-amplification bias and overall average depth-of-coverage. In another aspect, sample coverage data can be filtered. For example, sample coverage data can be filtered based on a level of GC content, based on a mappability score, based on a measure of central tendency of read coverage, based on occurrence of a calling window in a multi-copy duplication exome capture region, combinations thereof and the like. For example, read depths in low-mappability regions may not accurately represent the sequence dosage in the genome.

[0025] Once the sample coverage data has been normalized, the selected reference coverage data (nearest neighbors) can be used to fit a finite mixture model for one or more (or each) genomic (e.g., exome) capture regions in the sample coverage data. A finite mixture model can comprise a combination of two or more probability density functions. Finite mixture models can comprise one or more components, such as: N random variables corresponding to observations, each assumed to be distributed according to a mixture of K components, with each component belonging

to the same parametric family of distributions but with different parameters; N corresponding random latent variables specifying the identity of the component of the mixture model of each observation, each distributed according to a K -dimensional categorical distribution; a set of K mixture weights, each of which is a probability (a real number between 0 and 1 inclusive), all of which sum to 1; a set of K parameters, each specifying the parameter of the corresponding component of the mixture model. In some aspects, a parameter can comprise a set of parameters. In the present methods and systems, each component of the mixture model can model the expected distribution of coverage across samples for a particular integer copy number state. Accommodations can be made to handle homozygous deletions and sex chromosomes.

[0026] In an aspect, an expectation-maximization (EM) algorithm can be used to fit the finite mixture model. The EM algorithm is a general method for finding maximum likelihood estimates when there are missing values or latent variables. The EM algorithm can be an iterative algorithm. The iterations can alternate between performing an expectation (E) step, which can generate a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which can computer parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates can then be used to determine the distribution of the latent variables in the next E step.

[0027] In an aspect, CNVs can be called for the sample coverage data using a Hidden Markov Model (HMM). For example, the individual sample's normalized coverage values for each region can be the input sequence to the HMM. Emission probabilities of the HMM can be based on the trained (e.g., fit, adapted) mixture models. Transition probabilities of the HMM can be similar to those used by other models, such as XHMM, incorporated herein by reference. Mixture models allow for copy number polymorphic loci to be handled naturally, while the HMM incorporates the prior expectation that nearby anomalous signals are more likely to be part of a single CNV than multiple small CNVs. The present methods and systems can integrate mixture models and HMMs into a single probabilistic model.

[0028] **FIG. 2** is a flowchart illustrating an example method **200** for determining copy number variants. In an aspect, the present method and system can be configured for analyzing sample coverage data comprising a plurality of genomic

regions to detect CNVs. At step **202**, a sample grouping technique can be applied to select reference coverage data. For example, the sample grouping technique can comprise a technique (e.g., algorithm) for grouping samples by similarity. Applying a sample grouping technique to select reference coverage data can comprise receiving a plurality of metrics for the sample coverage data. A distance metric between the sample coverage data and the reference coverage data can be defined based on the plurality of metrics. The reference coverage data can be selected (e.g., for each sample) based on the distance metric. The sample grouping technique can comprise a grouping algorithm, clustering algorithm, classification algorithm, and/or the like. For example, the sample grouping technique can comprise a decision tree, a support-vector machine, a k-Nearest Neighbors (knn) algorithm, a Naïve Bayes algorithm, a CART (Classification and Regression Trees) algorithm, and/or the like. For example, applying the sample grouping technique to select reference coverage data the method can comprise scaling a plurality of metrics associated with the reference coverage data, generating a k-d tree based on the scaled plurality of metrics associated with the reference coverage data, scaling a plurality of metrics associated with the sample coverage data, adding the sample coverage data to the k-d tree based on the scaled plurality of metrics associated with the sample coverage data, identifying a predetermined number of nearest neighbors to the sample coverage data as the selected reference coverage data, and/or the like.

[0029] Application of the sample grouping technique to select reference coverage data is described in further detail as follows. Systematic coverage biases that arise due to variability in sequencing conditions are commonly referred to as “batch effects.” In an aspect, the present methods and systems can be configured to use a custom reference panel (e.g., selected reference coverage data) approach to correct for batch effects. For example, instead of comparing sample coverage data based on the sample’s coverage profiles— a high-dimensional space— the present methods and systems can be configured to consider a low-dimensional metric space based on sequencing quality control (QC) metrics. For example, the sequencing QC metrics can comprise seven sequencing QC metrics. The sequencing QC metrics can comprise sequencing QC metrics from a sequencing tool, such as Picard. Working in this low-dimensional space allows for improved scalability. For example, samples can be indexed ahead-of-time (e.g., using any appropriate indexing and/or search algorithm). As a further example, samples can be indexed ahead-of-time using a *k*-

nearest neighbor algorithm. For example, the k -nearest neighbor algorithm can use k -d tree structure that allows for fast nearest-neighbor queries and uses a minimal amount of RAM.

[0030] As an illustration, an example variant-calling pipeline can be configured to proceed as follows:

1. Query a laboratory information management system to retrieve seven Picard sequencing quality control metrics for each sample:
GCDROPOUT, ATDROPOUT, MEANINSERTSIZE,
ONBAITVSSELECTED, PCTPFUQREADS,
PCTTARGETBASES10X, and PCTTARGETBASES50X.
2. Insert each sample's QC-metric vector k -d tree data structure, after applying a linear transform to scale each metric into the range [0, 1] (e.g., scaled value = [raw value - min] / [max - min])
3. In parallel, for each sample:
 - (a) Compute depth-of-coverage from the BAM file using samtools and run CLAMMS' within-sample normalization step.
 - (b) Train CLAMMS models using the sample's 100 nearest neighbors in the k -d tree.
 - (c) Call CNVs using these models.

[0031] In an aspect, larger values of k can decrease variance in the statistical inference of the mixture model parameters but increase bias. A default k value can be selected according to specific applications. In some scenarios, a default value $k = 100$ can provide the best bias-variance trade-off. The pipeline can be extended to run via a network (e.g., web interface) if the k -d tree is stored in a database. In some scenarios, such as small-scale studies, the present methods and systems can also be used without having to compute QC metrics. For example, samples can be manually assigned to batches based on a PCA plot of a sample-by-exon coverage matrix. A separate set of models can be trained for each batch and used to call CNVs for samples in that batch.

[0032] In an aspect, the present methods and systems can divide the plurality of genomic regions of the sample coverage data into one or more calling windows (e.g., a plurality of calling windows). For example, the present methods and systems can divide genomic (e.g., exome) capture regions into equally sized calling windows. For example, genomic capture regions that are greater than or equal to

1000 bp long can be divided into equally-sized 500-1000 base pair (bp) calling windows. The present methods and systems can be configured to divide genomic regions into calling windows such that CNVs that partially overlap long exons can be detection. Examples of genes with extraordinarily long exons include *AHNAK*, *TTN*, and several Mucins. In an aspect, only genomic regions of the plurality of genomic regions larger than a predetermined size may be divided, for example, larger than 999 bases. It should be noted that any other appropriate number of bases can be used.

[0033] In an aspect, the methods and systems can optionally comprise filtering the sample coverage data. Filtering can be accomplished prior to step **202**, during step **202**, and/or during other steps of the method **200**. Filtering the sample coverage data can comprise filtering the one or more calling windows based on a level of guanine-cytosine (GC) content. Filtering the one or more calling windows based on a level of GC content can comprise excluding a calling window of the one or more calling windows if the level of GC content of the calling window is outside a predetermined range. In an aspect, the present methods and systems can filter windows with extreme guanine-cytosine content (GC) content. GC-amplification bias can be corrected when the bias is mostly consistent for any particular level of GC content. At very low or high GC content, however, the stochastic coverage volatility may increase dramatically, making it difficult to normalize effectively. Accordingly, the present methods and systems can filter windows where the GC-fraction is outside of a configurable (e.g., or predefined) range or threshold. As an illustration, the configurable range can comprise [0.3, 0.7], as shown in **FIG. 3**. It should be appreciated, however, that other ranges (e.g., thresholds) can be utilized as appropriate.

[0034] As a further explanation of filtering based on GC content, **FIG. 3** shows a graph illustrating the relationship of GC content and coverage. For example, the coefficient of variation (e.g., standard deviation divided by mean) of coverage is shown on the y-axis and GC content is shown in the x-axis. The graph shows 50 samples (e.g., points jittered for visibility). Above a default upper-limit (e.g., GC = 0.7) of the configurable range, coverage variance can be very high relative to the mean, making coverage-based CNV calls unreliable. Below a default lower-limit (e.g., GC content = 0.3) of the configurable range, additional problems arise. For example, the variance of coverage itself can be highly variable between samples.

