Supplementary Materials for CLAMMS: a scalable algorithm for calling common and rare copy number variants from exome sequencing data

Jonathan S. Packer
Evan K. Maxwell
Colm O'Dushlaine
Alexander E. Lopez
Frederick E. Dewey
Rostislav Chernomorsky
Aris Baras
John D. Overton
Lukas Habegger
Jeffrey G. Reid

Introduction

In the accompanying manuscript, we have described the goals and methods of CLAMMS (Copy number estimation using Lattice-Aligned Mixture Models) and summarized the results of several validation experiments that assess its performance relative to existing tools. In the following supplementary notes, we provide a detailed explanation of the CLAMMS algorithm–including rationales for algorithmic design choices and default parameters–and we present the methods and full results of the validation experiments.

Contents

1 CLAMMS algorithm details

1.1 Calling Windows and Filters

CLAMMS divides exome capture regions that are $\geq$ 1000 bp long into equally-sized 500-1000 bp windows. This makes it possible to detect CNVs that partially overlap long exons. Examples of genes with extraordinarily long exons include *AHNAK*, *TTN*, and several Mucins.

CLAMMS filters windows with extreme 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, we find that stochastic coverage volatility increases dramatically, making it impossible to normalize effectively. We therefore filter windows where the GC-fraction is outside of a configurable range which defaults to [0.3, 0.7]. Supplementary Figure 1 illustrates why we chose these particular thresholds.

Benjamini and Speed (2012) found that "it is the GC content of the full DNA fragment, not only the sequenced read, that most influences fragment count." Accordingly, when computing GC-fractions, we symmetrically extend windows to be at least slightly longer than the average fragment size (another configurable parameter, which defaults to 200 bp).

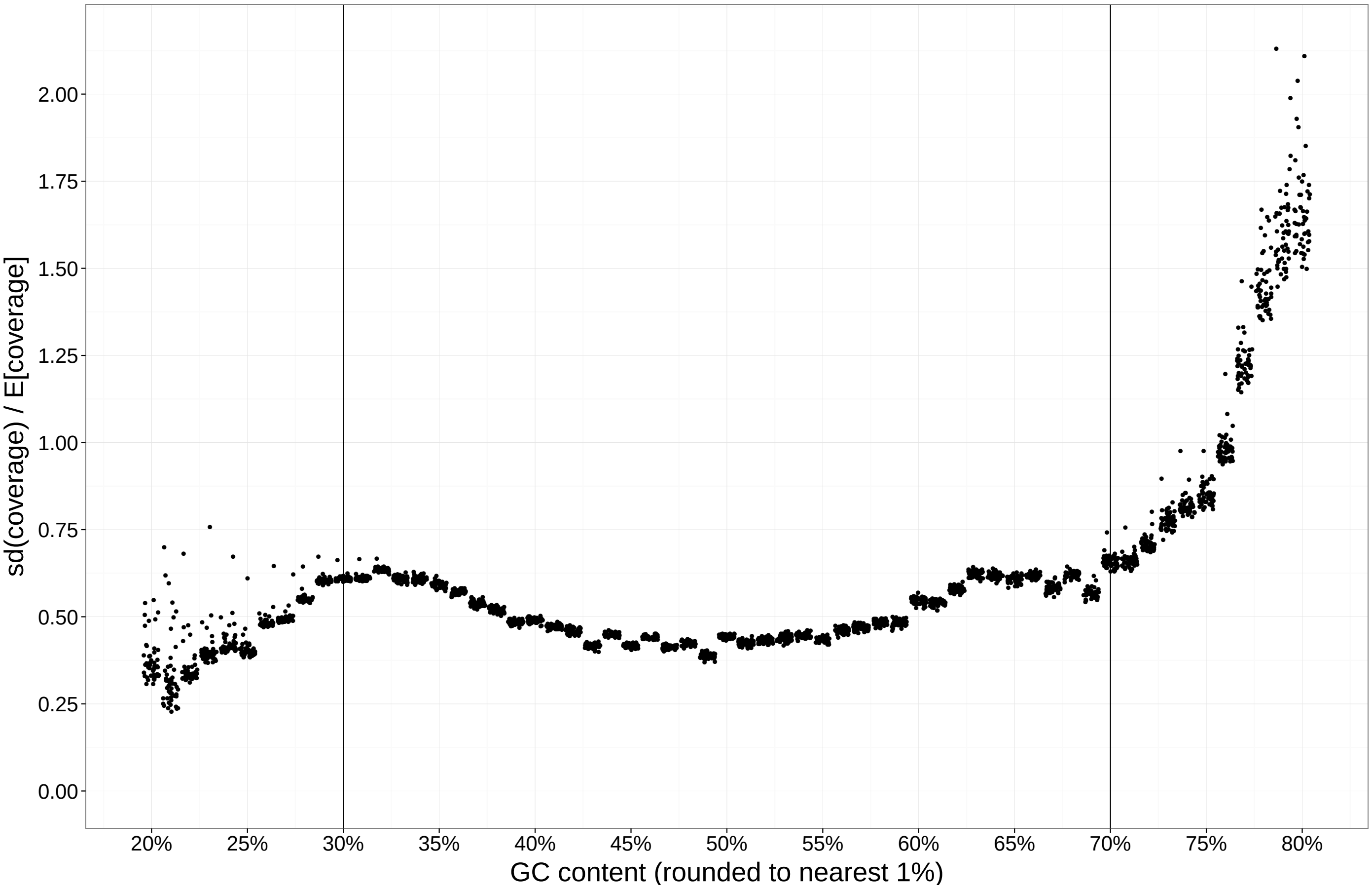
CLAMMS also filters windows where the mean mappability score for k-mers starting at each base in the window (default k=75) is < 0.75, windows in selected regions of extreme sequence polymorphism (where reads are difficult to map even if the reference sequence is unique), windows where the mean coverage across samples is $< 10\%$ of the expectation for windows with similar GC content, and windows where $> 1/3$ of 1000 Genomes Project samples have copy number > 2 (Handsaker *et al.*, 2015). In total, 12% of exome capture regions are excluded from the calling process.

In section 6 of this supplement, we evaluate CLAMMS CNV calls and calls made using four previous algorithms, which by default have much less stringent *a-priori* filters, against "gold-standard" calls from SNP genotyping arrays. CLAMMS achieves higher precision and equal recall for rare variants, suggesting that our filters are beneficial.

1.2 Within-Sample Normalization

The first step of CLAMMS is to normalize the coverage data for each individual sample to correct for GC-bias and overall average depth-of-coverage. This normalization step applies a simple formula: $Cov_{norm}(w) = Cov(w)\ /\ median(Cov \mid GC(w))$, where $median(Cov \mid GC(w))$ is the median coverage for the sample conditional on the GC fraction of window w.

The conditional median is computed by binning all windows for a sample by GC fraction (ex. [0.300, 0.310], [0.315, 0.325], etc.); computing the median coverage for each bin; and finally computing the normalizing factor for a given GC fraction by using a linear interpolation between the median coverage for the two bins nearest to it. While the binning resolution is configurable, we choose a default resolution that balances fine-grained binnings with the need to provide each bin with a sufficient sample size for estimation.

