CNV Methods

File format v2.0

Software v2.0.0

September, 2011

RM_CNV-02

Table of Contents

Introduction

Complete Genomics CNV analysis pipeline employs read-depth analysis to estimate the genomic copy number at a given region based on the count of reads aligned to that region. The goal of this *Methods* document is to describe the processing steps and algorithms used in the CNV analysis pipeline.

This document is written with respect to the Complete Genomics Assembly Pipeline version 1.12. We recommend that you are familiar with the following additional documentation as preparation for reading this document:

1. *Data File Formats* — A description of the organization and content of the format for complete genome sequencing data delivered by Complete Genomics. [www.completegenomics.com/documents/DataFileFormats-100357139.html]
2. *Release Notes* — The Assembly Pipeline Release Notes indicate new features and enhancements by release. [www.completegenomics.com/documents/ReleaseNotes-100358389.html]
3. *Human Genome Sequencing Using Unchained Base Reads on Self-assembling DNA Nanoarrays* — *Science* publication describing Complete Genomics proprietary sequencing technology, including brief methods and performance of our Assembly Pipeline. [www.completegenomics.com/knowledge-center/publications]

Additional documentation is available in the Support section of the Complete Genomics website:

www.completegenomics.com/customer-support/support/

Steps in CNV Calling

The determination of CNV calling for normal and tumor samples begins with three steps:

1. Computation of sequence coverage
2. Estimation and correction of bias in coverage
   a. Modeling of coverage bias
   b. Correction of modeled bias
   c. Coverage smoothing
3. Normalization of coverage by comparison to a baseline sample or set of samples

Following normalization of coverage, both normal and tumor samples are segmented using Hidden Markov Models (HMM), but with a different model for each sample types:

4. HMM segmentation, scoring and output

Finally, normal samples are subjected to a "no-calling" process that identifies CNV calls that are suspect:

5. Population-based no-calling/identification of low-confidence regions

Computation of Sequence Coverage

All mate-pair constraint-satisfying paired-end (i.e., full DNB) mappings are used to compute sequence coverage. In the majority of cases, a unique paired-end mapping contributes a single count to each base of the reference that is aligned to the DNB. A reference base aligned to a non-unique paired-end mapping is weighted based on the estimated probability that the mapping is the correct location of the DNB. Weighting of DNBs in proportion to the confidence in each mapping provides the ability to give reasonable coverage estimates in regions where mappings are non-unique.

Formally, each position *i* of the reference genome *R* receives the following coverage value c_i:

$$(1) \qquad c_i = \sum\nolimits_{m \in M_i} P(DNB|R,m) / (\propto + \sum\nolimits_{n \in N(m)} P(DNB_m|R,n))$$

where:

- M_i is the set of mappings over all DNBs such that a called base in each mapping is aligned to position *i*, DNB_m is the DNB described by mapping *m*
- *N(m)* is the set of all mappings involving DNB_m
- α is the probability that a DNB is generated in a manner that does not allow it to map to the reference. (See our *Science* publication for description of α and of $P(DNB|R,m)$)

Estimation and Correction of Bias in Coverage

Genome sequencing using the Complete Genomics pipeline results in coverage bias that may affect estimation of copy number. One important element of the bias involves the GC content over intervals approximately the length of the initial DNA fragment that becomes a DNB (i.e., approximately 400bp), though other factors are known as well. It is generally preferred to model and correct for such biases prior to or as part of copy number estimation. Additionally, it is desirable to smooth out short scale fluctuations in coverage, which may be at least partly specific to an individual library.

The following summarizes the GC normalization process:

- Compute GC content for the 1000-base window centered at each point of the genome (excluding positions less than 500 bases from the ends of contigs). *isGC(j)* is 1 if the base at position *j* is G or C, and 0 otherwise. We compute the GC content at position *i*, gc_i, as follows:

(2) $$gc_i = \sum_{j=i-500}^{i+499} isGC(j)$$

  Positions less than 500 bases from either end of a contig are not considered during the estimation of the GC correction factors.

- For each possible GC value γ, determine the mean coverage $\tilde{C}_\gamma$ for positions with $gc_i = \gamma$, letting n_γ be the number of positions *i* in the genome with $gc_i = \gamma$:

(3) $$\tilde{C}_\gamma = \frac{\sum_{gc_i=\gamma} c_i}{n_\gamma}$$

  In practice, we exclude positions with coverage > 500.

- Repeat the mean coverage calculation for a simulation, using the "*" superscript to indicate simulation results.

(4) $$\tilde{C}_\gamma^* = \frac{\sum_{gc_i=\gamma} c_i^*}{n_\gamma}$$

  Note that the result is not entirely flat due to factors such as the GC content of ubiquitous repeats and microsatellite regions not being the similar to the the genome as a whole.