This variance makes it difficult to accurately estimate the expected variance of coverage for a particular sample at a particular window, as each reference panel sample's coverage value is an observation from a different distribution.

[0035] In an aspect, the GC content of the full DNA fragment, not only the sequenced read, can influence fragment count. Accordingly, when computing GC-fractions, windows can be symmetrically extended to be at least slightly longer than the average fragment size. The average fragment size can be another configurable parameter of CLAMMS. Average fragment size can default to 200 bp, or other appropriate values may be used.

[0036] Filtering the sample coverage data can comprise filtering the one or more calling windows based on a mappability score of a genomic region of the plurality of genomic regions. For example, the present methods and systems can filter calling windows where the mean mappability score for k -mers starting at each base in the window (default $k=75$) is less than 0.75. Filtering the one or more calling windows based on a mappability score can comprise determining a mappability score for each genomic region of the plurality of genomic regions and excluding a calling window of the one or more calling windows that contains the genomic region of the plurality of genomic regions if the mappability score of the genomic region of the plurality of genomic regions is below a predetermined threshold. Determining a mappability score for each genomic region of the plurality of genomic regions can comprise determining an average of an inverse reference-genome frequency of k -mers whose first base overlaps the genomic region of the plurality of genomic regions.

[0037] In another aspect, filtering the sample coverage data can comprise filtering the one or more calling windows based on a measure of central tendency of read coverage. Filtering the one or more calling windows based on a measure of central tendency of read coverage can comprise excluding a calling window of the one or more calling windows if the calling window of the one or more calling windows comprises a measure of central tendency of read coverage less than an expected coverage value for calling windows with similar GC content. For example, the present methods and systems can filter windows where the median and/or mean coverage across samples is less than 10% of the expectation for windows with similar GC content.

[0038] In another aspect, filtering the sample coverage data can comprise filtering the one or more calling windows based on occurrence of a calling window in a

multi-copy duplication genomic region. Filtering the one or more calling windows based on occurrence of a calling window in a multi-copy duplication genomic region can comprise excluding a calling window of the one or more calling windows if the calling window of the one or more calling windows occurs within a region where multi-copy duplications are known to be present. As an illustration, a portion (e.g., 12% using the defaults above) of exome capture regions can be excluded from the calling process using these filters.

[0039] Returning to **FIG. 2**, at step **204**, sample coverage data can be normalized. The sample coverage data can comprise a plurality of genomic regions. The present methods and systems can normalize the sample coverage data for each individual sample to correct for GC-bias and overall average depth-of-coverage. Normalizing sample coverage data can comprise determining raw coverage for a calling window w , determining a median coverage for the sample coverage data across the one or more calling windows conditional on a GC-fraction of the calling window w , and dividing the raw coverage by the median coverage, resulting in the normalized sample coverage data. Determining a median coverage for the sample coverage data across the plurality of windows conditional on a GC-fraction of the calling window w can comprise binning the one or more calling windows by GC-fraction, resulting in a plurality of bins, determining a median coverage for each bin of the plurality of bins, and/or determining a normalizing factor for each distinct possible GC-fraction using a linear interpolation between the median coverage for two bins nearest to the calling window w .

[0040] Normalization of sample coverage data is described in more detail as follows. For example, a conditional median can be determined (e.g., computed, calculated) by binning all windows for a sample by GC fraction (e.g., [0.300, 0.310], [0.315, 0.325], etc.). For example, a plurality of bins can be determined based on GC fraction values. One or more (or each) of the bins of the plurality of bins can be determined by dividing (e.g., equally) the total GC fraction value range based on one or more incremental values (e.g., 0.01). The median coverage for each bin can be determined (e.g., calculated, computed). The normalizing factor for a given GC fraction can be determined (e.g., calculated, computed). For example, the normalizing factor for a given GC fraction can be determined by using a linear interpolation between the median coverage for the two bins nearest to bin at issue. In an aspect, the binning resolution (e.g., size of incremental values) can be

configurable. An example, default resolution can be determined (e.g., selected) that balances fine-grained binnings with the need to provide each bin with a sufficient sample size for estimation.

[0041] **FIG. 4** is a graph illustrating normalized coverage of various exons. The graph shows mixture models fit to the observed coverage distributions for exons of the gene *GSTT1* (e.g., after within-sample normalization has been applied). Each point (e.g., jittered for visibility) shows a sample's normalized coverage for an exon. The shading of the plot points indicates the most likely copy number given the model and opacity proportional to the likelihood ratio between the most- and second-most-likely copy numbers if the exon were to be treated independently of its neighbors.

[0042] Returning to **FIG. 2**, at step **206**, a mixture model can be fit (e.g., trained, modified, adapted) to the normalized sample coverage data based on the selected reference coverage data. For example, the mixture model can be trained according to the selected reference coverage data. Fitting the mixture model to the normalized sample coverage data based on the selected reference coverage data can comprise determining a plurality of mixture models (e.g., one for each of the plurality of genomic regions). One or more (or each) component of the plurality of mixture models can comprise a corresponding probability distribution. The probability distribution can represent an expected normalized coverage conditional on a particular copy number. The plurality of mixture models can be fit to the normalized sample coverage data using an expectation-maximization algorithm. For example, the plurality of mixture models can be fit to the normalized sample coverage data using an expectation-maximization algorithm to determine a likelihood for each copy number at each of the one or more calling windows. The selected reference coverage data can be input to the expectation-maximization algorithm.

[0043] As a further explanation, the present methods and systems can use mixture models to characterize the expected (e.g., normalized) coverage distribution at each calling window. The expected coverage distribution can be conditional on copy number state. These mixture models can be fit by using a fitting algorithm. For example the mixture models can be fit by identifying the model parameters that best match the shape of the data distribution. In an aspect, the fitting algorithm can comprise an optimization method for estimating the mixture model parameters, such as EM. Alternatively, an unsupervised clustering or sampling algorithm could be

used to identify distinct copy number states and/or model the distribution of coverage data over copy number states.

[0044] For example, the fitting algorithm can comprise an expectation-maximization algorithm (EM algorithm) with input data from a reference panel of samples (e.g., reference coverage data). In an aspect, the EM algorithm can comprise an optimization algorithm for fitting hidden (e.g., latent) model parameters. In some implementations, the fitting algorithm can comprise the use of gradient descent, Newton-Raphson, and/or the like algorithms. The components of the mixture model can correspond to the copy numbers 0, 1, 2, and 3. In some implementations, copy numbers greater than 3 can be ignored. For example, coverage that could be explained by copy number greater than 3 may be the result of stochastic GC-related bias.

[0045] In an aspect, one or more of the components of the mixture model corresponding to non-zero copy numbers can be defined to follow a Gaussian distribution. For example, the Gaussian distribution can be of the form: $\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ where μ indicates a mean and σ indicates a variance or standard deviation. The Gaussian distribution for a diploid copy can comprise at least two free parameters: μ_{DIP} (e.g., the mean of the mixture component corresponding to diploid copy) and σ_{DIP} (e.g., standard deviation of the mixture component corresponding to diploid copy). For each non-diploid copy number k , the mean can be constrained to equal $(k/2) * \mu_{DIP}$ (e.g., hence the term “lattice-aligned” in the CLAMMS acronym). The standard deviation for haploid samples, σ_{HAP} , can be set equal to $\sqrt{0.5} * \sigma_{DIP}$. Despite the Gaussian approximations, coverage conditional on a particular copy number can be Poisson-distributed with variance being equal to the mean. The standard deviation parameters for components corresponding to copy numbers greater than 2 can be set to be equal to σ_{DIP} . This configuration can avoid increasing the rate of false-positive duplications. The constraints imposed on the parameters of the non-diploid components can configure the model to avoid overfitting the training data.

[0046] In an aspect, the fitting algorithm can be configured to account for mismapped reads corresponding to deleted regions. For example, one or more of the components of the mixed model can be defined as an exponential distribution. Homozygous deletions (e.g., copy number 0) can show zero coverage, but

mismapped reads can give a small level of coverage even in truly deleted regions. Accordingly, the component corresponding to copy number 0 can be defined as an exponential distribution. The exponential distribution can comprise rate parameter λ . For example, an exponential distribution can be of the following form: $\lambda e^{-\lambda x}$. The exponential distribution can be configured with a mean (e.g., $1/\lambda$) initially equal to 6.25% of μ_{DIP} or other appropriate ratio. As a further example, the mean of this component can be constrained to be no greater than this initial value. If there are no mismapping issues with the region, iterations of the fitting algorithm can drive the mean to 0 (e.g., $\lambda \rightarrow \infty$). To address this issue, if the mean drops below 0.1% of μ_{DIP} , the fitting algorithm can replace the exponential distribution with a point mass at 0.