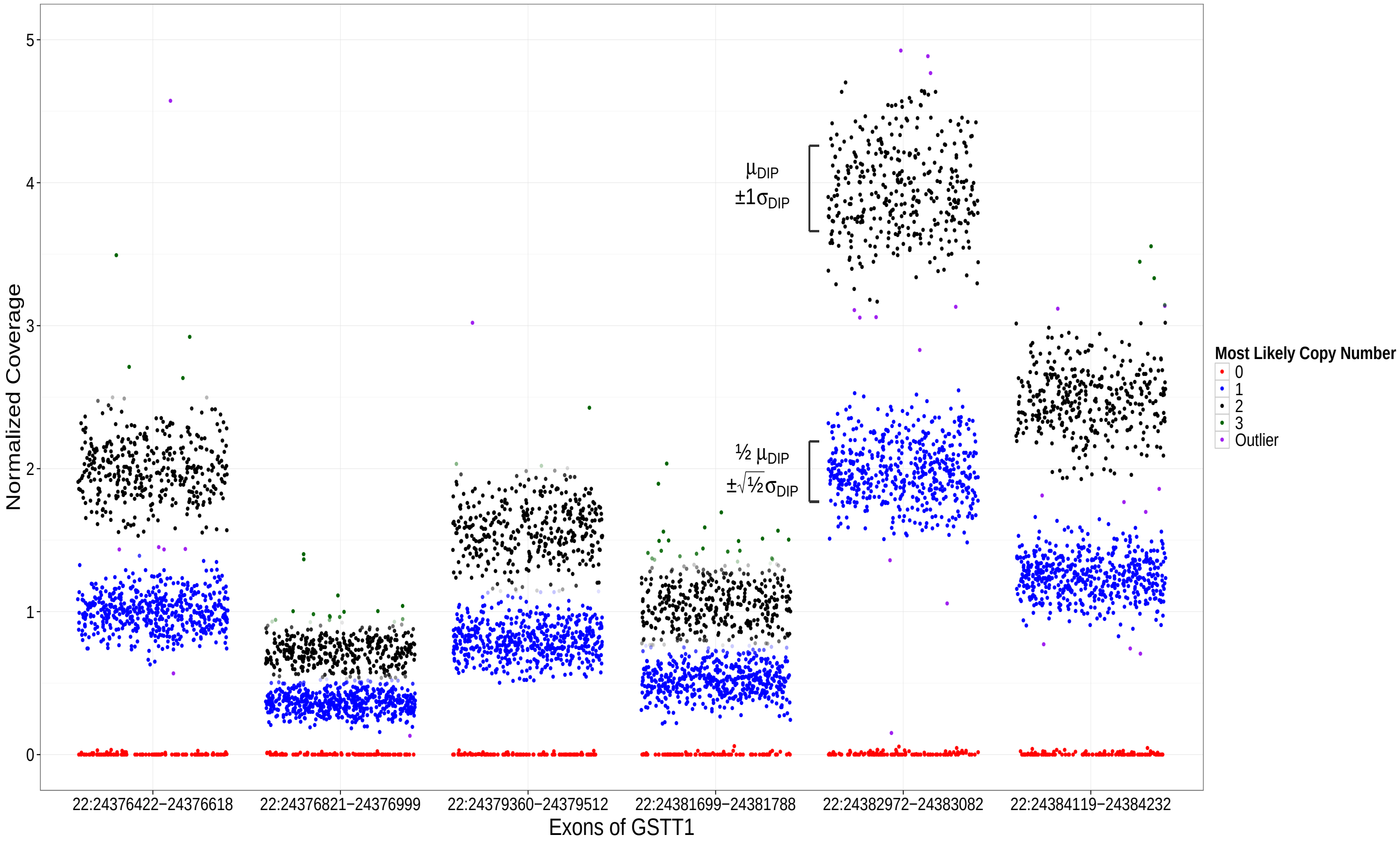


Supplementary Figure 1: shows the coefficient of variation (standard deviation / mean) of coverage, conditional on GC content, for 50 samples (points jittered for visibility). Above our default upper-limit of GC = 0.7, coverage variance becomes very high relative to the mean, making coverage-based CNV calls unreliable. Below our default lower-limit of GC = 0.3, the problem is more subtle: the variance of coverage itself is highly variable between samples. This makes it impossible 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.

1.3 Mixture Models

The second step of CLAMMS is to use mixture models to characterize the expected (normalized) coverage distribution at each calling window, conditional on copy number state. These mixture models are fit using the expectation-maximization algorithm (EM algorithm) with input data from a reference panel of samples. Each mixture model has components corresponding to at least copy numbers 0, 1, 2, and 3. Models for regions with known common duplications (duplication AF > 1% in 1000 Genomes data, Handsaker *et al.*, 2015) also include components for copy numbers 4, 5, and 6. The reason that we do not support copy number > 3 exome-wide is that we observed anomalous spikes in coverage to be prevalent in exome-sequencing data: for example, in a dataset of ˜3,000 samples, we observed more than

1,200 exons (excluding extreme-GC and low-mappability exons) which did not overlap any duplications in 1000 Genomes data but had > 0.5% of samples with coverage > 2x the estimated diploid mean.



Supplementary Figure 2: Mixture models fit to the observed coverage distributions for exons of the gene *GSTT1* (after within-sample normalization has been applied). Each point (jittered for visibility) shows a sample's normalized coverage for an exon, with color indicating 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. The mean coverage for a given copy number state differs significantly from exon to exon even after correcting for GC bias. The mixture model fitting procedure normalizes these additional non-GC-related coverage biases robustly, regardless of the frequency of non-diploid copy numbers in the region.

The components corresponding to non-zero copy numbers are defined to follow a Gaussian distribution. There are two free parameters related to these Gaussians: μ_{DIP} and σ_{DIP}, the mean and standard deviation of the mixture component corresponding to diploid copy number. For each non-diploid copy number k, the mean is constrained to equal $(k/2) * \mu_{DIP}$ (this is why we call the models "lattice-aligned" in the CLAMMS acronym). σ_{HAP}, the standard deviation for haploid samples, is set equal to $\sqrt{0.5} * \sigma_{DIP}$, as despite our Gaussian approximations, coverage conditional on a particular copy number is ideally Poisson-distributed with variance being equal to the mean. We considered the possibility of overdispersion, but an examination of the variance of male vs. female samples on chrX suggested that haploid samples did have approximately half the variance of diploid samples. For copy numbers > 2, coverage variance should theoretically be greater than for diploid samples, but we found that integrating this into the model increased the rate of false-positive duplications to an unacceptable extent. Therefore, we set the standard deviation parameters for components corresponding to copy numbers > 2 to simply be equal to σ_{DIP}. The constraints imposed on the parameters of the non-diploid components help the model avoid overfitting the training data.

Homozygous deletions (copy number 0) theoretically should show zero coverage, but mismapped reads can give a small level of coverage even in truly deleted regions. We therefore define the component for copy number 0 as an exponential distribution with mean $(1/\lambda)$ initially equal to 6.25% of μ_{DIP}. The mean of this component is constrained to be no greater than this initial value. If there are no mismapping issues with the region, iterations of the EM algorithm will drive the mean to 0 ($\lambda \to \infty$). To address this blow-up, if the mean drops below 0.1% of μ_{DIP}, we replace the exponential distribution with a point mass at 0.

In summary, the mixture model has 4 parameters: μ_{DIP} and σ_{DIP}; λ, the rate of the exponential component (copy number 0), and lastly a flag indicating if the exponential has been replaced by a point mass. The model is fit using a maximum of 30 iterations of the EM algorithm. A heuristic is used to detect early convergence. As EM is a local optimization procedure, we estimate the initial values of μ_{DIP} and σ_{DIP} robustly to decrease the chance that EM converges to a non-global optimum. μ_{DIP} is initialized as the median coverage across all samples for the region in question (in regions where the median sample is haploid, we observe that the EM iterations do eventually reach the proper diploid mean). σ_{DIP} is initialized to the median absolute deviation (MAD) of the coverage values around their median, scaled by a constant factor to achieve asymptotic normality (*c.f.* the mad function in R).

Samples that have low likelihoods for all considered copy number states ($> 2.5\sigma$ from the mean) are flagged as outliers for purposes of model-fitting. If a region has outlier samples, the mixture model is retrained with the outlier coverage values removed.

1.4 Hidden Markov Model

The third and final step of CLAMMS is to call CNVs using a Hidden Markov Model. The input to the HMM is the normalized coverage values (from the within-sample procedure described previously) for an individual sample at each calling window. The states of the HMM are DEL (deletion), DIP (diploid), and DUP (duplication). A specific integer copy number is assigned to a DEL or DUP call in a post-processing step based on mixture model likelihoods.

The transition probabilities based on those used in XHMM (Fromer *et al.*, 2012), except with the parameter $1/q$, the mean of the prior geometric distribution of # windows in a CNV, set to 0 ($q = \infty$). The effect of this is that the HMM places no prior on the # of windows in a CNV, instead only using the exponentially-distributed attenuation factor which is based on actual genomic distance. Therefore, the only two prior assumptions are that 1) DEL and DUP are equally likely, and 2) the size of CNVs is exponentially distributed.

The emission probabilities are derived from the mixture models. The probability of observing a (normalized) coverage value x, at a calling window w, given HMM state s, is determined by the components of the mixture model trained at w that correspond to state s. Components 0 and 1 correspond to the DEL state; components 3-6 correspond to the DUP state. A likelihood-weighted average of the probabilities for each relevant copy number is used, e.g. if for a given calling window, $L(CN = 1 \,|\, cov) = 9 * L(CN = 0 \,|\, cov)$, then the emission probability for the DEL state is $0.9 * P(cov \,|\, CN = 1) + 0.1 * P(cov \,|\, CN = 0)$.

Using this Hidden Markov Model, we identify CNVs as regions where the maximum-likelihood sequence of states, predicted by the Viterbi algorithm, is non-diploid. Running the Viterbi algorithm in only one direction introduces a directional bias to the CNV calls: there is effectively a high cost to "open" a CNV but a low cost of "extending" it, so the called CNV regions will tend to overshoot the trailing breakpoint. We therefore 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.

For each discovered CNV, five quality metrics are 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.

Even with the *a priori* filtering of windows with GC-content outside of the range [0.3, 0.7], we still find high rates of stochastic sequencing artifacts at the extreme ends of this range. We therefore modify the Viterbi and Forward-Backward algorithms to place less credence on windows with "moderately-extreme" GC-content without ignoring them entirely. This effect is 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 effectively reduces the relative significance of the data (observed coverage) at this window compared to the prior (encoded by the state transition probabilities). For GC-fraction f in the default *a priori* valid range of [0.3, 0.7], the window weight is set equal to $(1 - (5 * abs(f - 0.5))^{18})^{18}$. The high polynomial term makes the curve flat for non-extreme GC (ex. weight = 0.99993 for $f = 0.4$), but drop sharply at the edges of the valid GC range (ex. weight = 0.5 for $f = 0.3333$).