- Compute the ratio of sample coverage to simulation coverage for each GC value, adjusting for the overall average coverage of the sample and the simulation ($\bar{c}$ and $\bar{c}^*$ respectively).

(5) $$f_\gamma = \frac{\tilde{C}_\gamma}{\tilde{C}_\gamma^*} * \frac{\bar{c}^*}{\bar{c}}$$

- Obtain a smoothed coverage ratio as a function of GC, $\hat{f}_\gamma$.

(6) $$\hat{f}_\gamma = LOESS(f(\gamma))$$

  A local regression operation, LOESS smoothing is performed except in numerically unstable regions where LOWESS is performed instead. Smoothing is performed by invoking the R package.

- GC-correct the (single-base) coverage at every position of the genome:

(7) $$c_i' = c_i / \hat{f}_{gc_i}$$

  For the unsampled edges of the contigs, "missing bases" are filled in with genome-wide average GC content. If the GC content of the window for a given position is too extreme (that is, < 20% or > 80% GC), the coverage value is treated as unknown (missing data).

Window-smoothing is performed by taking the mean of $\hat{c}_i$ for each position *i* within a given window. Windows are tiled (adjacent, non-overlapping), with the choice of window boundaries as described in "Appendix: Window Boundary Definition." That is, for a window corresponding to the interval [*i,j*], we compute the average corrected coverage $\bar{c}_{i,j}'$ as

$$(8) \qquad \bar{c}'_{i,j} = \sum_{k=i}^{j-1} c'_k/(j-i)$$

For notational simplicity, we drop the "*j*" subscript and use $\bar{c}'_i$ in place of $\bar{c}'_{i,j}$, as there is at most one window starting at position *i.*

Normalization of Coverage by Comparison to a Baseline Sample

Bias in coverage not corrected by the adjustments described above may be taken into account by comparison to a baseline sample. However, to obtain coverage proportional to absolute copy number, it is necessary to adjust the baseline sample according to the copy number in the sample. Letting d'_i and p_i be the corrected coverage and ploidy at of the window beginning at *i* for the baseline sample, and $\tilde{d}$ be an estimate of the typical diploid coverage for the baseline sample[1], we determine a bias correction factor b_i as:

$$(9) \qquad b_i = \frac{\tilde{d}}{d'_i} * \frac{p_i}{2}$$

Normalized corrected coverage $\bar{c}''_i$ is then computed as:

$$(10) \qquad \bar{c}''_i = \bar{c}'_i * b_i$$

If $p_i = 0$ (in which case d_i is due to mismappings and not a reliable indicator of coverage behavior in this location), we treat $\bar{c}''_i$ as missing.

In practice, we typically use a group of samples, rather than a single sample, as the baseline, to reduce sensitivity to fluctuations due to sampling (statistical noise) or due to library-specific biases. The production pipeline uses the following for the set of baseline samples *S*:

$$(11) \qquad p_i = \sum_{s \in S} p_i^s$$

$$(12) \qquad d_i = \sum_{s \in S} d_i^s$$

where p_i is the ploidy at window *i*. Ideally, this would be the true ploidy for the baseline sample for this window. Since we do not know this, we must estimate it in some manner. The current baseline generation process includes CNV-calling for each baseline genome, using a simulation where copy number is 2 for autosomes and gender-appropriate for sex chromosomes. Using a simulation as the baseline provides an indirect means of correcting for variation in mapability of the genome, such as regions corresponding to high-copy, high-identity repeats. However, it does not address coverage bias due to biochemistry. In regions of moderate coverage bias and where the length scale of bias is short relative to the length of the window, the baseline genome(s) will be called at the correct ploidy and consequently the correction factor will appropriately compensate for the bias. However, regions with a sustained bias resulting in coverage being > 50% of the diploid-average away from the true ploidy will have their copy number miscalled on the baseline genomes; this results in a baseline "correction" that reinforces the tendency to call CNVs at this location and therefore results in robust/consistent miscalling of abnormal ploidy. Future changes in this process in this area would be based on improved estimation of the ploidy of baseline genomes; this could be based on external information (such as chip-based CNV calls), manual curation, or an automated process that attempts to determine population patterns by simultaneous analysis of multiple genomes. The current pipeline typically labels such regions as "invariant" (see "Annotation of Invariant Regions").

If no samples are used as the baseline, then we simply set $\bar{c}''_i = \bar{c}'_i$.

[1] In practice, we take $\tilde{d}$ as the 45% percentile of windows in the autosome.