[0047] In summary, the mixture model can be configured with one or more of the following parameters: μ_{DIP} and σ_{DIP} ; λ , the rate of the exponential component (e.g., copy number 0), and a flag indicating if the exponential has been replaced by a point mass.

[0048] In an aspect, the fitting algorithm can be configured to iteratively converge on a solution for fitting the mixing model, with each iteration reducing the differences between the model and the data.

[0049] In an aspect, the fitting algorithm can be configured with a maximum number of iterations. For example, the mixture model can be fit using the maximum number of iterations (e.g., 30, 40, 50). In some scenarios the fitting algorithm can use less than the maximum number of iterations. For example, a heuristic can be used to detect early convergence. In the case of the EM algorithm, which is a local optimization procedure, it can be estimated that the initial values of μ_{DIP} and σ_{DIP} may decrease the chance that the fitting algorithm converges to a non-global optimum. In some scenarios, μ_{DIP} can be initialized as the median coverage across all samples for the region in question (e.g., in regions where the median sample is haploid, the iterations may eventually reach the proper diploid mean). In an aspect, σ_{DIP} can be initialized to the median absolute deviation (MAD) of the coverage values around the median of the coverage values, scaled by a constant factor to achieve asymptotic normality (e.g., compare the “mad” function in R).

[0050] Samples that have low likelihoods for all considered copy number states (e.g., less than 2.5σ from the mean) can be flagged as outliers for purposes of model-

fitting. If a region has outlier samples, the mixture model can be retrained with the outlier coverage values removed.

[0051] At step **208**, one or more copy number variants (CNVs) can be identified (e.g., determined, predicted, estimated) according to a Hidden Markov Model (HMM), a Bayesian network, and/or other probabilistic models based on the normalized sample coverage data and the fitted mixture model. For example, identifying one or more copy number variants according to a Hidden Markov Model (HMM) based on the normalized sample coverage data and the fitted mixture model can comprise inputting the normalized sample coverage data for each calling window (e.g., of the one or more calling windows) into the HMM.

[0052] In another aspect, identifying one or more copy number variants according to a Hidden Markov Model (HMM) based on the normalized sample coverage data and the fitted mixture model can comprise determining one or more emission probabilities of the HMM based on the mixture model. For example, a probability of observing a normalized coverage value x , at a calling window w (e.g., of the one or more calling windows), given HMM state s , based on a component of the mixture model for w that corresponds to state s can be determined.

[0053] In another aspect, identifying one or more copy number variants according to a Hidden Markov Model (HMM) based on the normalized sample coverage data and the fitted mixture model can comprise identifying a calling window (e.g., of the one or more calling windows) as a CNV if a maximum likelihood sequence of states of the calling window is non-diploid. For example, a Viterbi algorithm can be performed in a 5' to 3' direction on a genomic region of the plurality of genomic regions. The Viterbi algorithm can be performed in a 3' to 5' direction the genomic region of the plurality of genomic region. A calling window (e.g., of the one or more calling windows) can be identified as a CNV if the genomic region of the plurality of genomic regions associated with the calling window has a most-likely state of non-diploid in the 5' to 3' direction and the 3' to 5' direction.

[0054] In an aspect the HMM can comprise a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (e.g., hidden) states. The hidden state space can comprise one of N possible values, modeled as a categorical distribution. The HMM can comprise transition probabilities. For each of N possible states that a hidden variable at time t can be in, there can be a transition probability from this state to each of the possible states of

the hidden variable at time $t + 1$, for a total of transition N^2 probabilities. The HMM can also comprise emission probabilities (e.g., for each of the N possible states) that govern the distribution of the observed variable at a particular time given the state of the hidden variable at that time.

[0055] The input to the HMM can be the normalized coverage values (e.g., from the within-sample procedure described previously) for an individual sample at each calling window. For example, the states of the HMM can comprise DEL (deletion), DIP (diploid), DUP (duplication), and/or the like. In some scenarios, the distinction between copy numbers 0 and 1 can be made in a post-processing step after DEL calls have been made.

[0056] In an aspect, the HMM can comprise that transition probabilities as input values. The transition probabilities can be based on those used in XHMM. For example, the transition probabilities of XHMM, except the parameter $1/q$ (e.g., the mean of the prior geometric distribution of the number windows in a CNV), can be set to 0 (e.g., $q = \infty$). For example, the transition probability can be similar (e.g., roughly the same as XHMM (non-zero)) to parameters of XHMM with the exception of the XHMM parameter $1/q$, which can be set to 0 by setting q equal to infinity. The effect of this setting is that the HMM can be configured to place no prior assumptions on the number of windows in a CNV. Instead the HMM can be configured to only use the exponentially-distributed attenuation factor which is based on actual genomic distance. In an aspect, setting the XHMM parameter $1/q$ to zero can result in the following two assumptions: 1) DEL and DUP are equally likely, and 2) the size of CNVs is exponentially distributed. The teachings related to the XHMM as taught by *Fromer et al.* (2012), "Discovery and statistical genotyping of copy-number variation from whole-exome sequencing depth." *Am J Hum Genet*, 91 (4), 597-607, are specifically incorporated herein by reference.

[0057] In an aspect, the HMM can comprise emission probabilities as input values. The emission probabilities can be derived from the mixture models. For example, the probability of observing a (e.g., normalized) coverage value x , at a calling window w , given HMM state s , can be given by the component of the mixture model trained at w , that corresponds to state s . For the DEL state, a likelihood-weighted average of the probabilities given copy number 0 and 1 can be used. For example, if $L(CN=1 | cov) = 9 * L(CN=0 | cov)$, then the emission probability can be $0.9 * P(cov | CN=1) + 0.1 * P(cov | CN=0)$.

[0058] Using this Hidden Markov Model, the present methods and systems can be configured to identify CNVs. For example, the present methods and systems can be configured to identify CNV as regions where the maximum-likelihood sequence of states (e.g., predicted by the Viterbi algorithm or other appropriate algorithm) is non-diploid. It should be noted that running the Viterbi algorithm in only one direction may introduce a directional bias to the CNV calls. There is effectively a high cost to “open” a CNV but a low cost of “extending” the CNV. Thus, the called CNV regions may tend to overshoot the trailing breakpoint. To solve this problem, the present methods and systems can be configured to only report as CNVs regions for which the most-likely state is non-diploid in both a run of the Viterbi algorithm in the 5' to 3' direction and a run in 3' to 5' direction.

[0059] In an aspect, for each discovered CNV, five quality metrics can be computed based on probabilities from the Forward-Backward algorithm: Q_{any} , the phred-scaled probability that the region contains any CNV; $Q_{extend\ left}$ and $Q_{extend\ right}$, phred-scaled probabilities that the true CNV extends at least one window further upstream/downstream from the called region; and $Q_{contract\ left}$ and $Q_{contract\ right}$, phred-scaled probabilities that the true CNV is contracted compared to the called region by at least one window upstream or downstream.

[0060] It should be noted that even with the *a priori* filtering of windows with GC-content outside of the threshold range (e.g., [0.3, 0.7]) as described above, high rates of stochastic sequencing artifacts may still occur at the extreme ends of this threshold range. The Viterbi and Forward-Backward algorithms can be modified (e.g., configured) to place less credence on windows with “moderately-extreme” GC-content without ignoring these windows entirely. This configuration can be accomplished by multiplying the log-emission-probability for all states at a given window by a weight in the range [0, 1] based on the GC-content of the window. This configuration can reduce the relative significance of the data (e.g., observed coverage) at this window compared to the prior window (e.g., encoded by the state transition probabilities). As an illustration, for GC-fraction f in the default *a priori* valid range of [0.3, 0.7], the window weight can be set equal to $(1 - (5 * abs(f - 0.5))^{18})^{18}$. The high polynomial term can make the curve flat for non-extreme GC (e.g., weight = 0.99993 for $f = 0.4$), but drop sharply at the edges of the valid GC range (e.g., weight = 0.5 for $f = 0.3333$).

[0061] In an aspect, the present methods and systems can fit models and make CNV calls for regions on the sex chromosomes if the sex of each input sample is provided. Basing the expected copy number (e.g., diploid or haploid) on sex explicitly can be more effective than normalizing the variance due to sex or comparing samples to highly-correlated samples because such approach accounts for the integer nature of copy number states. As an illustration, a female with 0.5x the expected coverage for a region on chrX is likely to have a heterozygous deletion. A male with the same level of coverage may not be likely to have a heterozygous deletion, because one cannot have a copy number of 1/2.