1.5 Sex Chromosomes

CLAMMS can fit models and make calls for regions on the sex chromosomes if it is given the sex of each input sample. Basing the expected copy number (diploid or haploid) on sex explicitly is more effective than normalizing the variance due to sex (XHMM, CoNIFER) or comparing samples to highly-correlated samples (ExomeDepth, CANOES) because it accounts for the integer nature of copy number states. 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 is not, because one cannot have a copy number of 1/2.

2 Batch effects and pipeline implementation details

Systematic coverage biases that arise due to variability in sequencing conditions are commonly referred to as "batch effects." Previous algorithms have used two strategies for addressing these biases. CoNIFER (Krumm *et al.*, 2012) and XHMM (Fromer *et al.*, 2012) compute the principal components of the sample-by-exon coverage matrix and remove the contributions of the largest few components. ExomeDepth (Plagnol *et al.*, 2012), and CANOES (Backenroth *et al.*, 2014) normalize each sample's coverage values against the average in a "custom" reference panel of samples which have coverage profiles that are highly-correlated to the individual sample in question. Both normalization strategies require that a group of samples be processed together and are therefore difficult to integrate into an automated variant-calling pipeline. They also require that each sample's coverage profile be compared to the coverage profile of every other sample, resulting in quadratic-time computational complexity.

CLAMMS uses the "custom reference panel" approach to correct for batch effects, but instead of comparing samples based on their coverage profiles–a high-dimensional space–it considers a low-dimensional metric space consisting of seven sequencing quality control metrics from Picard (http://broadinstitute.github.io/picard). Working in this low-dimensional space allows for improved scalability compared to previous algorithms: samples can be indexed ahead-of-time using a k-d tree structure that allows for fast nearest-neighbor queries and uses a minimal amount of RAM.

Our variant-calling pipeline works as follows:

1. Query our 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] (scaled value = [raw value - min] / [max - min])

3. In parallel, for each sample:

   (a) Compute depth-of-coverage from the BAM file using samtools (Li *et al.*, 2009) 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.

Sample code demonstrating how to run the pipeline is provided at the CLAMMS Github repository. Larger values of k decrease variance in the statistical inference of the mixture model parameters but increase bias. We chose the default value $k = 100$ as it seemed to have the best bias-variance trade-off. The pipeline can be extended to run in an online manner if the k-d tree is stored in a database (though we have not implemented this feature yet).

For small-scale studies, CLAMMS can also be used without having to compute Picard QC metrics if one manually assigns samples to batches based on a PCA plot of the sample-by-exon coverage matrix (an example is provided in the CLAMMS tutorial). A separate set of CLAMMS models should be trained for each batch and used to call CNVs for samples in that batch.

3 Effects of population diversity on coverage profiles

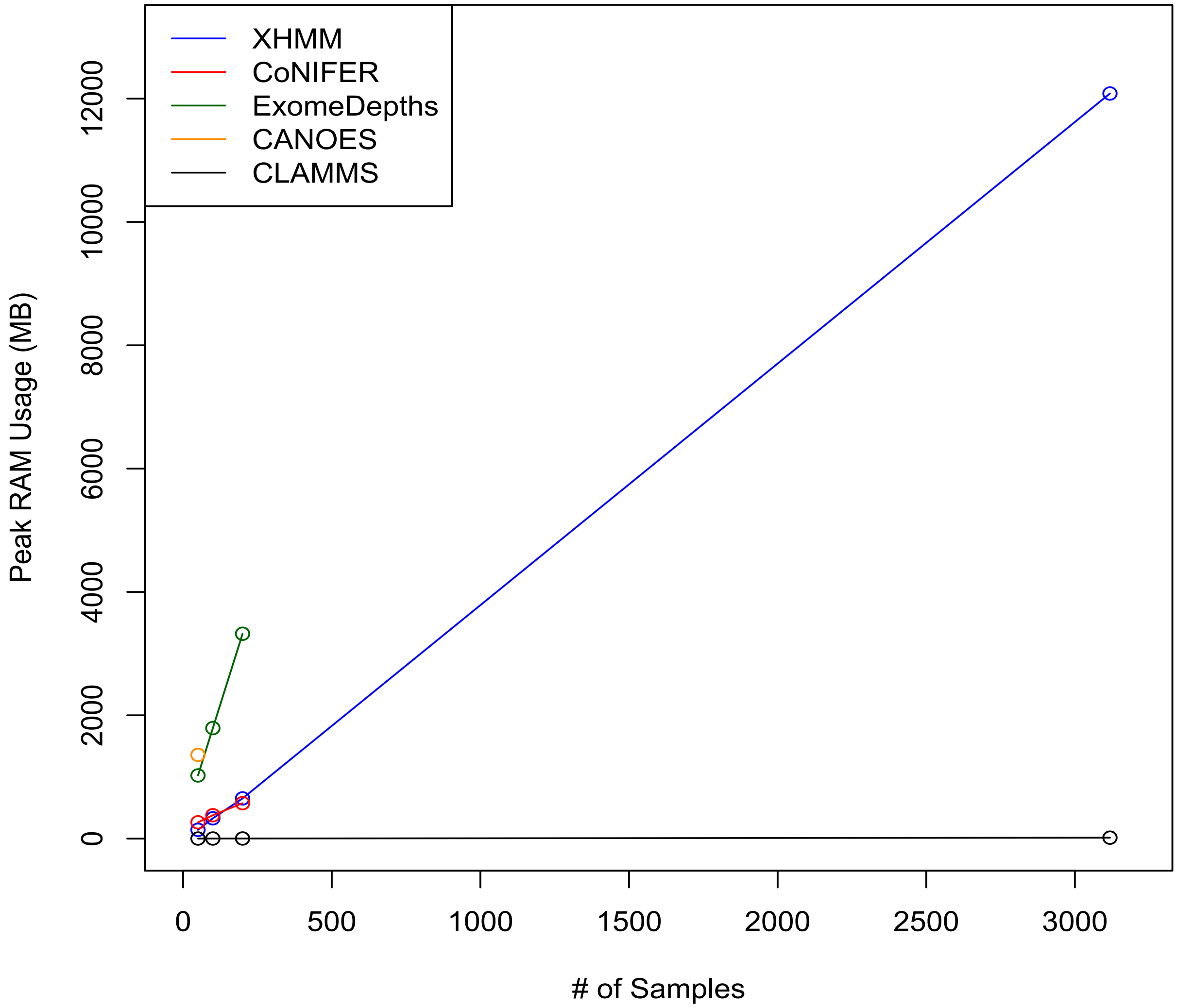
Variation in sample quality, sample preparation procedures, and sequencing procedures is reflected in the Picard metrics that CLAMMS uses to correct batch effects. Population diversity and admixture are not modeled. However, given modern sequencing and read mapping technologies, it is highly unlikely that population diversity will have a significant effect on coverage profiles outside of copy number polymorphic regions. While we do not have data from diverse populations to prove this claim, we will discuss theoretical considerations in this section.

The two largest factors determining the sequencing depth of coverage for an exon are 1) the GC content of sequencing DNA fragments which include the exon, and 2) the quality of read mapping to the exon. In an individual, sequence mutations (SNPs and indels) affect only a small proportion of the exome, reflecting evolutionary sequence conservation. Any given ˜150 bp DNA fragment is therefore unlikely to include a large enough number of common variants to have a significant effect on its GC content. Sequence mutation could have a greater effect on coverage by causing reads to be mismapped. But if one uses modern read lengths (75 bp or greater) and the GATK IndelRealigner, common variants are unlikely to cause significant problems with mappability–except in segmental duplications (which are filtered by the CLAMMS mappability filter) and regions of extreme sequence polymorphism (CLAMMS blacklists selected loci such as the HLA and KIR gene clusters).

CLAMMS should therefore be suitable to use with non-homogenous populations. Additionally, other algorithms with PCA-based batch effect correction may suppress signal from copy number polymorphisms if sample preparation and sequencing procedures are consistent enough that common CNVs are represented in the top few principal components of coverage. CLAMMS' Picard-based batch effect correction avoids this problem.

4 Computational performance

If n is the number of samples to be processed and k is the size of the reference panel selected for each sample, CLAMMS takes O(kn log n) time to call CNVs for each sample, an improvement over the O(n^2) time complexity of previous algorithms. In practice, samples take ˜45 seconds to process on an Amazon Web Services m3.xlarge server (excluding the time required to compute depth-of-coverage and the sequencing QC metrics from the BAM file, as we compute these anyway for other purposes besides CNV calling). Once reference panels are selected for each sample (which takes only a few seconds even for tens of thousands of samples), each "mini-batch" of sample plus reference panel can be processed in parallel, both across servers and across multiple processes on each individual server. Since each CLAMMS process only processes k samples at a time, RAM usage is minimal (˜50 MB/process if $k = 100$). Supplementary figure 3 compares the RAM usage of CLAMMS–O(k)–vs. other algorithms–O(n).



Supplementary Figure 3: RAM usage of CNV-calling algorithms for 50 samples (all algorithms); 100 and 200 samples (all but CANOES, which we stopped after running for 4 hours without finishing); and 3164 samples (CLAMMS, XHMM).