HMM Segmentation, Scoring and Output for Normal Samples

There are many approaches to segmenting a quantitative time series that can be applied to calling CNVs—that can be applied to coverage data produced by the three initial steps. HMMs provide one such approach with certain appealing properties (obvious model fitting methods, flexible models, natural confidence measures, ability to constrain models, ability to incorporate a variety of models of coverage generation), wherein states correspond to copy number levels, emissions are some form of coverage (observed/corrected/relative), and transitions between states correspond to changes in copy number.

In the current Complete Genomics CNV-calling process, GC-corrected, window-averaged, normalized coverage data, $\bar{c}_i''$, is the source for an HMM whose states correspond to integer ploidy (copy number). Copy number along the genome is estimated as the ploidy of the sequence of most likely states according to the model. Various scores are computed based on the posterior probabilities generated by the HMM.

Model definition

A fully-connected HMM with states corresponding to ploidy 0 though ploidy 9, and ploidy “10 or more” is defined by a matrix of priors or initial state probabilities, transition probabilities, and the emission probabilities.

Coverage distributions (i.e., state emission probabilities) are modeled as a negative binomial, which can be parameterized by the mean and variance of the distribution for each state.[2]

Model Estimation

In principle, model parameters can all be estimated by expectation-maximization (EM) by the Baum-Welch algorithm. In practice, unconstrained estimation, especially of coverage distributions, has not given satisfactory results. Instead, initial values are chosen and subsequent updates are constrained to reflect the following assumptions:

- Coverage depends on the number of copies of a given reference segment in the genome of interest.
- Copy number is assumed to be integer-valued.
- Coverage is assumed to be linearly related to copy number.
- The majority of the genome is assumed to be diploid, so that the “typical” value for the autosome can be used to fix the mean coverage for ploidy = 2.
- For states corresponding to ploidy >= 1, the standard deviation of a state is proportional to the mean of the state.
- For the state corresponding to ploidy = 0, a separate variance estimate is used to allow for the impact of mis-mappings and non-unique mappings.

Given these constraints, there are only two free parameters regarding the coverage distributions, namely a single value relating coverage to standard deviation for ploidy >= 1, and another variance parameter for ploidy = 0.

Additionally, transition probabilities could be estimated from the data, but the risk of over-fitting is high. Consequently, a set of default values is used, such that the probability of transition from one state to another at any “time” (window) is set to 0.003 (and the probability of remaining in a given state is taken as 1-0.003*10 = 0.97).

Initial state probabilities are all set to 1 divided by the number of states.

[2] The negative binomial is sometimes considered an “over-dispersed poisson” distribution, i.e. similar in some regards, but with higher variance.

The mean of the emission (coverage) distribution for a state with ploidy *n* is initialized as follows, except as noted below:

$$(13) \qquad \mu_n = n * \tilde{\bar{c}}''/2$$

where $\tilde{\bar{c}}''$ is the median of $\bar{c}_i''$for all positions at which normalized, smoothed, corrected coverage has been computed. To allow for the presence of some apparent coverage due to mis-mappings, we set $\mu_0 = 0.1 * \tilde{\bar{c}}''$. Initial estimates of the means are not updated during subsequent model fitting.

The initial variance of the ploidy 2 state is set to:

$$(14) \qquad \sigma_2^2 = 3 * \mu_2 = 3 * \tilde{\bar{c}}''$$

Variance for other states is set so that standard deviations will be proportional to the means:

$$(15) \qquad \sigma_n^2 = \sigma_2^2 * (n/2)^2$$

The variance-determining parameters are updated by EM until the model has "converged", that is, the change in log likelihood of the data given the model between successive iterations is sufficiently small.

Ploidy Inference, Segmentation, and Scoring

After convergence of the estimation procedure, the usual HMM inference computations are performed. The final result is based on the most likely state at each position. (A standard alternative is to assign ploidy corresponding to the states of the most likely single path.)

The "called ploidy" of each position in the input (*calledPloidy* column in the resulting data) is taken to be that of the most likely state at that position. The *ploidyScore* is estimated as a phred-like score reflecting the confidence that the called ploidy is correct. The *CNVTypeScore* is estimated as a phred-like score reflecting the confidence that the called ploidy correctly indicates whether the position has decreased ploidy, expected ploidy, or increased ploidy relative to the nominal expectation (diploid except that sex chromosomes in males are expected haploid).

A "segment" is a sequence of adjacent positions that have the same called ploidy. The *begin* and *end* positions of the segment are considered as the start and end positions of the first and last windows respectively. Each segment is given a *ploidyScore* equal to the average of the ploidy scores for the positions in the segment, and a *CNVTypeScore* that is the mean of the CNV type scores for the positions in the segment. (Taking the mean of log-likelihood ratios is equivalent to taking the log of the geometric mean of the likelihood ratios.)