[0062] At step **210**, the one or more copy number variants can be output. For example, the one or more copy number variants can be output to a user (e.g., via a user interface). The one or more copy number variants can be transmitted via a network to remote location. The one or more copy number variants can be provided as input to another executable program. The one or more copy number variants can be stored in a storage location, such as a database, or other file format. Example output is shown in **FIG. 15**.

[0063] **FIG. 5** is a flow chart illustrating another example method **500** for estimating copy number variants. At step **502**, sample coverage data comprising a plurality of genomic regions can be provided (e.g., by a user, from a first device to a second device). In an aspect, the plurality of genomic regions can be divided into one or more calling windows (e.g., a plurality of calling windows). For example, only genomic regions of the plurality of genomic regions larger than a predetermined size may be divided.

[0064] At step **504**, an indication of reference coverage data can be received (e.g., by a user, from a first device to a second device). The reference coverage data can be selected based on a sample grouping technique. For example, the sample grouping technique can comprise a technique (e.g., algorithm) for grouping samples by similarity. The sample grouping technique can comprise a clustering algorithm, a classification algorithm, a combination thereof, and/or the like. For example, the sample grouping technique can comprise receiving a plurality of metrics for the sample coverage data, defining a distance metric between the sample coverage data and the reference coverage data based on the plurality of metrics, selecting the reference coverage data for each sample based on the distance metric, and/or the like.

[0065] As an illustration, the sample grouping technique can comprise a k-Nearest Neighbors (knn) algorithm. Selecting reference coverage data based on the sample grouping technique can comprise one or more of: scaling a plurality of metrics associated with the reference coverage data, generating a *k*-d tree based on the scaled plurality of metrics associated with the reference coverage data, scaling a plurality of metrics associated with the sample coverage data, adding the sample coverage data to the *k*-d tree based on the scaled plurality of metrics associated with the sample coverage data, identifying a predetermined number of nearest neighbors to the sample coverage data as the selected reference coverage data, and/or the like.

[0066] At step 506, one or more filters can be selected (e.g., by a user, by the first device and/or the second device) to apply to the sample coverage data to normalize the sample coverage data. For example, the sample coverage data can be filtered. The one or more filters can be configured for one or more of: filtering the one or more calling windows based on a level of GC content, filtering the one or more calling windows based on a mappability score of a genomic region of the plurality of genomic regions, filtering the one or more calling windows based on a measure of central tendency of read coverage, filtering the one or more calling windows based on occurrence of a calling window in a multi-copy duplication genomic region, and/or the like.

[0067] In an aspect, filtering the one or more calling windows based on a level of GC content can comprise excluding a calling window of the one or more calling windows if the level of GC content of the calling window is outside a predetermined range.

[0068] In an aspect, filtering the one or more calling windows based on a mappability score can comprise determining a mappability score for each genomic region of the plurality of genomic regions. For example, determining a mappability score for each genomic region of the plurality of genomic regions can comprise determining an average of an inverse reference-genome frequency of *k*-mers whose first base overlaps the genomic region of the plurality of genomic regions. Filtering the one or more calling windows based on a mappability score can further comprise excluding a calling window of the one or more calling windows that contains the genomic region of the plurality of genomic regions if the mappability score of the genomic region of the plurality of genomic regions is below a predetermined threshold.

[0069] In an aspect, filtering the one or more calling windows based on a measure of central tendency of read coverage can comprise excluding a calling window of the one or more calling windows if the calling window of the one or more calling windows comprises a measure of central tendency of read coverage less than an expected coverage value for calling windows with similar GC content.

[0070] In an aspect, filtering the one or more calling windows based on occurrence of a calling window in a multi-copy duplication genomic region can comprise excluding a calling window of the one or more calling windows if the calling window of the one or more calling windows occurs within a region where multi-copy duplications are known to be present.

[0071] In an aspect, filtering and/or normalizing can comprise one or more of determining raw coverage for a calling window w , determining a median coverage for the sample coverage data across the one or more calling windows conditional on a GC-fraction of the calling window w , dividing the raw coverage by the median coverage (e.g., resulting in the normalized sample coverage data), and/or the like. For example, determining a median coverage for the sample coverage data across the plurality of windows conditional on a GC-fraction of the calling window w can comprise one or more of: binning the one or more calling windows by GC-fraction (e.g., resulting in a plurality of bins), determining a median coverage for each bin of the plurality of bins, determining a normalizing factor for each distinct possible GC-fraction using a linear interpolation between the median coverage for two bins nearest to the calling window w , and/or the like.

[0072] At step **508**, fitting of a mixture model to the normalized sample coverage data based on the reference coverage data can be requested (e.g., by a user, from the first device to the second device). For example training of the mixture model according to the selected reference coverage data can be requested. Fitting the mixture model to the normalized sample coverage data based on the reference coverage data can comprise determining a plurality of mixture models, one for each of the plurality of genomic regions. Each component of the plurality of mixture models can comprise a probability distribution that represents an expected normalized coverage conditional on a particular copy number. Fitting the mixture model to the normalized sample coverage data based on the reference coverage data can comprise fitting the plurality of mixture models to the normalized sample coverage data using an expectation-maximization algorithm to determine a

likelihood for each copy number at each of the one or more calling windows. The selected reference coverage data can be input to the expectation-maximization algorithm.

[0073] At step **510**, one or more copy number variants can be identified (e.g., by the user, by the first device, by the second device) according to a Hidden Markov Model (HMM) based on the normalized sample coverage data and the fitted mixture model. For example, identifying one or more copy number variants according to a Hidden Markov Model (HMM) based on the normalized sample coverage data and the fitted mixture model can comprise one or more of inputting the normalized sample coverage data for each calling window (e.g., of the one or more calling windows) into the HMM, determining one or more emission probabilities of the HMM based on the mixture model, identifying a calling window (e.g., of the one or more calling windows) as a CNV if a maximum likelihood sequence of states of the calling window is non-diploid, and/or the like.

[0074] In an aspect, determining one or more emission probabilities of the HMM based on the mixture model can comprise determining a probability of observing a normalized coverage value x , at a calling window w (e.g., of the one or more calling windows), given HMM state s , based on a component of the mixture model for w that corresponds to state s .

[0075] In an aspect, identifying a calling window (e.g., of the one or more calling windows) as a CNV if a maximum likelihood sequence of states of the calling window is non-diploid can comprise one or more of performing a Viterbi algorithm in a 5' to 3' direction on a genomic region of the plurality of genomic regions, performing the Viterbi algorithm in a 3' to 5' direction the genomic region of the plurality of genomic regions, identifying a calling window (e.g., of the one or more calling windows) as a CNV if the genomic region of the plurality of genomic regions associated with the calling window has a most-likely state of non-diploid in the 5' to 3' direction and the 3' to 5' direction, and/or the like.

[0076] At step **512**, an indication of the one or more copy number variants can be received (e.g., by a user, by the first device, by the second device). For example, the indication can be provided to a display, via a network, and/or the like. An example indication of the one or more copy number variants is shown in **FIG. 15**.

[0077] **FIG. 6** is a flow chart illustrating yet another example method **600** for estimating copy number variants. At step **602**, sample coverage data comprising a

plurality of genomic regions can be received. In an aspect, the plurality of genomic regions can be divided into one or more calling windows (e.g., a plurality of calling windows). For example, only genomic regions of the plurality of genomic regions larger than a predetermined size may be divided.

[0078] In an aspect, the sample coverage data can be filtered. For example, filtering the sample coverage data can comprise one or more of filtering the one or more calling windows based on a level of GC content, filtering the one or more calling windows based on a mappability score of a genomic region of the plurality of genomic regions, filtering the one or more calling windows based on a measure of central tendency of read coverage, filtering the one or more calling windows based on occurrence of a calling window in a multi-copy duplication genomic region, and/or the like.

[0079] In an aspect, filtering the one or more calling windows based on a level of GC content can comprise excluding a calling window of the one or more calling windows if the level of GC content of the calling window is outside a predetermined range. Filtering the one or more calling windows based on a mappability score can comprise determining a mappability score for each genomic region of the plurality of genomic regions. For example, determining a mappability score for each genomic region of the plurality of genomic regions can comprise determining an average of an inverse reference-genome frequency of k -mers whose first base overlaps the genomic region of the plurality of genomic regions. Filtering the one or more calling windows based on a mappability score can further comprise excluding a calling window of the one or more calling windows that contains the genomic region of the plurality of genomic regions if the mappability score of the genomic region of the plurality of genomic regions is below a predetermined threshold.