5 Validation using CEPH pedigree 1463

Our 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 (a subset of the Centre de'Etude du Polymorphism Humain / Utah pedigree 1463, including grandparents NA12889, NA12890, NA12891, NA12892; parents NA12877, NA12878; and children NA12880, NA12882). Low transmission rates or an excess of putative *de novo* variants result from either false positives or false negatives. Each of the 8 pedigree members was sequenced in three technical replicates. We made CNV calls using each algorithm's default parameters as described in their respective tutorials. A reference panel of 92 unrelated samples was made available to each algorithm. To ensure a fair comparison, we applied the *a-priori* filters used by CLAMMS (i.e. filtering extreme-GC and low-mappability regions) to the input data for all algorithms, so differences in performance cannot be attributed to CLAMMS' exclusion of the most problematic genomic regions. We also exclude sex chromosomes from the comparison.

We computed three metrics 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. We used a 50% overlap criterion when determining whether a call is transmitted/inherited (i.e. a CNV in a child is inherited if any CNV in its parents overlaps at least 50% of it).

Supplementary Table 1: Performance metrics for CNV calls on the CEPH pedigree

	# Calls			**Call Statistics (%)**		
Algorithm	**Total**	**Common**	**Rare**	**Consistent**	**Transmitted**	**Inherited**
CLAMMS	323	276	47	91.7	61.9	95.0
XHMM	94	35	59	41.5	9.5	22.1
CoNIFER	37	12	25	68.3	34.8	72.2
CANOES	29	18	11	93.3	0.0	0.0
ExomeDepth	659	419	240	65.3	32.3	61.8

# Calls is for the 8 pedigree members across 3 technical replicates (24 samples in total). CNVs are classified as common if their allele frequency in Handsaker *et al.*, 2015 or Coe *et al.*, 2014 is $\geq 1\%$, and classified as rare otherwise (note that "rare" CNVs may be false-positives). We exclude ExomeDepth calls with Bayes Factor < 10 (the authors do not recommend any particular threshold).

Supplementary Table 1 shows the number of calls made by each algorithm, their consistency across technical replicates, and their adherence to Mendelian inheritance patterns. All of the algorithms except CLAMMS are focused exclusively on rare variants, assuming that reference panel samples are diploid (presenting a unimodal coverage distribution) at all loci. Their poor performance is therefore to be expected, as by definition, most CNVs in the pedigree are common variants. CLAMMS on the other hand performs very well for genotyping deletions in the pedigree (98.6% of calls are inherited) and reasonably well for genotyping duplications (83% of calls are inherited). The higher-than-Mendelian computed transmission rate (62%) is due to false negatives in parents. The CNV calls of each algorithm for the pedigree are available CLAMMS Github repository (https://github.com/jspacker/clamms).

6 Validation using "gold-standard" array-based CNV calls

Our second validation experiment was to compare CNV calls from CLAMMS, XHMM, CoNIFER, CANOES, and ExomeDepth to "gold-standard" calls from PennCNV (a CNV-caller by Wang *et al.*, 2007, that uses data from SNP genotyping arrays) for a set of 3164 samples in the Regeneron Genetics Center's human exome variant database. We excluded from the test set samples for which:

- # PennCNV calls > 50
- LRR_SD ("standard deviation of log R ratio") > 0.23 (95th percentile)
- BAF_drift ("B-allele frequency drift") >0.005 (95th percentile)

Array-based CNV calls, despite generally being more accurate than CNV calls from exome sequencing read depth, are not a true "gold-standard" and include false positives, including several putative copy number polymorphic loci (AF > 1%) that did not overlap any variants in two published datasets (CNV calls from 849 whole genomes by Handsaker *et al.*, 2015, and array-based CNV calls from 19,584 controls in an autism study by Coe *et al.*, 2014). PennCNV is also not designed to genotype common CNVs. To minimize the false positive rate in the test set, we only included CNVs that were rare and not small. We specifically exclude PennCNV calls for which:

- CNV length < 10 kb or > 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 (GRCh37)
  or a common genomic rearrangement in HapMap
- allele frequency > 0.1% in Handsaker *et al.*, 2015, Coe *et al.*, 2014, or the 3,164 test samples (CNVs are included in the allele frequency count if they overlap at least 33.3% of the CNV in question)

The final test set after all filters have been applied includes 1,715 CNVs (46% DEL, 54% DUP) in 1,240 samples. For this evaluation, each algorithm was run with default parameters and procedures as described in their respective tutorials. The CLAMMS tutorial recommends considering samples with >2x the median # of calls for any particular dataset to be outliers. For this dataset, the median # of CLAMMS calls/sample is 14, so we exclude CLAMMS calls from 26 samples (0.8% of the total) where it makes >28 calls. Array calls from these samples are still included in the test set.

CoNIFER fails if the number of exome capture regions on any chromosome is less than the number of samples. To get it to work, we had to exclude chromosomes 18 and 21 from its input data. CANOES ran out of memory on a server with 30 GB RAM available, so we had to exclude it from the comparison. Testing CANOES on a smaller set of 200 samples, it was very slow, taking over 8 minutes per sample to call CNVs. ExomeDepth makes a very large number (˜200) CNV calls per sample and does not provide specific guidelines for how to filter them. We filtered ExomeDepth calls with Bayes Factor < 10.

Supplementary Table 2: CNV calls from four algorithms compared to PennCNV “gold-standard”

Metric	**Algorithm**	**Any Overlap**	**33% Overlap**	**50% Overlap**
Precision	CLAMMS	78.4	71.9	67.2
	XHMM (Q30)	66.4	60.2	55.4
	XHMM (Q60)	71.2	64.5	59.2
	CoNIFER	21.6	11.9	7.6
	CANOES	NA	NA	NA
	ExomeDepth	57.1	53.0	49.3
Recall	CLAMMS	65.4	49.7	41.9
	XHMM (Q30)	64.1	51.7	44.3
	XHMM (Q60)	59.9	49.7	42.5
	CoNIFER	70.9	70.7	70.4
	CANOES	NA	NA	NA
	ExomeDepth	80.9	57.8	49.2
F-score	CLAMMS	71.3	58.8	51.6
	XHMM (Q30)	65.2	55.6	49.2
	XHMM (Q60)	65.1	56.1	49.5
	CoNIFER	33.1	20.4	13.7
	CANOES	NA	NA	NA
	ExomeDepth	66.9	55.3	49.2

We calculate precision as the % of exome-based calls that could possibly be supported by a PennCNV call–meaning that they are subject to the same filtering criteria–that are in fact overlapped by a PennCNV call at the specified overlap threshold. We calculate recall (sensitivity) as the % of PennCNV calls that are overlapped by any exome-based call (no filters applied) at the specified overlap threshold. F-score is defined as the harmonic mean of precision and recall.

As mentioned on the previous page, CANOES is unable to process a dataset of this size on a server with 30 GB RAM, so it is excluded from the comparison. CoNIFER achieves high recall, but at the cost of unusably-low precision.

CLAMMS achieves a 9.3% higher F-score than XHMM and a 6.6% higher F-score than ExomeDepth using the any-overlap criterion. Using the stricter 50%-overlap criterion, CLAMMS achieves a 4.9% higher F-score than both XHMM and ExomeDepth. This improvement is driven by CLAMMS’ higher precision (18-20% higher than XHMM and 36-37% higher than ExomeDepth depending on the overlap threshold).

While CLAMMS’ default parameters favor precision over recall (preferable for population-level analyses), it can also be configured to increase recall at the cost of precision by increasing a parameter –cnv_rate that determines the transition probabilities of the Hidden Markov Model. This may preferable for analyses of Mendelian disease pedigrees.

7 Validation using TaqMan qPCR

We used TaqMan quantitative-PCR to validate a selection of CNV loci (20 rare, 23 common) predicted by CLAMMS. For each locus, we compared the PCR-based copy number predictions to CLAMMS CNV genotypes for 56 / 165 samples for rare and common loci respectively. The CNV loci were selected randomly from the set of all loci that overlapped at least one gene associated with disease in the Human Gene Mutation Database (Stenson *et al.*, 2012; 7430 genes total; disease associations are from all mutation types, not just known CNVs).

19/20 (95%) of the rare variants were validated.

4/23 common variant loci were plausibly correct, but had high variance in the PCR data, making the results ambiguous (statistics for these loci are listed as "NA" in Supplementary Table 4). An additional 2/23 had ambiguous genotypes for a few (˜5) borderline samples, but were clear for the rest (statistics for these loci are marked with "˜" in Supplementary Table 4).

16/17 (94%) of the unambiguous common variant loci had no false positives and one locus had 5/6 calls correct. 14/17 (82%) had ≥ 90% sensitivity (including 9/17 with 100% sensitivity); the other 3/17 had sensitivities of 88.0%, 87.5%, and 54.7%.

The means of the precision/sensitivity values for the 17 unambiguous loci plus the two mostly-clear loci were 99.0% and 94.0% respectively. Plots of the PCR results are available at the CLAMMS Github repository. Supplementary Figures 4-9 show examples of these plots.