See "Appendix: Score Computation" for precise definitions and justification of the scores.

HMM References

For background information on Hidden Markov Models, we recommend the following reading:

Wikipedia:

▪ HMMs	en.wikipedia.org/wiki/Hidden_Markov_model
▪ Baum-Welch	en.wikipedia.org/wiki/Baum%E2%80%93Welch_algorithm
▪ Forward-Backward	en.wikipedia.org/wiki/Forward-backward_algorithm
▪ Viterbi algorithm	en.wikipedia.org/wiki/Viterbi_algorithm

A classic review:

Rabiner, L.R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE, 1989, 77.2:257-286:

www.ece.ucsb.edu/Faculty/Rabiner/ece259/Reprints/tutorial%20on%20hmm%20and%20applications.pdf

Modifications to HMM Segmentation and Output for Tumor CNV Calling

Copy-number calling in tumor samples poses several challenges to the methods described so far. Due to the possibility of high average copy number, it is not advisable to assume that diploid ("normal") regions of the genome will have coverage near the sample median. Even if we could determine the typical coverage for a diploid region (such as by analysis of minor allele frequency), the expected change in coverage for an increase or decrease of a single copy is not necessarily 50% of this value. This is due to the possibility of an unknown amount of contamination from adjacent or mixed-in normal cells ("normal contamination"). And even among tumor cells, a segment of the genome may not be characterized by an integer copy number, due to tumor heterogeneity.

Consequently, it is useful to relax the assumptions that constrain the coverage levels of the states of the model, allowing the ratios of coverage to be continuously valued. This increases the challenge of finding the correct values and also introduces the problem of deciding how many states to include, leading the analysis to include a model selection component. Further, it poses problems for the paradigm of labeling a given region with an integer copy number. Consequently, we modify the goal of the analysis to segment the genome into regions of uniform "abundance class", without forcing an interpretation of a given class as being an integer copy number.

In theory, we could simply fit HMMs with varying numbers of states, using EM to determine the expected coverage level for each state and choosing the number of states that gives the best fit. In practice, unconstrained estimation of model parameters for any given number of states is not a robust process. Consequently, we introduce the following additional steps:

- Generate an initial model by estimating the number of states and their means based on the overall coverage distribution.
- Optimize the initial model by sequentially adding states to the model.
- Sequentially remove states from the model.

Initial Model Generation

The distribution of (corrected, normalized, window-averaged) coverage over the whole genome to be segmented is a mixture of the distributions of the different abundance classes. One approach to identifying distinct abundance classes is to identify peaks in the smoothed whole genome coverage distribution.[3] An improvement over direct identification of peaks is realized by applying the quantile function for a normal distribution to the cumulative distribution function(cdf), and then take the difference between successive values, prior to smoothing and peak detection. This latter approach improves sensitivity for identification of small peaks outside the central abundance classes.

Given a histogram of coverage $H = h_0, h_1, h_2, ... H_n$ where h_i is the number of positions with coverage i and n is the smallest value such that less than 0.001 of the full histogram is truncated, and letting $Q(c)$ be the quantile function for a normal distribution, the resulting peak locations, $P(as\ computed\ below)$, are used as states in an initial model, with expected coverage values equal to the center of each peak.

$$(16) \qquad N = \sum_{i=0}^{i=n} h_i$$

[3] Another approach would be to identify a mixture model which closely fits the observed coverage distribution.

(17) $c_i = h_i/N$

(18) $q_i = Q(c_i)$

(19) $d_i = q_i - q_{i-1}$

(20) $D = d_1, d_2, \ldots, d_n$

(21) $S = smooth(D)$

(22) $s_i = S(i)$

(23) $$m(i) = \begin{cases} 1 \text{ if } s_i > s_{i-1} \text{ and } s_i > s_{i+1} \text{ and for } i \text{ an integer} \\ \qquad\qquad 0 \text{ otherwise} \end{cases}$$

(24) $\mathrm{P} = \{\, \mathrm{i} \,|\mathrm{m(i)} = 1 \text{ and } \mathrm{d_i} > .002\}$

Model Optimization

Once an initial model has been inferred in this manner and variances estimated via EM as described above, the model is refined iteratively. First, additional states are evaluated. The addition of a state is evaluated between each successive pair of states (abundance classes, ordered by expected coverage), accepting the addition if the improvement in likelihood (Pr(data|model)) exceeds some threshold. That is, between each successive pair of states *i* and *j* with expected coverage c_i and c_j we attempt to add a state *i'* with initial coverage $c_{i'}=(c_i+c_j)/2$. We optimize $c_{i'}$ using EM, holding the expected coverage levels of all other (pre-existing) states fixed. If the optimization results in a value outside the interval (c_i ,c_j), or if the reduction Pr(data|model) does not exceed an acceptance threshold, the addition is rejected; otherwise, the addition is accepted. If an addition is accepted, addition of a further state between *i* and *i'* is attempted, with recursion until the further addition is not accepted. Once addition between all pairs of successive states is rejected, the addition process is terminated. Second, removal of states is evaluated. States are removed from the model one at a time and the resulting model is optimized using EM; if the resulting model is not sufficiently worse than the previous model, then the state removal is accepted.

