A machine-learning approach for accurate detection of copy number variants from exome sequencing

Vijay Kumar Pounraja,[1] Gopal Jayakar,[2] Matthew Jensen,[1] Neil Kelkar,[2] and Santhosh Girirajan[1,3,4]

[1]Bioinformatics and Genomics Graduate Program of the Huck Institutes of the Life Sciences, The Pennsylvania State University, University Park, Pennsylvania 16802, USA; [2]The Schreyer Honors College, The Pennsylvania State University, University Park, Pennsylvania 16802, USA; [3]Department of Biochemistry and Molecular Biology, [4]Department of Anthropology, The Pennsylvania State University, University Park, Pennsylvania 16802, USA

Copy number variants (CNVs) are a major cause of several genetic disorders, making their detection an essential component of genetic analysis pipelines. Current methods for detecting CNVs from exome-sequencing data are limited by high false-positive rates and low concordance because of inherent biases of individual algorithms. To overcome these issues, calls generated by two or more algorithms are often intersected using Venn diagram approaches to identify "high-confidence" CNVs. However, this approach is inadequate, because it misses potentially true calls that do not have consensus from multiple callers. Here, we present CN-Learn, a machine-learning framework that integrates calls from multiple CNV detection algorithms and learns to accurately identify true CNVs using caller-specific and genomic features from a small subset of validated CNVs. Using CNVs predicted by four exome-based CNV callers (CANOES, CODEX, XHMM, and CLAMMS) from 503 samples, we demonstrate that CN-Learn identifies true CNVs at higher precision (~90%) and recall (~85%) rates while maintaining robust performance even when trained with minimal data (~30 samples). CN-Learn recovers twice as many CNVs compared to individual callers or Venn diagram–based approaches, with features such as exome capture probe count, caller concordance, and GC content providing the most discriminatory power. In fact, ~58% of all true CNVs recovered by CN-Learn were either singletons or calls that lacked support from at least one caller. Our study underscores the limitations of current approaches for CNV identification and provides an effective method that yields high-quality CNVs for application in clinical diagnostics.

[Supplemental material is available for this article.]

Copy number variants (CNVs) are a major source for genomic variation, evolution, and disease (Sebat et al. 2004; Redon et al. 2006; Perry et al. 2008; Girirajan et al. 2011). About 15% of affected individuals referred for clinical genetic testing carry a disease-associated CNV (Miller et al. 2010), making CNV detection an essential aspect of genetic analysis pipelines (Sathirapongsasuti et al. 2011; Krumm et al. 2012). Although the clinical utility of microarrays has not diminished (Coughlin et al. 2012), exome sequencing is becoming a prevalent technology for genetic testing (de Ligt et al. 2013; Yang et al. 2013; Retterer et al. 2016; Stark et al. 2016; Tan et al. 2017). Several algorithms are available to call CNVs from exome-sequencing data in both clinical (Retterer et al. 2016) and disease-specific cohorts, including autism (Krumm et al. 2012), schizophrenia (Fromer et al. 2012), epilepsy (Epilepsy Phenome/Genome Project Epi4K Consortium 2015), and cancer (Koboldt et al. 2012). A common strategy used by these CNV callers is to apply various statistical distributions to model the aggregate read depth of the exons and use read-depth fluctuations between adjacent exons to identify duplication or deletion events (Fromer et al. 2012; Krumm et al. 2012; Backenroth et al. 2014; Jiang et al. 2015; Packer et al. 2015).

Several themes have emerged because of variations in the approaches used by different CNV callers to model read-depth distributions. First, the distributions and algorithms chosen to model read depth depend on the expertise of the researchers and their subjective assumptions about the underlying data. For example, callers such as XHMM (Fromer et al. 2012) assume the read-depth distribution to be Gaussian, whereas CANOES (Backenroth et al. 2014) assumes a negative binomial distribution, CODEX (Jiang et al. 2015) assumes a Poisson distribution, and CoNIFER (Krumm et al. 2012) makes no assumptions about the read-depth distribution. Second, although every method normalizes data to eliminate noise and outliers resulting from GC and repeat content biases, the number of samples required for normalization and the definition of outliers are inconsistent among the callers. For example, a principal component analysis (PCA) based method such as XHMM requires at least 50 unrelated samples for effective normalization, whereas CANOES only requires as few as 15 samples (Backenroth et al. 2014). Similarly, the annotations for "extreme" GC content differ among XHMM (<0.1 or >0.9), CODEX (<0.2 or >0.8), and CLAMMS (<0.3 or >0.7) (Packer et al. 2015). Further, XHMM only considers exome capture targets between the size range of 10 bp and 10 kbp and with average coverage >10× across all samples, whereas CODEX uses targets that are >20 bp long and