[0080] In an aspect, filtering the one or more calling windows based on a measure of central tendency of read coverage can comprise excluding a calling window of the one or more calling windows if the calling window of the one or more calling windows comprises a measure of central tendency of read coverage less than an expected coverage value for calling windows with similar GC content. Filtering the one or more calling windows based on occurrence of a calling window in a multi-copy duplication genomic region can comprise excluding a calling window of the one or more calling windows if the calling window of the one or more calling windows occurs within a region where multi-copy duplications are known to be

present.

[0081] At step **604**, a first plurality of metrics for the sample coverage data can be retrieved. The first plurality of metrics can comprise, for example, sequencing quality control (QC) metrics, sample metadata, ancestry-based values, sequence-similarity scores, and/or any metric that captures sample-level variability. For example, in the case of sequencing QC metrics, seven QC metrics can be used. As an illustration, the sequencing QC metrics can comprise GCDROPOUT, ATDROPOUT, MEANINSERTSIZE, ONBAITVSSELECTED, PCTPFUQREADS, PCTTARGETBASES10X, PCTTARGETBASES50X, and/or the like. The sequencing QC metrics can be scaled (e.g., by applying a linear transform) and processed to build a k -d tree.

[0082] At step **606**, a sample grouping technique can be applied to the sample coverage data and reference coverage data to select a subset of the reference coverage data. The sample grouping technique can comprise a technique (e.g., algorithm) for grouping samples by similarity. For example, the sample grouping technique can comprise a clustering algorithm, a classification algorithm, a combination thereof, and/or the like. In an aspect, applying a sample grouping technique to the sample coverage data and reference coverage data to select a subset of the reference coverage data can comprise defining a distance metric between the sample coverage data and the reference coverage data based on the first plurality of metrics. The reference coverage data can be selected for each sample based on the distance metric.

[0083] As another example, the sample grouping technique can comprise a k -Nearest Neighbors (knn) algorithm. Applying the sample grouping technique to the sample coverage data and reference coverage data to select a subset of the reference coverage data can comprise one or more of: retrieving a second plurality of metrics associated with the reference coverage data, scaling the second plurality of metrics associated with the reference coverage data, generating a k -d tree based on the scaled second plurality of metrics associated with the reference coverage data, scaling the first plurality of metrics for the sample coverage data, adding the sample coverage data to the k -d tree based on the scaled first plurality of metrics for the sample coverage data, identifying a predetermined number of nearest neighbors to the sample coverage data as the subset of the reference coverage data, and/or the like.

[0084] At step **608**, the sample coverage data comprising the plurality of genomic regions can be normalized. For example, normalizing the sample coverage data comprising the plurality of genomic regions can comprise one or more of determining raw coverage for a calling window w , determining a median coverage for the sample coverage data across the one or more calling windows conditional on a GC-fraction of the calling window w ; dividing the raw coverage by the median coverage (e.g., resulting in the normalized sample coverage data), and/or the like.

[0085] In an aspect, determining a median coverage for the sample coverage data across the plurality of windows conditional on a GC-fraction of the calling window w can comprise one or more of binning the one or more calling windows by GC-fraction (e.g., resulting in a plurality of bins), determining a median coverage for each bin of the plurality of bins, determining a normalizing factor for each distinct possible GC-fraction using a linear interpolation between the median coverage for two bins nearest to the calling window w , and/or the like.

[0086] At step **610**, a mixture model can be fit to the normalized sample coverage data based on the subset of the reference coverage data. For example, the mixture model can be trained according to the subset of the reference coverage data. Fitting the mixture model to the normalized sample coverage data based on the subset of the reference coverage data can comprise determining a plurality of mixture models, one for each of the plurality of genomic regions. One or more (or each) component of the plurality of mixture models can comprise a probability distribution that represents an expected normalized coverage conditional on a particular copy number. Fitting the mixture model to the normalized sample coverage data based on the subset of the reference coverage data can also comprise fitting the plurality of mixture models to the normalized sample coverage data using an expectation-maximization algorithm to determine a likelihood for each copy number at each of the one or more calling windows. The subset of the reference coverage data can be input to the expectation-maximization algorithm.

[0087] At step **612**, one or more copy number variants can be identified according to a Hidden Markov Model (HMM) based on the normalized sample coverage data and the fitted mixture model. For example, identifying one or more copy number variants according to a Hidden Markov Model (HMM) based on the normalized sample coverage data and the fitted mixture model can comprise one or more of inputting the normalized sample coverage data for each calling window (e.g., of the

one or more calling windows) into the HMM, determining one or more emission probabilities of the HMM based on the mixture model, identifying a calling window (e.g., of the one or more calling windows) as a CNV if a maximum likelihood sequence of states of the calling window is non-diploid, and/or the like. In an aspect, determining one or more emission probabilities of the HMM based on the mixture model can comprise determining a probability of observing a normalized coverage value x , at a calling window w (e.g., of the one or more calling windows), given HMM state s , based on a component of the mixture model for w that corresponds to state s .

[0088] In an aspect, identifying a calling window (e.g., of the one or more calling windows) as a CNV if a maximum likelihood sequence of states of the calling window is non-diploid can comprise one or more of: performing a Viterbi algorithm in a 5' to 3' direction on a genomic region of the plurality of genomic regions, performing the Viterbi algorithm in a 3' to 5' direction the genomic region of the plurality of genomic regions, identifying a calling window (e.g., of the one or more calling windows) as a CNV if the genomic region of the plurality of genomic regions associated with the calling window has a most-likely state of non-diploid in the 5' to 3' direction and the 3' to 5' direction, and/or the like.

[0089] At step **614**, the one or more copy number variants can be output. For example, the one or more copy number variants can be output to a user (e.g., via a user interface). The one or more copy number variants can be transmitted via a network to remote location. The one or more copy number variants can be provided as input to another executable program. The one or more copy number variants can be stored in a storage location, such as a database, or other file format. Example output is shown in **FIG. 15**.

[0090] In an exemplary aspect, the methods and systems can be implemented on a computer **701** as illustrated in **FIG. 7** and described below. Similarly, the methods and systems disclosed can utilize one or more computers to perform one or more functions in one or more locations. **FIG. 7** is a block diagram illustrating an exemplary operating environment for performing the disclosed methods. This exemplary operating environment is only an example of an operating environment and is not intended to suggest any limitation as to the scope of use or functionality of operating environment architecture. Neither should the operating environment be interpreted as having any dependency or requirement relating to any one or

combination of components illustrated in the exemplary operating environment.

[0091] The present methods and systems can be operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well-known computing systems, environments, and/or configurations that can be suitable for use with the systems and methods comprise, but are not limited to, personal computers, server computers, laptop devices, and multiprocessor systems. Additional examples comprise set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that comprise any of the above systems or devices, and the like.

[0092] The processing of the disclosed methods and systems can be performed by software components. The disclosed systems and methods can be described in the general context of computer-executable instructions, such as program modules, being executed by one or more computers or other devices. Generally, program modules comprise computer code, routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The disclosed methods can also be practiced in grid-based and distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules can be located in both local and remote computer storage media including memory storage devices.

[0093] Further, one skilled in the art will appreciate that the systems and methods disclosed herein can be implemented via a general-purpose computing device in the form of a computer **701**. The components of the computer **701** can comprise, but are not limited to, one or more processors **703**, a system memory **712**, and a system bus **713** that couples various system components including the one or more processors **703** to the system memory **712**. The system can utilize parallel computing.

[0094] The system bus **713** represents one or more of several possible types of bus structures, including a memory bus or memory controller, a peripheral bus, an accelerated graphics port, or local bus using any of a variety of bus architectures. By way of example, such architectures can comprise an Industry Standard Architecture (ISA) bus, a Micro Channel Architecture (MCA) bus, an Enhanced ISA (EISA) bus, a Video Electronics Standards Association (VESA) local bus, an Accelerated Graphics Port (AGP) bus, and a Peripheral Component Interconnects

(PCI), a PCI-Express bus, a Personal Computer Memory Card Industry Association (PCMCIA), Universal Serial Bus (USB) and the like. The bus **713**, and all buses specified in this description can also be implemented over a wired or wireless network connection and each of the subsystems, including the one or more processors **703**, a mass storage device **704**, an operating system **705**, CNV calling software **706**, CNV calling data **707**, a network adapter **708**, the system memory **712**, an Input/Output Interface **710**, a display adapter **709**, a display device **711**, and a human machine interface **702**, can be contained within one or more remote computing devices **714a,b,c** at physically separate locations, connected through buses of this form, in effect implementing a fully distributed system.