Clearly there are many variations on the preceding process. For instance, we might try removal of each state from the maximal model to determine which has the least impact; the state is removed and the process repeated.

Once the model is selected and variance parameters are optimized, segmentation and segment scoring are as described for normal samples. In brief, continuous segments of positions with the same most-likely state are reported, with scores indicating the average over positions in the segment of the probability of a classification error. Population-based no-calling (described in the following section) is not performed for tumor CNV calling.

Population-Based No-Calling

The HMM-based calls described in the previous sections typically contain a variety of inferred CNVs that are either artifacts or of lesser interest. Primarily, these arise in one of two situations:

- The reference genome sequence does not provide an explanation for coverage patterns in most or all sample genomes, with most or all sample genomes matching one another.
- There is more variation in coverage than can be explained by a small number of discrete ploidy levels.

The utility of CNV inference may be increased by identifying and annotating such regions. In the explanations below, regions so-annotated are considered to be "no-called" in the sense that a discrete estimate of ploidy may not be given for these regions. Note that population-based no-calling described in this section is not applied to tumor samples.

Such behaviors may result from multiple causes; some of the possible mechanisms include:

- **Errors in the reference genome:** For example, two contigs may in fact overlap one another—correspond to a single genomic interval—in most or all genomes. In this case, the two contig ends may consist in part of highly similar sequences that are otherwise unique, causing DNBs to map to both locations. Observed/measured coverage will be reduced, leading to an apparent copy number reduction. Alternatively, most or all sample genomes may contain a duplication that is not present in the reference. In this case, observed coverage will be elevated over the portion of the reference corresponding to the duplicated segment, leading to a copy number increase relative to the reference but not a true polymorphism.
- **Uncorrected coverage bias:** A region that is substantially over- or under-represented in the sequencing results may appear to be a CNV relative to the reference. To retain the ability to make absolute copy number inferences, baseline correction as described in "Normalization of Coverage by Comparison to a Baseline Sample" is done, taking into account an initial copy number inference for the baseline genome(s). A consequence of this may be that regions that are severely biased in the baseline as well as the sample of interest may be interpreted as CNVs. The signature of this sort of event will be that most or all samples will show similarly elevated or suppressed coverage patterns.
- **Analysis artifacts:** Though rare, there are occasional mapping artifacts that can result in a large number of spurious mappings at a given location. Such artifacts may result from particular arrangements of variations from the reference in repeated segments, such that the wrong reference copy of a repeat is more similar to the sequence of the sample of interest. These can result in a substantial spike in coverage at certain locations on the reference, in a manner that is dependent on the variations present in a given sample.
- **Segmental duplications and tandem repeats:** A segment that is present in duplicated form in the reference and subject to population variation may result in changes in coverage—among samples—that are smaller than typical of a copy number gain or loss in unique sequence. In the limit, sufficient variability in the population regarding a high-copy sequence type may result in an essentially continuous range of coverage values across a large number of samples.
- **Unstable estimation due to extreme correction factors or very low raw coverage:** Examples include: 1) regions where coverage is very low due to GC correction, and the GC correction factors are correspondingly large, so that noise in the coverage estimate is amplified by the correction factors; 2) regions where coverage is very low due to mapping overflow, in simulations as well as real data, leading to large correction terms in the baseline bias correction factors; 3) regions where nearly all baseline genomes have ploidy 0.

Identification of such regions could be conducted in various ways. Ultimately, manual curation of coverage patterns at individual locations could be highly effective, but may be prohibitive in some circumstances due to lack of data, degree of effort, and/or process instability. Use of sequence similarity and/or structural annotations has some promise, as a large fraction of problematic regions in practice correspond to known repetitive portions of the reference genome (segmental duplications, self-chains, STRs, repeat-masker elements); however, since many real copy number polymorphisms occur in such regions, it is unappealing to broadly exclude such segments and challenging to find criteria that are more selective. Thus, it is desirable to be able to identify problematic regions directly from coverage data.