Supplementary Table 3: Rare CNV TaqMan Validations

CNV	**Size**	**Type**	**Gene**†	**Validated?**
chr1:230371759-230415205	43,446	DUP	*GALNT2*	YES
chr10:113913307-114136207	222,900	DUP	*GPAM*	YES
chr11:104815479-105009807	194,328	DUP	*CARD16*	YES
chr12:21007961-21392124	384,163	DEL	*SLCO1B1*	YES
chr13:114514707-114538608	23,901	DUP	*GAS6*	YES
chr14:74753163-74991929	238,766	DUP	*NPC2*	YES
chr15:85147158-85681135	533,977	DEL	*NMB*	YES
chr16:21152620-21289573	136,953	DUP	*CRYM*	YES
chr17:12798256-12920439	122,183	DUP	*ELAC2*	YES
chr18:2544651-2707644	162,993	DUP	*SMCHD1*	YES
chr18:64176231-64239442	63,211	DEL	*CDH19*	YES
chr19:52271911-52588055	316,144	DUP	*FPR3*	YES
chr2:55910919-55920959	10,040	DUP	*PNPT1*	YES
chr3:124390506-124492760	102,254	DUP	*UMPS*	YES
chr4:10560030-10567775	26,550	DEL	*CLNK*	YES
chr7:82595086-82595804	718	DEL	*PCLO*	NO
chr7:121738503-121773781	35,278	DEL	*AASS*	YES
chrX:53560269-53622364	62,095	DUP	*HUWE1*	YES
chrX:105137824-105571052	433,228	DUP	*SERPINA7*	YES
chrX:120181538-120183935	2397	DEL	*GLUD2*	YES

† gene that led CNV to be selected (because of a disease association in HGMD)

Supplementary Table 4: Common CNV TaqMan Validations

CNV Locus	**Size**	**Type**	**Gene**†	**# True CNV**‡	**False +**	**False -**
chr1:1634914-1663963	29,049	Both	*CDK11A*	27	0	1
chr1:16370987-16390132	19,145	DUP	*CLCNKB*	NA*	NA	NA
chr1:152573207-152586575	13,368	DEL	*LCE3B*	148	0	67
chr1:206317576-206331229	13,653	DEL	*CTSE*	10	0	0
chr10:27687222-27703180	15,958	DEL	*PTCHD3*	10	0	0
chr10:135340899-135379034	38,135	DUP	*SYCE1*	13	0	1
chr11:8959162-8959721	559	DEL	*ASCL3*	8	0	1
chr11:134151918-134214350	62,432	Both	*GLB1L3*	NA*	NA	NA
chr16:55844428-55866968	22,540	Both	*CES1*	˜49	˜0	˜5
chr19:46623586-46627907	4,321	DEL	*IGFL3*	10	0	1
chr19:54801926-54804222	2,296	DEL	*LILRA3*	69	0	0
chr2:110881367-110962546	81,179	Both	*NPHP1*	7	0	0
chr21:37510122-37618976	108,854	DUP	*CBR3*	8	0	0
chr22:24373137-24384232	11,095	DEL	*GSTT1*	108	0	13
chr22:42523843-42526794	2,951	Both	*CYP2D6*	NA*	NA	NA
chr2:241627221-241710522	83,301	DUP	*KIF1A*	6	0	0
chr3:151531950-151545961	14,011	DEL	*AADAC*	11	0	1
chr4:3446037-3478270	32,233	DEL	*HGFAC*	6	0	0
chr4:69403342-69434203	30,861	DEL	*UGT2B17*	100	0	1
chr5:70307101-70308743	1,642	Both	*NAIP*	˜42	˜1	˜1
chr5:138651764-138658657	6,893	DUP	*MATR3*	NA*	NA	NA
chr7:142829209-142881529	52,320	DEL	*PIP*	9	0	0
chr9:215201-464220	249,019	DUP	*DOCK8*	5	1	0

* The variance of the TaqMan data for these loci was too high to determine copy number state

† gene that led CNV to be selected (because of a disease association in HGMD)

‡ Using the copy number predicted by TaqMan as the ground-truth.
This is not representative of CNV allele frequency.
The 165 samples genotyped for common CNV loci were not randomly selected:
we attempted to minimize the number of samples required to ensure that each locus
had a reasonable number of samples with non-diploid copy number
(which is why several loci in the table have exactly 10 predicted CNV).

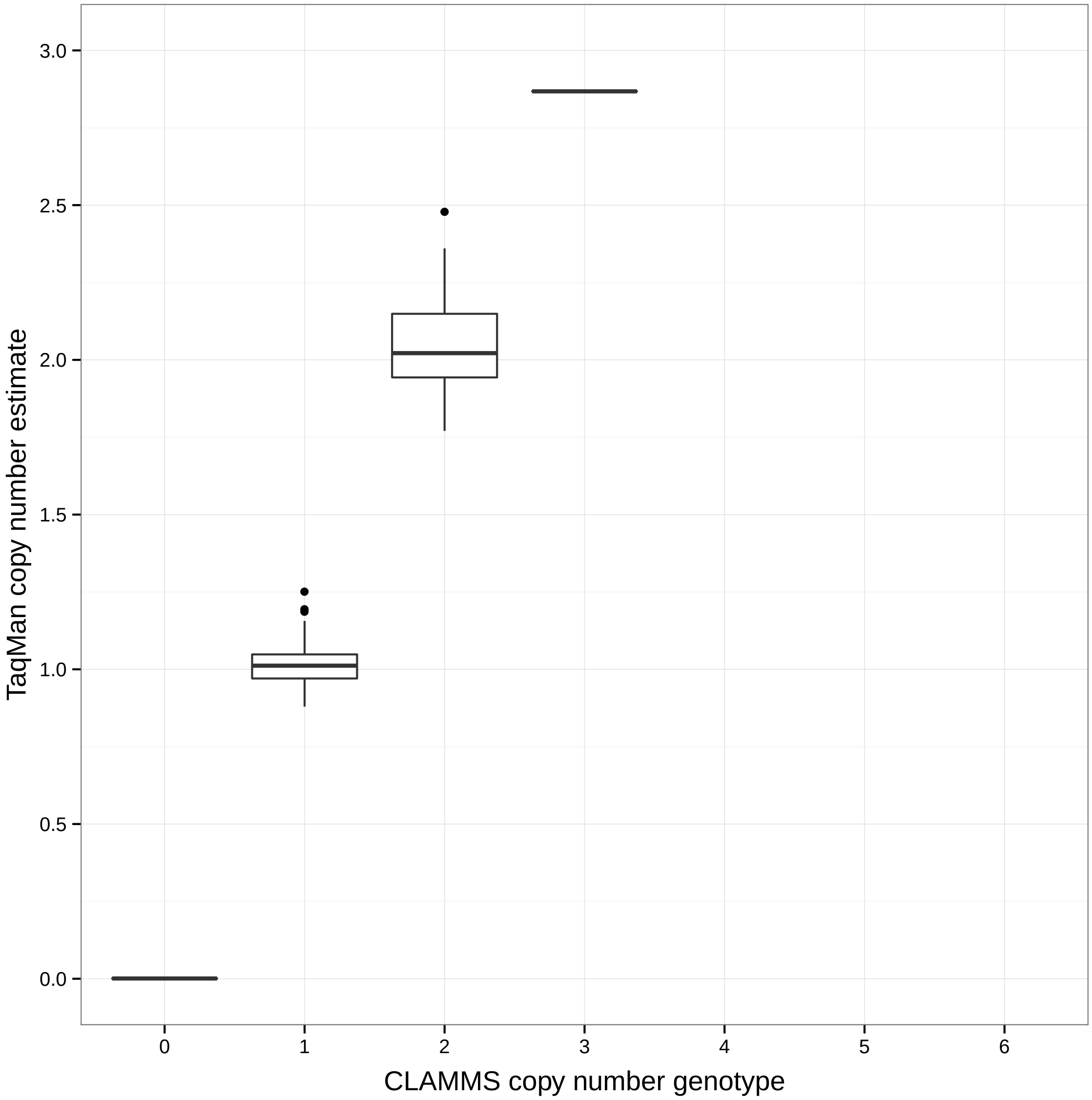
Supplementary Table 5: False positive and false negative rates for CLAMMS and four other algorithms at TaqMan-validated common CNV loci