[0095] The computer **701** typically comprises a variety of computer readable media. Exemplary readable media can be any available media that is accessible by the computer **701** and comprises, for example and not meant to be limiting, both volatile and non-volatile media, removable and non-removable media. The system memory **712** comprises computer readable media in the form of volatile memory, such as random access memory (RAM), and/or non-volatile memory, such as read only memory (ROM). The system memory **712** typically contains data such as the CNV calling data **707** and/or program modules such as the operating system **705** and the CNV calling software **706** that are immediately accessible to and/or are presently operated on by the one or more processors **703**.

[0096] In another aspect, the computer **701** can also comprise other removable/non-removable, volatile/non-volatile computer storage media. By way of example, **FIG. 7** illustrates the mass storage device **704** which can provide non-volatile storage of computer code, computer readable instructions, data structures, program modules, and other data for the computer **701**. For example and not meant to be limiting, the mass storage device **704** can be a hard disk, a removable magnetic disk, a removable optical disk, magnetic cassettes or other magnetic storage devices, flash memory cards, CD-ROM, digital versatile disks (DVD) or other optical storage, random access memories (RAM), read only memories (ROM), electrically erasable programmable read-only memory (EEPROM), and the like.

[0097] Optionally, any number of program modules can be stored on the mass storage device **704**, including by way of example, the operating system **705** and the CNV calling software **706**. Each of the operating system **705** and the CNV calling software **706** (or some combination thereof) can comprise elements of the

programming and the CNV calling software **706**. The CNV calling data **707** can also be stored on the mass storage device **704**. The CNV calling data **707** can be stored in any of one or more databases known in the art. Examples of such databases comprise, DB2®, Microsoft® Access, Microsoft® SQL Server, Oracle®, mySQL, PostgreSQL, and the like. The databases can be centralized or distributed across multiple systems.

[0098] In another aspect, the user can enter commands and information into the computer **701** via an input device (not shown). Examples of such input devices comprise, but are not limited to, a keyboard, pointing device (*e.g.*, a “mouse”), a microphone, a joystick, a scanner, tactile input devices such as gloves, and other body coverings, and the like. These and other input devices can be connected to the one or more processors **703** via the human machine interface **702** that is coupled to the system bus **713**, but can be connected by other interface and bus structures, such as a parallel port, game port, an IEEE 1394 Port (also known as a Firewire port), a serial port, or a universal serial bus (USB).

[0099] In yet another aspect, the display device **711** can also be connected to the system bus **713** via an interface, such as the display adapter **709**. It is contemplated that the computer **701** can have more than one display adapter **709** and the computer **701** can have more than one display device **711**. For example, a display device can be a monitor, an LCD (Liquid Crystal Display), or a projector. In addition to the display device **711**, other output peripheral devices can comprise components such as speakers (not shown) and a printer (not shown) which can be connected to the computer **701** via the Input/Output Interface **710**. Any step and/or result of the methods can be output in any form to an output device. Such output can be any form of visual representation, including, but not limited to, textual, graphical, animation, audio, tactile, and the like. The display **711** and computer **701** can be part of one device, or separate devices.

[00100] The computer **701** can operate in a networked environment using logical connections to one or more remote computing devices **714a,b,c**. By way of example, a remote computing device can be a personal computer, portable computer, smartphone, a server, a router, a network computer, a peer device or other common network node, and so on. Logical connections between the computer **701** and a remote computing device **714a,b,c** can be made via a network **715**, such as a local area network (LAN) and/or a general wide area network (WAN). Such network

connections can be through the network adapter **708**. The network adapter **708** can be implemented in both wired and wireless environments. Such networking environments are conventional and commonplace in dwellings, offices, enterprise-wide computer networks, intranets, and the Internet.

[00101] For purposes of illustration, application programs and other executable program components such as the operating system **705** are illustrated herein as discrete blocks, although it is recognized that such programs and components reside at various times in different storage components of the computing device **701**, and are executed by the one or more processors **703** of the computer. An implementation of the CNV calling software **706** can be stored on or transmitted across some form of computer readable media. Any of the disclosed methods can be performed by computer readable instructions embodied on computer readable media. Computer readable media can be any available media that can be accessed by a computer. By way of example and not meant to be limiting, computer readable media can comprise “computer storage media” and “communications media.” “Computer storage media” comprise volatile and non-volatile, removable and non-removable media implemented in any methods or technology for storage of information such as computer readable instructions, data structures, program modules, or other data. Exemplary computer storage media comprises, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by a computer.

[00102] The methods and systems can employ Artificial Intelligence techniques such as machine learning and iterative learning. Examples of such techniques include, but are not limited to, expert systems, case based reasoning, Bayesian networks, behavior based AI, neural networks, fuzzy systems, evolutionary computation (e.g. genetic algorithms), swarm intelligence (e.g. ant algorithms), and hybrid intelligent systems (e.g. Expert inference rules generated through a neural network or production rules from statistical learning).

[00103] The following examples are put forth so as to provide those of ordinary skill in the art with a complete disclosure and description of how the compounds, compositions, articles, devices and/or methods claimed herein are made and

evaluated, and are intended to be purely exemplary and are not intended to limit the scope of the methods and systems. Efforts have been made to ensure accuracy with respect to numbers (e.g., amounts, etc.), but some errors and deviations should be accounted for.

[00104] The present methods and systems were validated using a variety of validation experiments. A first experiment evaluated the adherence of CNV calls from CLAMMS and other algorithms to Mendelian inheritance patterns on a pedigree. Calls from CLAMMS, XHMM (another widely-used algorithm), and SNP genotyping arrays for a set of 3164 samples are compared. Another validation experiment used TaqMan qPCR to validate CNVs predicted by CLAMMS. For example, TaqMan qPCR can be used as an illustration to validate CLAMMS at 37 loci (95% of rare variants validate), across 17 common variant loci. The mean precision and recall are 99% and 94%, respectively.

[00105] Validation of the present method and systems included analysis of the complexity of operation and scalability of the CLAMMS algorithm. For example, sequencing n samples may take $O(n \log n)$ time, as maintaining the k -d tree takes only $O(\log n)$ time per sample. This approach improves on the $O(n^2)$ complexity of previous algorithms (e.g., both PCA and the reference-panel selection methods of CANOES and ExomeDepth require the coverage profile of each sample to be compared to every other sample).

[00106] As described further herein the adherence of CNV calls from CLAMMS, XHMM, CoNIFER, CANOES, and ExomeDepth to Mendelian inheritance patterns can be evaluated. As an illustration, the adherence of CNV calls from these algorithms were evaluated for eight members of the CEPH pedigree 1463, sequenced in three technical replicates. 92 additional samples were provided as a reference panel. It should be noted that most CNVs in the pedigree are common variants (e.g., by definition). 98% of calls were inherited and 94% were consistent across all three technical replicates. Statistics for other evaluated algorithms are presented further below.

[00107] The CLAMMS algorithm's improved performance for common CNVs does not come at a cost of reduced performance for rare CNVs. For example, as another validation experiment CNV calls from CLAMMS and XHMM were compared against "gold-standard" calls from PennCNV (e.g., which uses data from SNP genotyping arrays) for 3164 samples. The PennCNV calls were subject to several

quality-control filters. For rare variants (e.g., $AF \leq 0.1\%$ in the array data), CLAMMS had 78% precision and 65% recall, compared to XHMM's 66% precision and 64% recall.

[00108] As another validation experiment, TaqMan qPCR can be used to validate a random subset of CNVs predicted by CLAMMS. TaqMan qPCR was used to validate at 20 rare variant loci and 20 common variant loci that overlap disease-associated genes in the Human Gene Mutation Database. In this example validation experiment, 19/20 (95%) of the rare variants predicted by CLAMMS were validated. Three common variant loci were excluded due to high variance in the TaqMan data. The remaining 17 loci yielded mean precision/recall values of 99.0% and 94.1% respectively. As another result, 16/17 (94%) loci had no false positives. As a further result, 13/17 loci (76%) had greater than or equal to 90% sensitivity for 165 samples genotyped. **FIG. 8** through **FIG. 14** illustrate these validation experiments in greater detail.

[00109] **FIG. 8** compares the RAM usage of CLAMMS vs. other algorithms. The RAM usage of CLAMMS appears constant while the RAM usage of other algorithms increasing linearly with the number of samples. RAM usage of CNV-calling algorithms is shown for 50 samples for all algorithms. RAM usage of CNV-calling algorithms is shown for 100 and 200 samples with all algorithms but CANOES, which ran 4 hours without finishing. RAM usage is shown for 3164 samples using CLAMMS and XHMM.