Two types of coverage patterns typify several of the above circumstances. The first involves regions where coverage is more variable than can be explained by a small number of discrete ploidy levels ("hypervariable"). The second involves regions where coverage is not as expected of a euploid region matching the reference but it is similar among all samples ("invariant").

Given a substantial number of genomes (for example, 50 or more), the "background set" of summary statistics on bias-corrected and smoothed, but un-normalized coverage data is sufficient to meaningfully (if heuristically/imperfectly) separate the genome into well-behaved regions, hypervariable regions and invariant regions. The following summary statistics computed for every genomic position i over a set G of n genomes can be used in this way. Let $\bar{c}'_{i<x>}$ for $1 \leq x \leq n$ be the x'th order statistic of $\bar{c}'_i(g), \ \ g \in G$, i.e. the x'th smallest corrected and smoothed coverage at position i among the genomes in the background set.

Median $\tilde{m}_i$:

$$(25) \qquad \tilde{m}_i = \begin{cases} \bar{c}'_{i<\frac{n+1}{2}>} & \text{for } n \text{ odd} \\ \frac{(\bar{c}'_{i<(n/2)>}+\bar{c}'_{i<n/2+1>})}{2} & \text{for } n \text{ even} \end{cases}$$

Spread s_i:

$$(26) \qquad s_i = \bar{c}'_{i<n>} - \bar{c}'_{i<1>} \ = \ \max_{g \in G} \bar{c}'_i(g) - \ \min_{g \in G} \bar{c}'_i(g)$$

Clustering coefficient q_i:

$$(27) \qquad q_i = \frac{\min_{1 \leq q<r<s<n} \ \ \mathrm{SSE}(i,0,q)+\mathrm{SSE}(i,q,r)+\mathrm{SSE}(i,r,s)+\mathrm{SSE}(i,s,n)}{\mathrm{SSE}(i,0,n)}$$

where $\mathrm{SSE}(i,x,y)$is the sum of squared error for $\bar{c}'_{i<x+1>}, \dots, \bar{c}'_{i<y>}$:

$$(28) \qquad C_{i,x,y} = \sum\nolimits_{x<t\leq y}(\bar{c}'_{i<x+1>}, \dots, \bar{c}'_{i<y>})/(y-x)$$

$$(29) \qquad \mathrm{SSE}(i,x,y) = \sum\nolimits_{x<t\leq y} \ \left(\bar{c}'_{i<t>} - C_{i,x,y}\right)^2$$

Given these summary statistics, we may define criteria for labeling positions as hypervariable or invariant.

Note: The clustering coefficient will be recognizably related to the F statistic of ANOVA, the between-group sum of squares / within-group sum of squares ratio. However, in a standard ANOVA, the separation into groups is done based on a categorical covariate distinct from the dependent variable whose variance is being measured, whereas here we are splitting samples into groups based precisely on the variable—coverage—whose variance is being measured. Thus, standard significance of the F statistic is not valid.

Annotation of Hypervariable Regions

A position that satisfies all of the following four criteria may be labeled "hypervariable," rather than being marked as a CNV or classified as euploid:

1. The position would be called a CNV/aneuploid by the HMM inference process described in "HMM Segmentation, Scoring and Output for Normal Samples."
2. Coverage values in the background set are not clustered in ways suggesting simple polymorphism in the population (i.e., coverage values across background set are separated into discrete levels that correspond to integer copy numbers). Formally, for Q a value that can be chosen empirically as:

$$q_i > Q$$

3. The range of coverage values at this position in the background set is wider than is seen at most of the (euploid) genome. Formally, for *S* a value that can be chosen empirically as:

$$s_i/\widetilde{m}_i > S$$

4. The observed coverage for the sample of interest falls within the range of values seen in the background set, or outside the observed range by a small absolute amount (i.e., an amount that could readily be explained by sampling or process variation). Formally, for *R* and *X* values that can be chosen empirically as:

$$|\bar{c}_i' - \widetilde{m}_i| < \ min(s_i * R, X)$$

Annotation of Invariant Regions

A position that satisfies all of the following three criteria may be labeled “invariant” (rather than being marked as a CNV):