Type	**Gene**	**# True CNV**	**False +**					**False -**				
			CL	**ExD**	**CAN**	**XH**	**CoN**	**CL**	**ExD**	**CAN**	**XH**	**CoN**
Both	*CDK11A*	27	0	0	0	0	0	1	0	8	1	13
DUP	*CLCNKB*	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
DEL	*LCE3B*	148	0	43	0	0	0	67	0	148	148	148
DEL	*CTSE*	10	0	1	0	0	0	0	0	0	0	0
DEL	*PTCHD3*	10	0	0	0	0	0	0	0	3	1	0
DUP	*SYCE1*	13	0	0	0	0	0	1	1	1	1	1
DEL	*ASCL3*	8	0	0	0	0	0	1	1	3	8	8
Both	*GLB1L3*	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
Both	***CES1***	~49	~0	~0	~0	~0	~0	**~5**	**~20**	~38	~26	~36
DEL	*IGFL3*	10	0	0	0	0	0	1	1	5	6	5
DEL	***LILRA3***	69	**0**	**56**	0	0	0	**0**	**24**	55	55	55
Both	*NPHP1*	7	0	0	0	0	0	0	0	0	0	0
DUP	*CBR3*	8	0	0	0	0	0	0	0	0	0	0
DEL	*GSTT1*	108	0	8	0	1	2	13	1	73	103	101
Both	*CYP2D6*	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
DUP	*KIF1A*	6	0	0	0	0	0	0	0	0	0	0
DEL	*AADAC*	11	0	0	0	0	0	1	1	4	1	1
DEL	*HGFAC*	6	0	1	0	0	0	0	0	1	2	2
DEL	***UGT2B17***	100	**0**	**18**	0	0	1	**1**	**7**	72	100	99
Both	***NAIP***	~42	**~1**	**~87**	~4	~0	~3	~1	~0	~38	~41	~19
DUP	*MATR3*	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
DEL	*PIP*	9	0	0	0	0	0	0	0	0	0	0
DUP	*DOCK8*	5	1	1	1	1	1	0	0	0	0	0

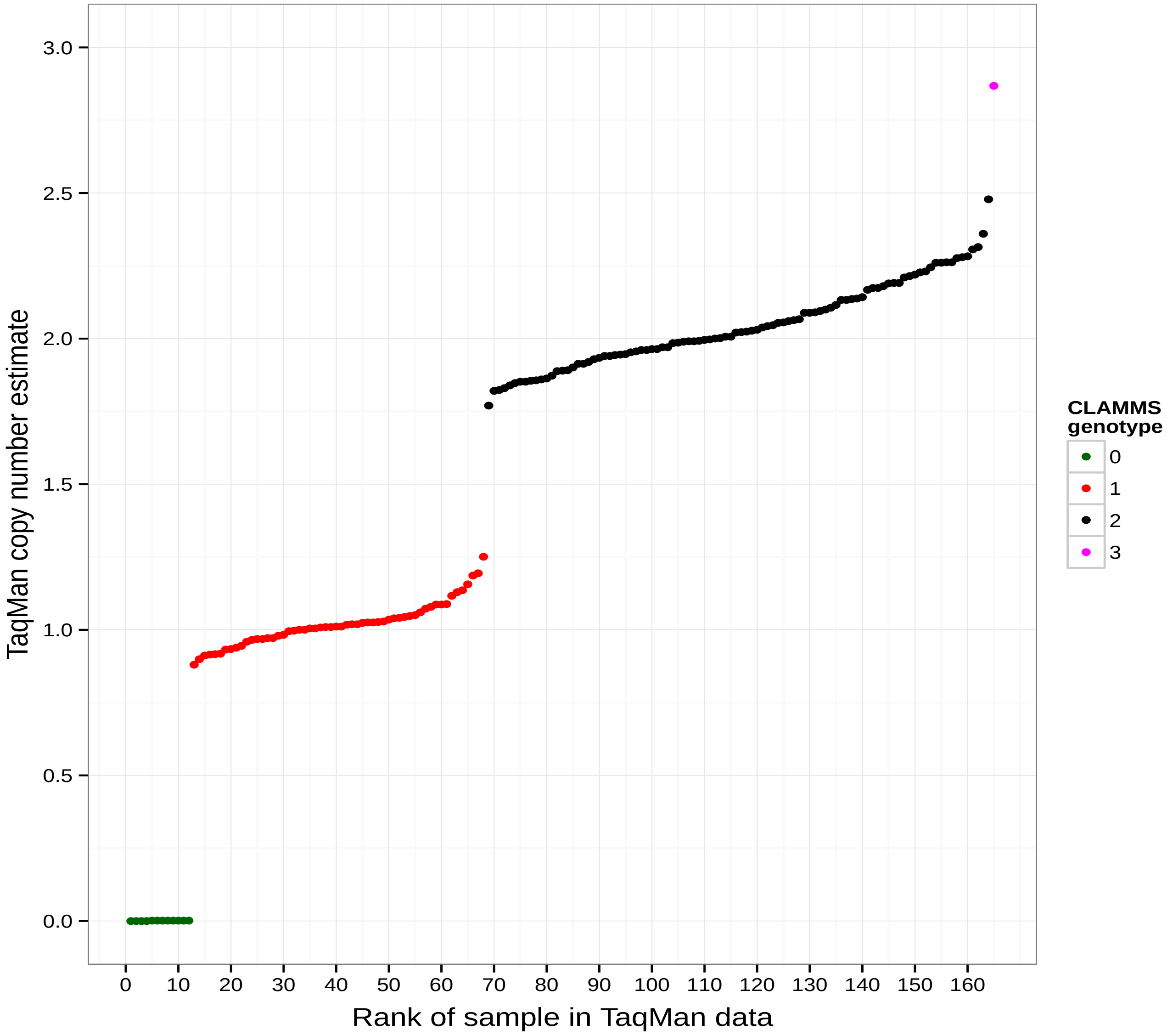
CL = CLAMMS, **ExD** = ExomeDepth, **CAN** = CANOES, **XH** = XHMM, **CoN** = CoNIFER
"NA" values indicate loci that had too much variance in the TaqMan data to assess copy number.

We called CNVs for the 165 samples used in the TaqMan common CNV validations using ExomeDepth, CANOES, XHMM, and CoNIFER, and compared their genotyping accuracy to that of CLAMMS. CANOES, XHMM, and CoNIFER frequently have false negatives at mid-frequency loci and are almost completely insensitive to very common variants.

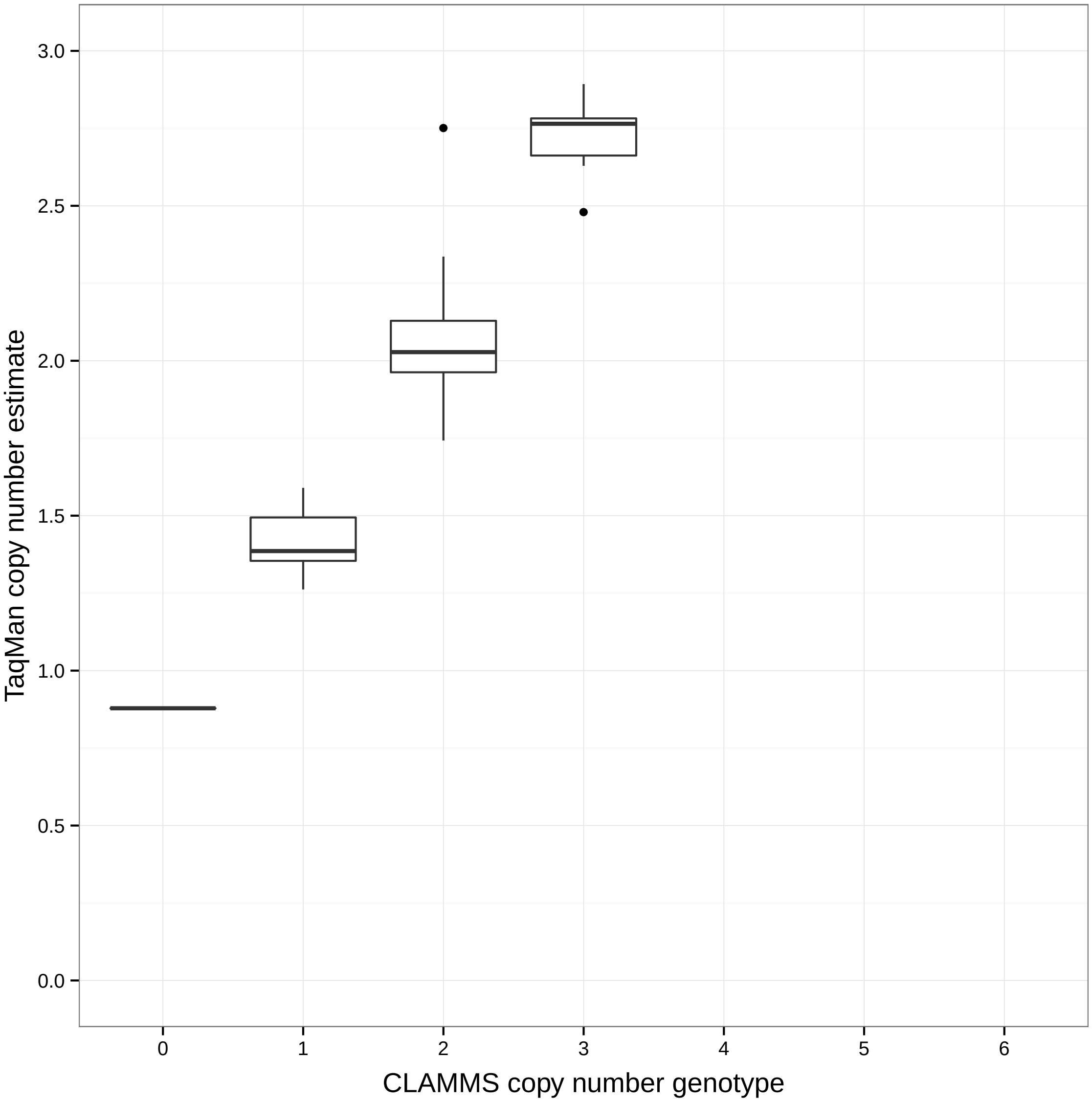
Both CLAMMS and ExomeDepth have near-perfect genotyping accuracy at mid-frequency loci. ExomeDepth genotypes are unreliable however for very common variants, e.g. at the *CES1*, *LILRA3*, *UGT2B17*, and *NAIP* loci (highlighted in red).