1134 **Genome Research**
www.genome.org
29:1134–1143 Published by Cold Spring Harbor Laboratory Press; ISSN 1088-9051/19; www.genome.org

with median coverage >20×. CODEX eliminates targets with mappability scores <0.9, whereas CLAMMS eliminates regions with scores <0.75 along with a custom list of "blacklisted" regions (Packer et al. 2015). Third, the validation methods and the subsets of CNV calls used to estimate sensitivity and specificity measures vary widely among the callers. For example, large CNVs (>100 kbp) validated with microarrays and analyzed with PennCNV (Wang et al. 2007) were used to estimate the performance of CANOES (Backenroth et al. 2014), but only a subset of CLAMMS calls with minor allele frequency <0.1 were validated using polymerase chain reaction (PCR). Also, not all callers report confidence scores for the CNVs they identify. Even when they do, their confidence scales are not directly comparable. These issues influence the number and type of CNVs detected by each caller in a given sample, resulting in pronounced differences in accuracy, false-positive rates, and concordance among the callers (Hong et al. 2016; Yao et al. 2017). Finally, studies using exome-sequencing data to detect CNVs either use predictions made by a single caller (Krumm et al. 2013; Poultney et al. 2013) or use a Venn diagram approach to identify calls with concordance among multiple callers as "high-confidence" CNVs (Krumm et al. 2015; Bademci et al. 2016; Kataoka et al. 2016; Priest et al. 2016). Although using data from multiple callers minimizes false-positive rates, this approach discards a large subset of nonconcordant CNVs, thereby reducing the overall CNV yield (Hong et al. 2016). In addition to a low CNV yield, the reported breakpoints of the concordant calls do not necessarily agree between callers. These limitations associated with individual CNV callers as well as the methods used to integrate predictions from different CNV callers necessitate a better approach to identify and prioritize clinically relevant CNVs.

Here, we propose a machine-learning method called CN-Learn that overcomes the limitations of high false-positive and low concordance rates among calls generated by different CNV algorithms to identify high-confidence CNVs. Our method leverages several attributes intrinsic to each CNV call, such as GC content, mappability of the genomic region, and CNV size, in addition to concordance among the callers. CN-Learn learns the associations between these attributes and the presence or absence of CNVs using a small subset of validated CNVs in the cohort (known truth), and then segregates true CNVs from false positives in the test samples with high precision. Using exome-sequencing data and validations from 503 samples, we demonstrate CN-Learn's ability to recover more than twice as many potentially true variants compared to a Venn diagram approach. Our study reiterates the limitations of existing CNV detection and integration methods, and offers a better alternative that yields a set of high-quality CNVs for application in clinical diagnostics.

Results

We developed CN-Learn as a binary Random Forest classifier that can be trained to differentiate true CNV predictions from false positives using a small subset of validated CNVs (Fig. 1). We identified 12 features that represented the extent of support from individual algorithms and the genomic context for each CNV (Methods). We detected statistically significant correlations for several pairs of the quantitative features (Fig. 2; Supplemental Table 1). Based on a collection of decision trees built using the 12 features extracted from CNVs in the training samples, CN-Learn estimates the probability of each CNV in the test sample to be true (Fig. 2). In addition to a

Random Forest classifier, CN-Learn can also be built as a Logistic Regression (LR) or Support Vector Machine (SVM) based classifier (Supplemental Methods and Results).