1. The position would be called a CNV/aneuploid by the HMM segmentation process described above.
2. Coverage values in the background set are not clustered in ways suggesting simple polymorphism in the population. Formally, for *Q* a value that can be chosen empirically as:

$$q_i > Q$$

3. Coverage at this position across the background samples shows low variability, suggesting both absence of a high-minor-allele-frequency polymorphism in the population and low process variation (artifact). Formally, for *S* a value that can be chosen empirically as:

$$s_i/\widetilde{m}_i < S$$

4. The observed coverage for the sample of interest falls within the range of values seen in the background set, or outside the background range by a small absolute amount (i.e., an amount that could readily be explained by sampling or process variation). Formally, for a value *R* that can be chosen empirically as:

$$|\bar{c}_i' - \widetilde{m}_i| < \ min(s_i * R, X)$$

Refinement of Annotations

The above criteria may cause CNV calls to be overly fragmented into alternating called and no-called segments. It may be desirable to permit short intervals that would be “no-called” (i.e. annotated as “hypervariable” or “invariant”) based on the criteria above to be allowed to be called (left unannotated) if the observed coverage is sufficiently similar to a flanking interval that is not annotated. Concretely, we may suppress the “hypervariable” or “invariant” labeling of intervals less than *L* bases that satisfy the above criteria but are part of longer segments in the HMM output. Currently, *L*= 10 kb.

Selection of Cutoff Values

Cutoffs *Q, S, R, X* and *L* in the above criteria may be selected based on analysis of a subset of initial CNV calls and comparison to genome-wide distributions on the background coverage summary statistics. Given a classification of an initial set of CNV calls (the “training set”) into those that are suspect (to be labeled “hypervariable” or “invariant”) and those that are believed to be true

CNVs, as well as summary statistics for the entire genome (that is, of selected positions spaced along the genome, such as those resulting from the windows described above), we wish to identify cutoffs that are near optimal with regard to the following criteria:

- Most of the genome is called either euploid or CNV/aneuploid (only a small fraction of the genome is no-called/annotated as hypervariable or invariant).
- Most of the problematic regions in the "training set" are no-called.
- Most of the trusted regions in the training set are called (not annotated).

The training set can be derived based on manual curation of a collection of preliminary CNV calls. This curation may involve manual inspection of coverage profiles to identify calls and comparison to external datasets of putative CNVs identified by independent means.

Candidate values of *Q, R, S* and *L* may be evaluated by determining concordance with the training set or a separate test set, as well as the fraction of the genome that is no-called. The final choice of cutoffs will involve a tradeoff between completeness of calling (fraction of the genome called, and/or overlap with an independent set of trusted CNV calls) and the amount of problematic CNV calling.

Modifications to Normalization for Somatic (Paired-sample) CNV Calling

To identify somatic CNVs, e.g. between a tumor sample (the 'target') and a matched normal sample (the 'baseline'), a variation on the procedure for normalization of GC-corrected coverage described above is employed. The GC-corrected coverage of the target is normalized by the GC-corrected coverage of the baseline, without reference to CNV calls in the latter. This might be considered the naïve or standard approach to sample/sample coverage normalization. Concretely, we redefine the normalization constant for a given window *i*, b_i, replacing equation (9) with the following:

$$(30) b_i = \frac{\tilde{d}}{d'_i}$$

HMM-related calculations proceed as described above, with the caveat that population-based no-calling is not performed. The resulting CNV calls are "somatic" in the sense that CNVs relative to the reference genome that are shared by the baseline and the target will typically not be called, as the normalization process will make the tumor coverage appear "typical" in such a region. One negative consequence is that somatic CNVs that overlap germline CNVs will generally have normalized coverage ratios that do not transparently correspond to the degree of copy number change.

Appendix: Window Boundary Definition

For the most part, windows are defined so that their chromosome coordinates are even multiples of the window length, so that for 2K windows, the chromosome positions of window boundaries will end with “x000”, where x is an even digit. Call the boundaries of these windows the “default boundaries”. Exceptions to these default boundaries will be windows at the ends of contigs. Windows will never span bases taken from more than one contig, even if the gap between contigs is small enough to permit this. Moreover, there will be special treatment of the bases outside the outermost full default windows for each contig. These “outside base” will either be added to the first full window towards the center of the contig or be placed in their own window, depending on whether the number of bases is larger than ½ the window width or not. For example, for a contig running from position 17891 to position 25336, and window width of 2000, use the following list of window intervals:

- (17891,20000), (20000,22000), (22000,24000), (24000,25336)

Note that the first 109 bases of the contig are added to the 2K interval immediately to their right, while the last 1336 bases are placed in their own window. A contig that is smaller than the window width (such as chrM for 100K windows) will be made into a single window that includes the entire contig. No windows will be reported for inter-contig gaps. To complete the illustration, suppose we have a chromosome consisting of three contigs as follows:

Contig ID	Begin Position	End Position
1	17891	25336
2	25836	29277
3	33634	34211

This would result in the following windows being used/reported; contig ID is shown only for clarity of presentation here.