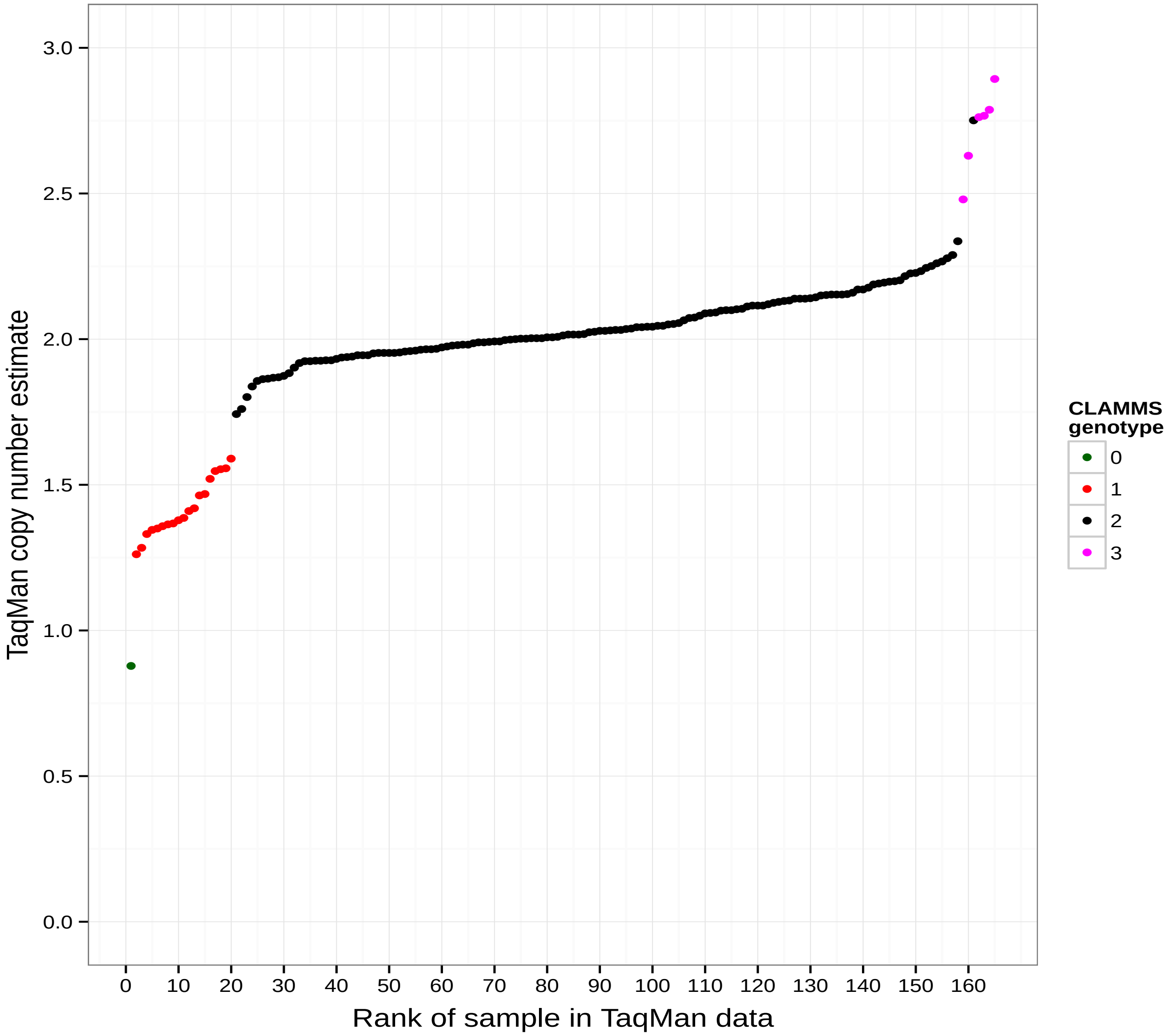
Supplementary Figure 4: **Comparison of CLAMMS and TaqMan copy number predictions for the LILRA3 common variant locus.**



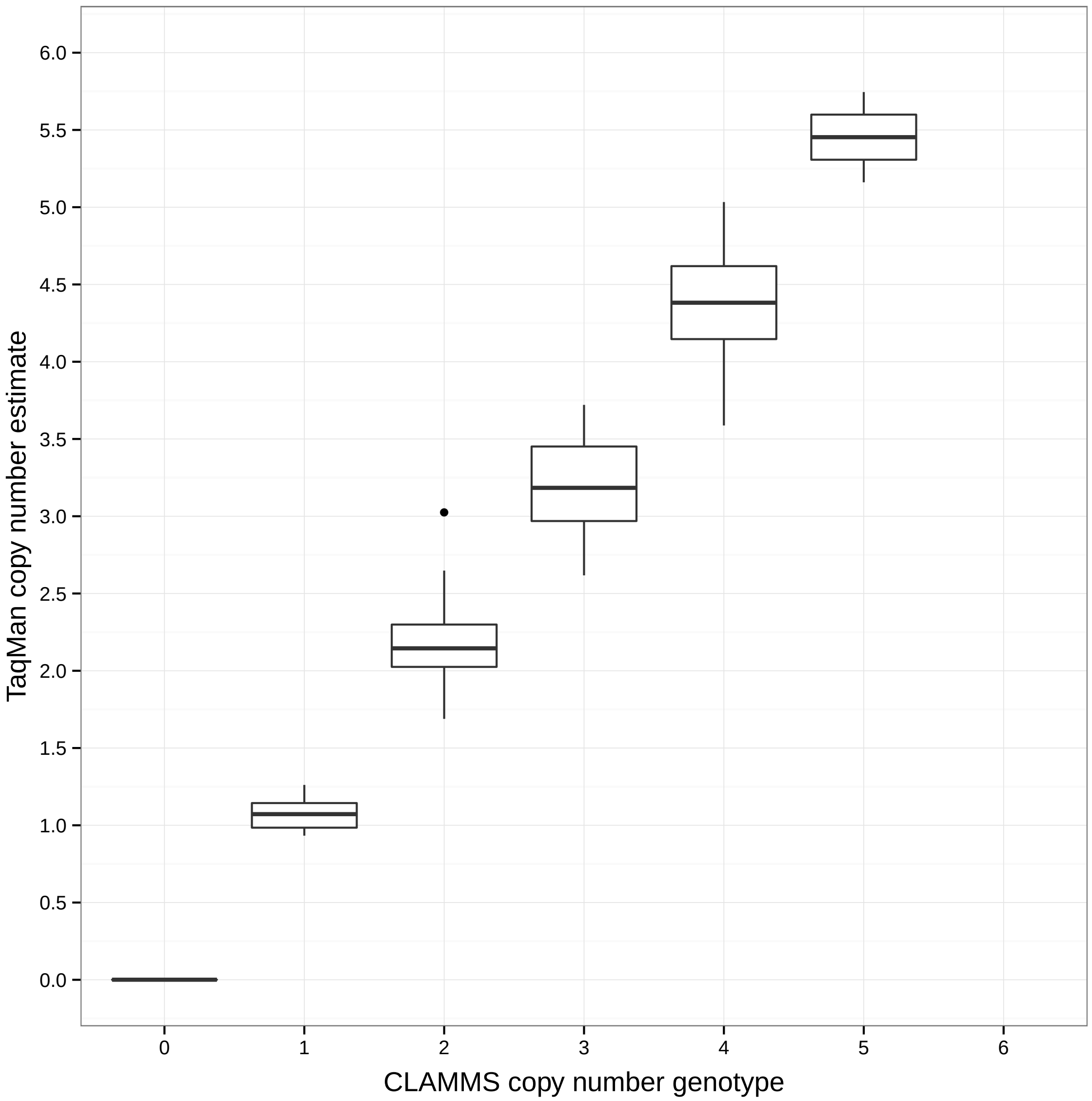
Supplementary Figure 5: **Comparison of CLAMMS and TaqMan copy number predictions for the LILRA3 common variant locus.**