CN-Learn detects high-confidence CNVs with high precision and recall rates

To build the CN-Learn classifier, we first identified 41,791 CNV predictions from 503 samples using four exome-based CNV callers (CANOES, CODEX, CLAMMS, and XHMM). Using a read depth–based method (Supplemental Fig. S1) to resolve breakpoint conflicts of overlapping CNV predictions obtained from different callers, we identified 29,101 unique CNV events among the 503 samples (Supplemental Methods and Results). An alternate approach to resolve breakpoint conflicts (Supplemental Fig. S2) also provided the same number of unique CNV events. We selected 2506 of these CNVs from 291 samples with microarray validations that were between 50 kbp and 5 Mbp and spanned regions covered by microarray probes (Supplemental Fig. S3). After determining the proportion of CNVs that overlapped with microarray validations at different thresholds (Supplemental Table 2), we labeled each of the selected CNVs as either "True" or "False" based on a 10% reciprocal overlap threshold. We next used CNVs from 70% of the 291 samples to train CN-Learn as a Random Forest classifier and the remaining 30% of samples to test its performance. Given the uneven distribution of the labels between the two CNV classes (11% true vs. 89% false), we chose precision and recall as the measures of classifier performance (Supplemental Table 2). We captured the aggregate performance of CN-Learn using 10 random draws of training data (10-fold cross-validation), stratified by sample. CN-Learn used the 12 predictors supplied with each CNV in the test set (Methods) and classified the calls as either true or false at 91% precision and 86% recall rates. The overall diagnostic ability of the binary classifier, measured as the area under the receiver operating characteristic (ROC) curve, was 95% (Fig. 3A). Despite sampling different sets of training data for each iteration during cross-validation, the performance of CN-Learn was consistent across all 10 iterations for both precision (91 ± 5%) and recall rates (86 ± 5%) (Fig. 3B).

We also assessed the relative importance of each feature toward making accurate CNV predictions by calculating the Gini index (defined as the total decrease in improvements to the node purity for all splits on each feature averaged over all trees in the forest) (Breiman 2001). We found that features such as the number of exome capture probes spanning a given call, the extent to which CANOES agreed with a CNV prediction, concordance among the callers, and GC content provided the most discriminatory power to the classifier (Fig. 3C). Postclassification analysis of the concordance profile indicated that only 34% of all CNVs classified as true had support from all four callers, whereas the remaining 66% lacked support from at least one caller (Supplemental Fig. S4). We further assessed the performance of CN-Learn on an independent set of 90 samples from the 1000 Genomes Project (Jiang et al. 2015) and observed a precision rate of 93% and recall rate of 86% when using 70% training data (Supplemental Figs. S5, S6). When we generated precision/recall rates that are directly comparable with the concordance-based methods, we observed a consistently superior performance for CN-Learn (Supplemental Fig. S7; Supplemental Tables 3, 4; Supplemental Methods and Results). Overall, these results highlight the ability of CN-Learn to look beyond the single measure of concordance typically used in a Venn diagram–based approach and to use the discriminatory power of
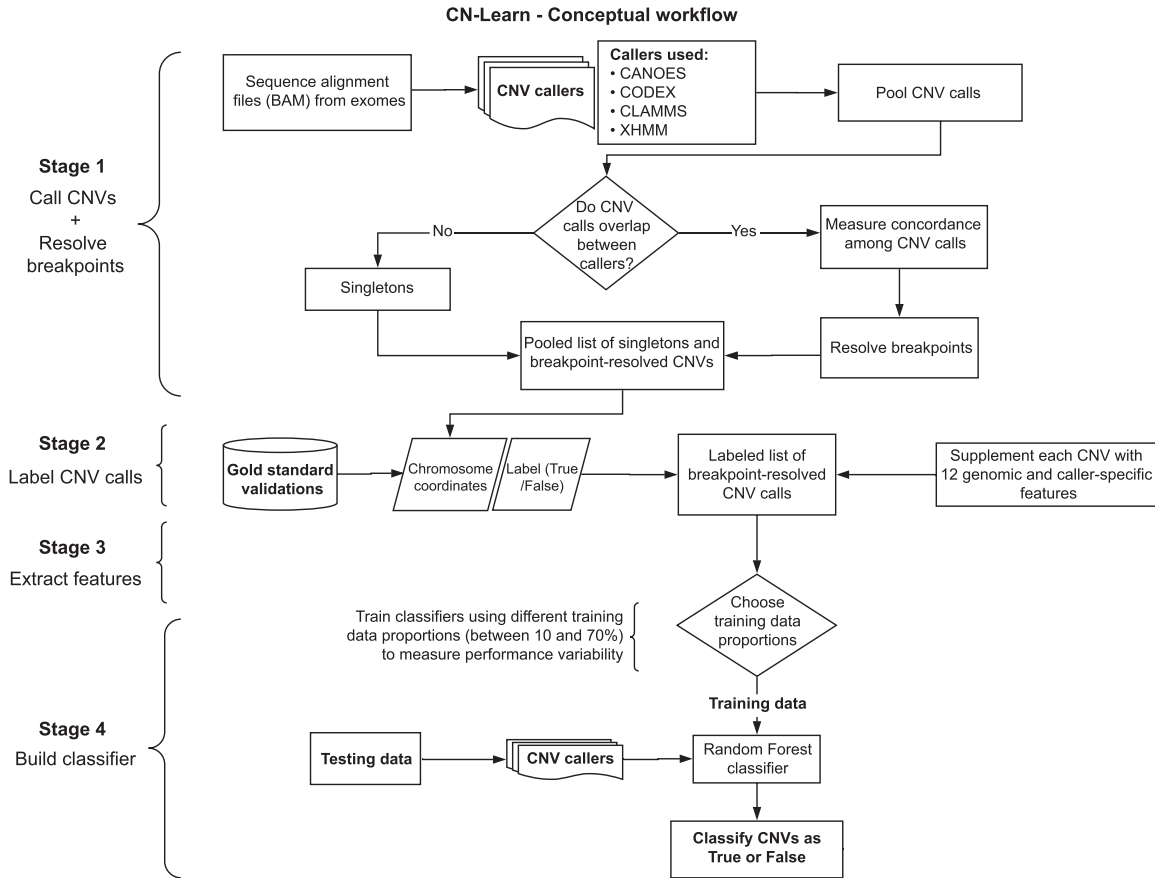
CN-Learn - Conceptual workflow