[00110] In an aspect, the CLAMMS algorithm can be validated as follows. Validation can be performed, for example, using data from a repository, such as CEPH pedigree 1463. A first validation experiment was to evaluate the adherence of CNV calls from CLAMMS and four other algorithms (XHMM, CoNIFER, CANOES, and ExomeDepth) to Mendelian inheritance patterns on an 8-member pedigree (e.g., a subset of CEPH pedigree 1463, including grandparents NA12889, NA12890, NA12891, NA12892; parents NA12877, NA12878; and children NA12880, NA12882). Each of the 8 pedigree members were sequenced in three technical replicates. CNV calls were made using each algorithm's default parameters as described herein. A reference panel of 92 unrelated samples were made available to each algorithm. To ensure a fair comparison, the *a-priori* filters used by CLAMMS (e.g., filtering extreme-GC and low-mappability regions) can be applied to the input data for all algorithms, so differences in performance may not be attributed to

CLAMMS' exclusion of the most problematic genomic regions. Sex chromosomes were also excluded from the comparison.

[00111] Three evaluation metrics can be computed for each algorithm: 1) the proportion of calls that were consistent across all 3 technical replicates, 2) the transmission rate of calls in the 1st and 2nd generations, and 3) the proportion of calls in the 2nd and 3rd generations that were inherited. A 50% overlap criterion was used when determining whether a call is transmitted and/or inherited (e.g., a CNV in a child is inherited if any CNV in its parents overlaps at least 50% of it).

[00112] **FIG. 9** is a table illustrating performance metrics for CNV calls on the CEPH pedigree. The # Calls column is for the 8 pedigree members across 3 technical replicates (e.g., 24 samples in total). CNVs were classified as common if the CNV's allele frequency was greater than or equal to 1%, and classified as rare otherwise (e.g., note that rare CNVs may be false-positives). ExomeDepth calls can be excluded with Bayes Factor of less than 10 (e.g., or other threshold). **FIG. 9** also shows the number of calls made by each algorithm, consistency across technical replicates, and corresponding Mendelian inheritance patterns. As previously explained, all of the algorithms mentioned except CLAMMS are focused exclusively on rare variants, assuming that reference panel samples are diploid (e.g., presenting a unimodal coverage distribution) at all loci. The poor performance of the other algorithms is therefore to be expected as most CNVs in the pedigree are common variants. CLAMMS on the other hand genotypes these common variants accurately (e.g., with only 2% of its calls being putatively *de novo*). The higher-than-Mendelian transmission rate (e.g., 61%) can simply be due to chance (e.g., there are only 27 unique CNV loci in the 1st and 2nd generations).

[00113] In an aspect, validation can be performed using "gold-standard" array-based CNV calls. Our second validation experiment was to compare CNV calls from CLAMMS andXHMM against "gold-standard" calls from PennCNV, that uses data from SNP genotyping arrays for a set of 3164 samples in the Regeneron Genetics Center's human exome variant database. Test set samples were excluded if any of the following test conditions were met: # PennCNV calls greater than 50, LRR_SD (standard deviation of log R ratio) greater than 0.23 (95th percentile), and BAF_drift (B-allele frequency drift) greater than 0.005 (95th percentile).

[00114] In an aspect, array-based CNV calls, despite generally being more accurate than CNV calls from exome sequencing read depth, may not be a true "gold-

standard” and can include false positives, including several putative copy number polymorphic loci (e.g., AF greater than 1%) that did not overlap any variants in two published datasets (CNV calls from 849 whole genomes, and array-based CNV calls from 19,584 controls in an autism study). To minimize the false positive rate in the test set, only CNVs were included that were rare and not small. PennCNV calls were excluded for which one or more of the following conditions are met: CNV length less than 10 kb or greater than 2 Mb, CNV does not overlap at least 1 exon and at least 10 SNPs in the array design, the CNV overlaps a gap in the reference genome (e.g., GRCh37) or a common genomic rearrangement in HapMap, allele frequency greater than 0.1% specific data sets and/or or the 3,164 test samples (e.g., CNVs are included in the allele frequency count if they overlap at least 33.3% of the CNV in question).

[00115] The final test set after all filters have been applied can comprise 1,715 CNVs (e.g., 46% DEL, 54% DUP) in 1,240 samples. For this evaluation, both CLAMMS and XHMM were run with default parameters and procedures. It is recommended to consider samples with 2x the median number of calls for any particular dataset to be outliers. For this example dataset, the median number of CLAMMS calls/sample is 11. CLAMMS calls from 26 samples (e.g., 0.8% of the total) were excluded where CLAMMS performs greater than 22 calls. Array calls from these samples can still be included in the test set.

[00116] **FIG. 10** shows CLAMMS & XHMM CNV calls compared to PennCNV “gold-standard.” Precision can be calculated as the percentage of CLAMMS/XHMM calls that could possibly be supported by a PennCNV call--meaning that the two algorithms are subject to the same filtering criteria--that are in fact overlapped by a PennCNV call at the specified overlap threshold. The recall (e.g., sensitivity) can be calculated as the percentage of PennCNV calls that are overlapped by any CLAMMS/XHMM call (e.g., no filters applied) at the specified overlap threshold. F-score can be defined as the geometric mean of precision and recall.

[00117] In an aspect, CLAMMS can achieve an 9.3% higher F-score than XHMM using the any-overlap criterion, 5.8% higher using the 33%-overlap criterion, and 4.9% higher using the 50%-overlap criterion. This improvement is driven by CLAMMS' greater precision (e.g., 18%-20% greater depending on the overlap threshold).

[00118] CLAMMS is generally more conservative when estimating a CNV's breakpoints (e.g., reporting smaller CNVs) than PennCNV orXHMM, which is why recall is significantly greater using any overlap vs. 50% overlap. As discussed herein, algorithms including PennCNV and XHMM use the Viterbi algorithm to identify CNV regions, scanning across the exome in one direction (e.g., 5' to 3'). Such approach introduces directional bias into the CNV calls: there is effectively a high cost of “opening” a CNV but a low cost of “extending” the CNV, so the called CNV regions will tend to overshoot the 3'-end breakpoint. CLAMMS on the other hand can be configured to only report the intersection of the CNV regions called when Viterbi is run forwards (5' to 3') and backwards (3' to 5'), eliminating the directional bias.

[00119] In an aspect, validation can be performed using TaqMan qPCR as follows. TaqMan quantitative-PCR can be used to validate a selection of CNV loci (e.g., 20 rare, 20 common) predicted by CLAMMS. For each locus, the PCR-based copy number predictions can be compared to CLAMMS CNV genotypes for 56 / 165 samples for rare and common loci respectively. The CNV loci can be selected randomly from the set of all loci that overlapped at least one gene with a disease association recorded in the Human Gene Mutation Database.

[00120] Using this approach, 19/20 (95%) of the rare variants were validated. 3/20 common variant loci were plausibly correct, but had high variance in the PCR data, making the results ambiguous. 16/17 (94%) of the remaining common variant loci had no false positives and one locus had 5/6 calls correct. 13/17 (76%) non-ambiguous common variant loci had greater than or equal to 90% sensitivity (e.g., including 9/17 loci with 100% sensitivity). The other 4/17 had sensitivities of 87.5%, 87.3%, 81.5%, and 70.1%. The means of the precision/sensitivity values for the 17 loci were 99.0% and 94.1% respectively.

[00121] **FIG. 11** shows a table illustrating Rare CNV TaqMan Validations. In this example validation, the 165 samples tested for common CNV loci were not randomly selected in an attempt to minimize the number of samples required to ensure that each locus had a reasonable number of samples with non-diploid copy number (e.g., which is why several loci in the table have exactly 10 predicted CNV).

[00122] **FIG. 12** shows a table illustrating common CNV TaqMan Validations. **FIG. 13** is a graph illustrating comparison of CLAMMS and TaqMan copy number predictions for the LILRA3 common variant locus. **FIG. 14** is a graph illustrating

comparison of CLAMMS and TaqMan copy number predictions for the LILRA3 common variant locus.

[00123] While the methods and systems have been described in connection with preferred embodiments and specific examples, it is not intended that the scope be limited to the particular embodiments set forth, as the embodiments herein are intended in all respects to be illustrative rather than restrictive.

[00124] Unless otherwise expressly stated, it is in no way intended that any method set forth herein be construed as requiring that its steps be performed in a specific order. Accordingly, where a method claim does not actually recite an order to be followed by its steps or it is not otherwise specifically stated in the claims or descriptions that the steps are to be limited to a specific order, it is in no way intended that an order be inferred, in any respect. This holds for any possible non-express basis for interpretation, including: matters of logic with respect to arrangement of steps or operational flow; plain meaning derived from grammatical organization or punctuation; the number or type of embodiments described in the specification.