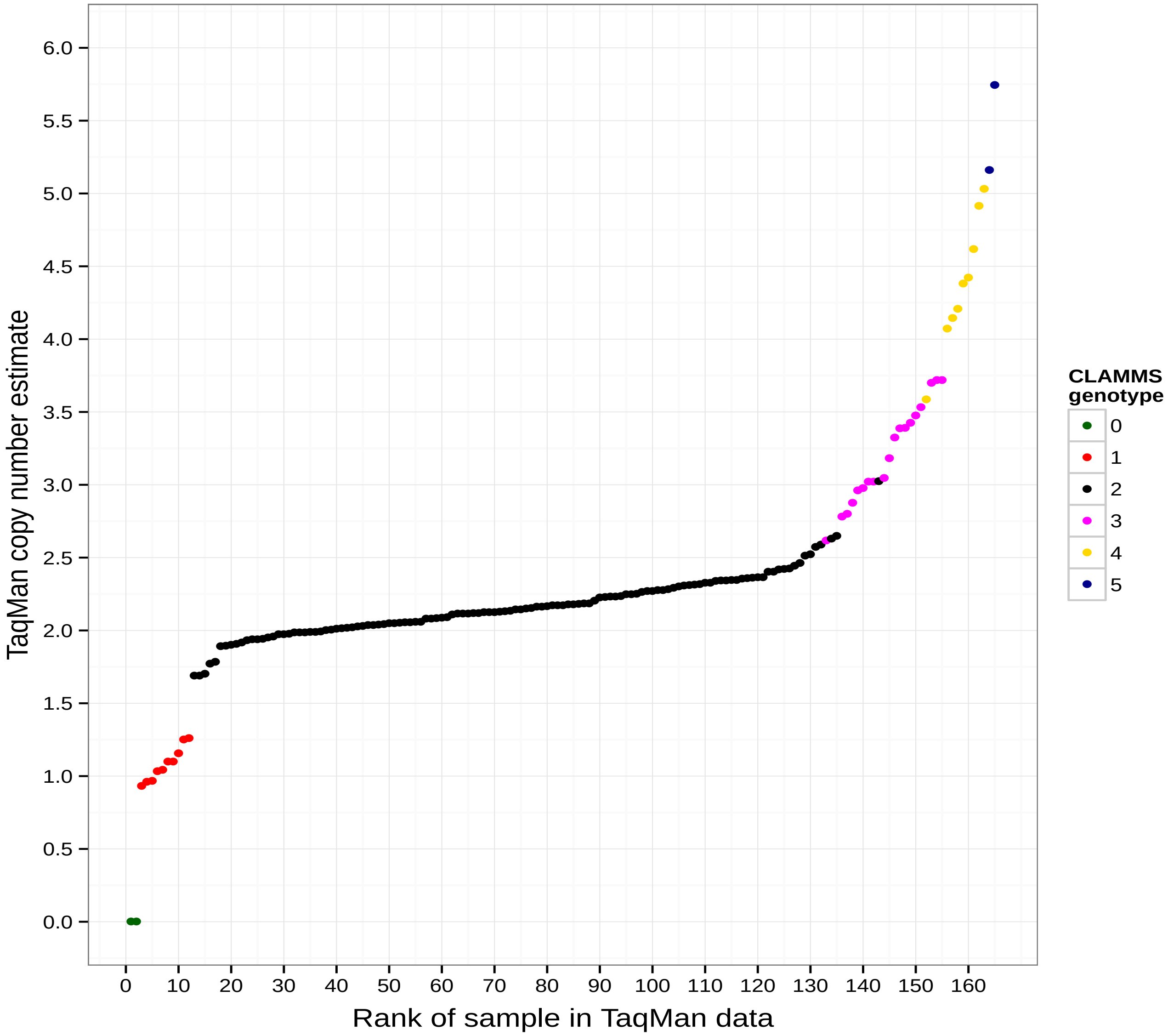
Supplementary Figure 6: **Comparison of CLAMMS and TaqMan copy number predictions for the CDK11A common variant locus.**



Supplementary Figure 7: **Comparison of CLAMMS and TaqMan copy number predictions for the CDK11A common variant locus.**



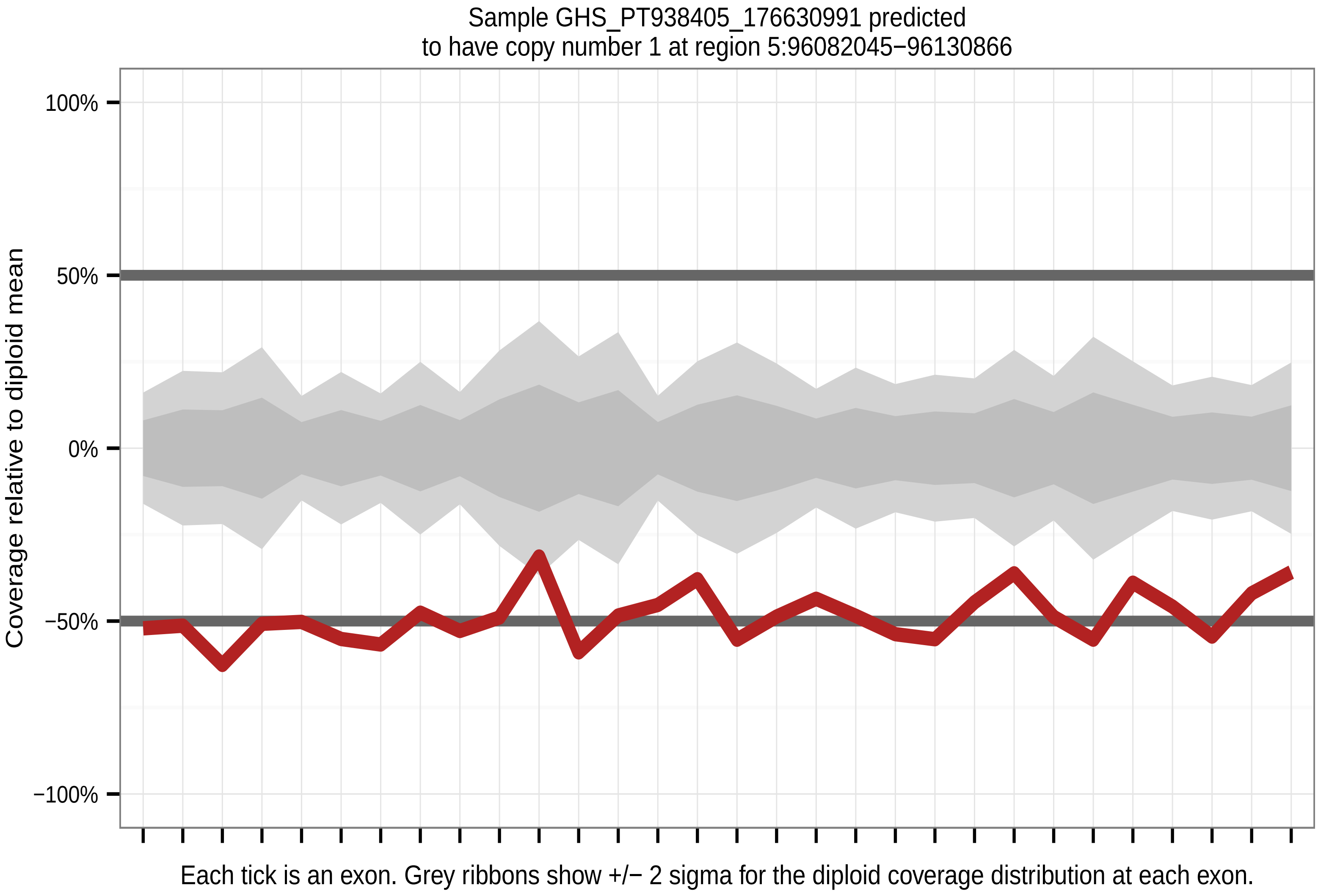
Supplementary Figure 8: **Comparison of CLAMMS and TaqMan copy number predictions for the NAIP common variant locus.**



Supplementary Figure 9: **Comparison of CLAMMS and TaqMan copy number predictions for the NAIP common variant locus.**

8 CNV Visualization Script

Along with CLAMMS' source code, the CLAMMS Github repository also includes a simple script to visualize CNVs. Supplementary Figure 10 shows example output for this script.



References

Backenroth *et al.* (2014) CANOES: detecting rare copy number variants from whole exome sequencing data, *Nucleic Acids Res*, **42** (12), e97.

Benjamini, Yuval, and Speed, Terence P. (2012) Summarizing and correcting the GC content bias in high-throughput sequencing, *Nucleic Acids Res*, **40** (10), e72.

Coe *et al.* (2014) Refining analyses of copy number variation identifies specific genes associated with developmental delay.. *Nat Genet*, **46** (10): 1063-71.

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.

Handsaker *et al.* (2015) Large multiallelic copy number variations in humans. *Nat Genet*, **47** (3), 296-303.

Krumm *et al.* (2012) Copy number variation detection and genotyping from exome sequence data. *Genome Res*, **22** (8), 1525-32.

Li *et al.* (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, **25** (16), 2078-9.

Plagnol *et al.* (2012) A robust model for read count data in exome sequencing experiments and implications for copy number variant calling. *Bioinformatics*, **28** (21), 2747-54.

Stenson *et al.* (2012) The Human Gene Mutation Database (HGMD) and its exploitation in the fields of personalized genomics and molecular evolution. *Curr Protoc Bioinformatics*. doi: 10.1002/0471250953.bi0113s39

Wang *et al.* (2007). PennCNV: An integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome Res*, **17** (11): 1665-1674.