Figure 1. Overview of the CN-Learn pipeline. The CN-Learn pipeline consists of preprocessing steps (Stages 1 and 2), followed by building the classifier using training data and discriminatory features, and finally running the classifier on the test data. The complete pipeline is outlined as follows. Stage 1: CNV predictions were made using four exome-based CNV callers. Although CANOES, CODEX, CLAMMS, and XHMM were used in this study, a generic pipeline can be constructed with a different set or number of callers. Breakpoints of overlapping calls from multiple callers were then resolved. Stage 2: Breakpoint-resolved CNVs were labeled as "True" or "False" based on the overlap with "gold standard" calls and subsequently used to train CN-Learn. Stage 3: Caller-specific and genomic features were extracted for the labeled CNVs in the training and testing sets. Stage 4: CN-Learn was trained as a Random Forest classifier using the extracted features of the CNVs in the training set to make predictions on the CNVs from the testing set.

additional variables to identify high-confidence CNVs in a systematic manner.

Performance of CN-Learn is robust across varying CNV sizes, frequencies, and training sets

We independently trained CN-Learn using varying proportions of training data (between 10% and 70% in increments of 10%) and observed steady performance gains with increase in the number of training samples (Fig. 3A). Even when the classifier was built using just 10% of the total samples ($n = 29$ samples), we obtained 90% precision and a recall rate of 75%, indicating the robustness of the classifier when learning from minimal training data. We further trained CN-Learn independently at four size ranges of CNVs and observed a modest increase in precision with increase in CNV size (90% for 50–100 kbp CNV to 97% for 0.5–5 Mbp CNV) (Fig. 3D). In fact, the precision achieved by CN-Learn at each size interval was substantially higher than the precision achieved by the individual CNV callers. We also observed comparable precision and recall rates when CN-Learn was run on breakpoint-resolved CNVs obtained by merging overlapping predictions (Supplemental Fig. S8). We then tested the performance of CN-Learn for differ-

ent classes of CNVs based on their frequency in the cohort (very rare, rare, common, very common) and found precision (>90%) and recall (>80%) rates that were consistent across the CNV frequency spectrum (Supplemental Fig. S9; Supplemental Methods and Results). In fact, when added as an additional predictor to CN-Learn, CNV frequency showed the highest discriminatory power (16%) of any predictor but did not contribute to significant improvements in the performance of the classifier (Supplemental Figs. S10, S11). Although the performance of the Random Forest classifier was robust, Logistic Regression and SVM classifiers failed to match its performance (Supplemental Figs. S12, S13; Supplemental Methods and Results).