[00125] Throughout this application, various publications are referenced. The disclosures of these publications in their entireties are hereby incorporated by reference into this application in order to more fully describe the state of the art to which the methods and systems pertain.

[00126] It will be apparent to those skilled in the art that various modifications and variations can be made without departing from the scope or spirit. Other embodiments will be apparent to those skilled in the art from consideration of the specification and practice disclosed herein. It is intended that the specification and examples be considered as exemplary only, with a true scope and spirit being indicated by the following claims.

CLAIMS

What is claimed is:

1. A method comprising:

receiving, by a computing device, a sample coverage data set comprising a plurality of genomic sequences obtained from sequencing of nucleic acid samples of a subject, and sample sequencing quality control (SSQC) metrics;

grouping, by the computing device, sets of sequencing quality control (SQC) metrics into a multidimensional tree data structure according to similarity, wherein each set of SQC metrics is associated with a respective reference coverage data set that comprises a plurality of genomic regions and read depths;

selecting a reference panel of the reference coverage data sets using the multidimensional tree data structure, wherein the selected reference coverage data sets have SQC metrics similar to the SSQC metrics;

normalizing, by the computing device, the sample coverage data and the reference panel;

fitting, by the computing device, the normalized reference panel to a mixture model at each of the plurality of genomic regions to generate an expected coverage distribution at each of the plurality of genomic regions;

identifying one or more copy number variants (CNVs) by using the computing device to compare, according to a Hidden Markov Model (HMM), the normalized sample coverage data set to the expected coverage distribution at each of the plurality of genomic regions from the mixture models; and

outputting, by the computing device, the one or more CNVs.

2. The method of claim 1, wherein selecting the reference panel of reference coverage data sets using the multidimensional tree data structure comprises:

defining a distance metric between the SSQC metrics ; and

selecting the reference panel of the reference coverage data sets based on the distance metric.

3. The method of claim 1, wherein grouping the plurality of reference coverage data sets comprises use of a clustering algorithm, a classification algorithm, or a combination thereof.

4. The method of claim 1, wherein grouping the sets of SQC metrics comprises use of a k-Nearest Neighbors (knn) algorithm and wherein the method further comprises:

scaling the sets of SQC metrics;

scaling the sets of SSQC metrics;

wherein grouping the sets of SQC metrics into the multidimensional tree data structure according to similarity comprises, generating a k -d tree based on the scaled set of SQC metrics;

adding the scaled SSQC metrics to the k -d tree; and

wherein selecting the reference panel of reference coverage data sets using the multidimensional tree data structure comprises, identifying a predetermined number of nearest neighbor scaled SQC metrics to the scaled SSQC metrics.

5. The method of claim 1, further comprising dividing a plurality of genomic regions of the sample coverage data set into one or more calling windows.
6. The method of claim 5, wherein normalizing the sample coverage data set comprises:
- determining raw coverage for a calling window w ;
 - determining a median coverage for the sample coverage data set across the one or more calling windows conditional on a GC-fraction of the calling window w ;
 - and
 - dividing the raw coverage by the median coverage, resulting in the normalized sample coverage data set.
7. The method of claim 6, wherein determining a median coverage for the sample coverage data set across the plurality of windows conditional on a GC-fraction of the calling window w comprises:
- binning the one or more calling windows by GC-fraction, resulting in a plurality of bins;
 - determining a median coverage for each bin of the plurality of bins; and
 - determining a normalizing factor for each distinct possible GC-fraction using a linear interpolation between the median coverage for two bins nearest to the calling window w .
8. The method of claim 5, further comprising filtering the sample coverage data set.
9. The method of claim 8, wherein filtering the sample coverage data set comprises:
- filtering one or more calling windows based on a mappability score of a genomic region of a plurality of genomic regions; and
 - filtering the one or more calling windows based on occurrence of a calling window in a multi-copy duplication genomic region.
10. The method of claim 9, wherein filtering the one or more calling windows based on the mappability score comprises:
- determining a mappability score for each genomic region of the plurality of genomic regions; and

excluding a calling window of the one or more calling windows that contains the genomic region of the plurality of genomic regions if the mappability score of the genomic region of the plurality of genomic regions is below a predetermined threshold.

11. The method of claim 9, wherein filtering the one or more calling windows based on occurrence of a calling window in a multi-copy duplication genomic region comprises:

excluding a calling window of the one or more calling windows if the calling window of the one or more calling windows occurs within a region where multi-copy duplications are known to be present.

12. The method of claim 1, wherein fitting the reference panel to the mixture model comprises:

determining a plurality of mixture models, one for each of the plurality of genomic regions, wherein each component of the plurality of mixture models comprises a probability distribution that represents an expected normalized coverage conditional on a particular copy number; and

fitting the normalized reference panel to the plurality of mixture models using an expectation-maximization algorithm to determine a likelihood for each copy number at each of the one or more calling windows, wherein the normalized reference panel is input to the expectation-maximization algorithm.

13. The method of claim 12, wherein identifying one or more copy number variants (CNVs) by using the computing device to compare, according to a Hidden Markov Model (HMM) the normalized sample coverage data set to the expected coverage distribution at each of the plurality of genomic regions from the mixture models comprises:

inputting the normalized sample coverage data set for each calling window of the one or more calling windows into the HMM;

determining one or more emission probabilities of the HMM based on the mixture model; and

identifying a calling window of the one or more calling windows as a CNV if a maximum likelihood sequence of states of the calling window is non-diploid.

14. The method of claim 13, wherein determining one or more emission probabilities of the HMM based on the mixture model comprises:

determining a probability of observing a normalized coverage value x , at a calling window w of the one or more calling windows, given HMM state s , based on a component of the mixture model for w that corresponds to state s .

15. The method of claim 13, wherein identifying the calling window of the one or more calling windows as a CNV if a maximum likelihood sequence of states of the calling window is non-diploid comprises:

performing a Viterbi algorithm in a 5' to 3' direction on a genomic region of the plurality of genomic regions;
performing the Viterbi algorithm in a 3' to 5' direction the genomic region of the plurality of genomic regions; and
identifying the calling window of the one or more calling windows as a CNV if the genomic region of the plurality of genomic regions associated with the calling window has a most-likely state of non-diploid in the 5' to 3' direction and the 3' to 5' direction.

16. A method comprising:

providing, by computing device, sample coverage data comprising a plurality of genomic sequences obtained from sequencing of nucleic acid samples of a subject and sample sequencing quality control (SSQC) metrics;
receiving an indication of reference coverage data, wherein the reference coverage data is selected by:
grouping, by the computing device, sets of sequencing quality control (SQC) metrics into a multidimensional tree data structure according to similarity, wherein each set of SQC metrics is associated with a

respective reference coverage data set that comprises a plurality of genomic regions and read depths;

selecting a reference panel of the reference coverage data sets using the multidimensional tree data structure, wherein the selected reference coverage data sets have SQC metrics similar to the SSQC metrics;

selecting one or more filters to apply to the sample coverage data and the reference panel to normalize the sample coverage data and the reference panel;

requesting fitting of the normalized reference panel to a mixture model at each of the plurality of genomic regions to generate an expected coverage distribution at each of the plurality of genomic regions;

requesting identifying one or more copy number variants (CNVs) by using the computing device to compare, according to a Hidden Markov Model (HMM) the normalized sample coverage data set to the expected coverage distribution at each of the plurality of genomic region from the mixture models; and

receiving an indication of the one or more CNVs.

17. The method of claim 16, wherein fitting the normalized reference panel to the mixture model comprises:

- determining a plurality of mixture models, one for each of the plurality of genomic regions, wherein each component of the plurality of mixture models comprises a probability distribution that represents an expected normalized coverage conditional on a particular copy number; and
- fitting the normalized reference panel to the plurality of mixture models using an expectation-maximization algorithm to determine a likelihood for each copy number at each of one or more calling windows, wherein the normalized reference panel is input to the expectation-maximization algorithm.

18. The method of claim 17, wherein identifying one or more copy number variants (CNVs) by using the computing device to compare, according to a Hidden Markov Model (HMM) the normalized sample coverage data set to the expected coverage distribution at each of the plurality of genomic regions from the mixture models comprises:

- inputting the normalized sample coverage data set for each calling window of the one or more calling windows into the HMM;
- determining one or more emission probabilities of the HMM based on the mixture model;
- and
- identifying a calling window of the one or more calling windows as a CNV if a maximum likelihood sequence of states of the calling window is non-diploid.

19. An apparatus for performing the method of claim 1.

20. An apparatus for performing the method of claim 16.