- Contig 1: (17891,20000), (20000,22000), (22000,24000), (24000,25336)
- Contig 2: (25836,2800), (28000,29277)
- Contig 3: (33634,34211)

Consequences of this approach are:

- All non-gap bases of the genome are included in a window (and only one window).
- Windows are restricted to a single contig.
- Windows are between 0.5 and 1.5 times the nominal window width.
- Window boundaries are mostly round numbers, making it more obvious that segment boundaries correspond to window boundaries, with less chance of over-interpreting the precision of the CNV call boundaries.

Appendix: Score Computation

The CNV segmentation scores described in "Ploidy Inference, Segmentation, and Scoring" are more explicitly described here.

The probability of a given sequence *D=d*$_1$*,…,d*$_t$ of outputs of length *t* occurring as the result of a specific sequence of states *σ=s*$_1$*,…,s*$_t$ can be computed on a given HMM consisting of *n* states defined by initial state probabilities *P=p*$_1$*,…p*$_n$, transition probabilities $T = \{t_{ij}\}$ and emission probabilities $E = \{e_{ij}\}$ as follows:

$$Pr(D,\sigma|P,T,E) = p_{s_1} * e_{s_1,d_1} * \sum_{i=2}^{t} t_{s_{i-1},s_i} e_{s_i,d_i}$$

The probability of the data given the model is the sum over all possible sequences of states, that is for *S* the set of all possible sequences of states of length *t*:

$$Pr(D|P,T,E) = \sum_{\sigma \in S} Pr(D,\sigma|P,T,E)$$

This and other equations involving sums over subsets of *S* can be efficiently computed using the Forward/Backward algorithm. Application of Bayes' Rule allows us to determine the probability of a given path given the data and the model:

$$Pr(\sigma|P,T,E,D) = \frac{Pr(D,\sigma|P,T,E)}{Pr(D|P,T,E)}$$

From this, we can see that the most probable path, given the data and model, is the path which maximizes $Pr(D,\sigma|P,T,E)$. The path which maximizes this equation can efficiently be determined using the Viterbi algorithm.

However, we can also compute the probability of partial paths. For example, the probability that the path through the model that actually led to an observed sequence of data was in a particular state *q* at a particular time *u* can be computed as follows:

$$Pr(s_u = q|P,T,E,D) = \frac{Pr(D,s_u = q|P,T,E,)}{Pr(D|P,T,E)}$$

The denominator is discussed above, and the numerator can be obtained by summing the probability of the data and a particular path over all paths for which $s_u = q$, denoted $S_{s_u=q}$:

$$Pr(D,s_u = q|P,T,E) = \sum_{\sigma \in S_{s_u=q}}^{t} Pr(D,\sigma|P,T,E)$$

Thus:

$$Pr(s_u = q|P,T,E,D) = \frac{\sum_{\sigma \in S_{s_u=q}} Pr(D,\sigma|P,T,E)}{\sum_{\sigma \in S} Pr(D,\sigma|P,T,E)}$$

State assignment ("called ploidy") is done as follows; the state (ploidy) inferred at position u , $\widehat{s_u}$ is that state with maximal probability:

$$\widehat{s_u} = \mathrm{argmax}_q\, Pr(s_u = q|P, T, E, D)$$

(In case of a tie, choose arbitrarily.) The ploidyScore at position u, π_u is then:

$$\pi_u = -10 * \log_{10}(1 - Pr(s_u = \widehat{s_u}|P, T, E, D))$$

And the CNVTypeScore at position u, δ_u, is:

$$\delta_u = -10 * \log_{10}(1 - \sum_{q=a}^{b} Pr(s_u = q|P, T, E, D))$$

The bounds on the summation, a and b, are as follows. For a region expected to be diploid, if $\widehat{s_u}$<2, *a=0,b=1;* if $\widehat{s_u}$<2, *a=b=2*; if $\widehat{s_u}$>2, *a=3, b*=maximum ploidy (typically, 10). For a region expected to be haploid (male sex chromosomes), if $\widehat{s_u}$<1, *a=0,b=0;* if $\widehat{s_u}$=1, *a=b=1*; if $\widehat{s_u}$>1, *a=2, b*=maximum ploidy (typically, 10).

A segment is defined as a maximal run of like-ploidy positions. For a segment from position *l* to position *r*, we take the *ploidyScore* $\pi_{l,r}$, to be the mean of the ploidy scores for the constituent positions:

$$\pi_{l,r} = \frac{\sum_{u=l}^{r} \pi_u}{r - l + 1}$$

And similarly the *CNVTypeScore* of a segment, $\pi_{l,r}$, is the mean of the CNV type scores for the constituent positions:

$$\delta_{l,r} = \frac{\sum_{u=l}^{r} \delta_u}{r - l + 1}$$