We next assessed the performance of CN-Learn by considering calls made by CLAMMS as the truth set and classified CNVs obtained from predictions made by the other three callers (CANOES, CODEX, and XHMM). We categorized 25,019 breakpoint-resolved CNVs (Supplemental Fig. S3) from all 503 samples as either "True" or "False" based on their intersection (10% reciprocal overlap) with CNVs predicted by CLAMMS. Given the higher resolution of CNVs detected from exome data relative to SNP microarrays, we used a total of 16,497 CNVs between 5 kbp and 5 Mbp in size to build CN-Learn (Supplemental Fig. S3). CN-Learn achieved an aggregate
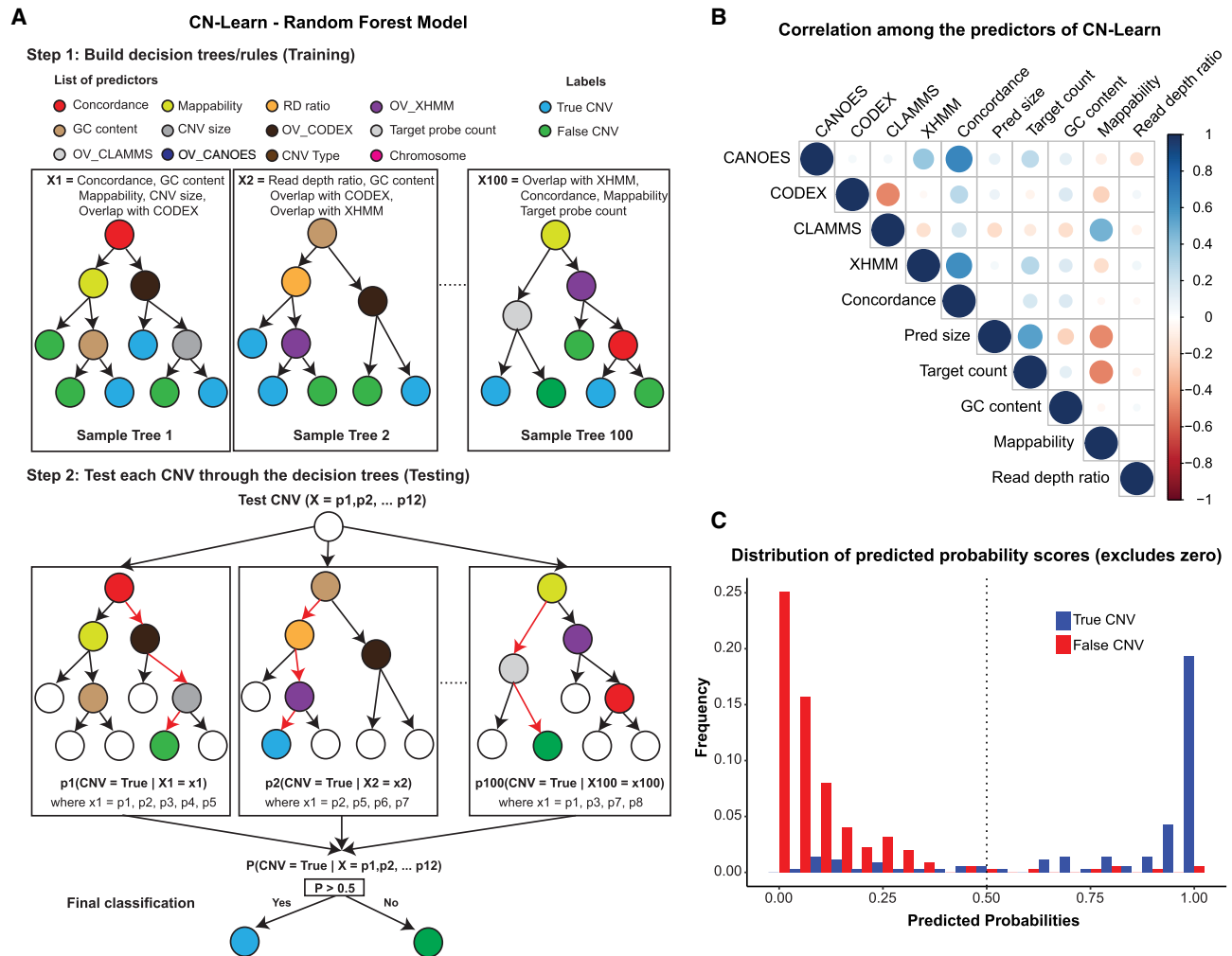
Figure 2. Illustration of the Random Forest model used to build CN-Learn. (*A*) The inner workings of the Random Forest model used for training CN-Learn is shown. Twelve features were used to grow 100 trees with different subsamples of predictors and training data to classify each CNV in the test set as either true or false. If the predicted probability score was >0.5, the CNV call was classified as true. Calls with predicted probability score <0.5 were labeled as false. (*B*) A Spearman's rank correlation between pairs of quantitative predictors used by the CN-Learn classifier is shown. The color of the circles indicates the direction of the correlation, and the size of the circles indicates the strength of the correlation. The correlation scores are provided in Supplemental Table 1. (*C*) The frequency of microarray-validated and -invalidated CNVs, distributed across 20 bins of increasing predicted probability scores, is shown. For the probability bins <0.5, the proportion of CNVs that were validated was higher than the proportion of CNVs that were not validated. This indicated that the classification score of 0.5 is an appropriate threshold for distinguishing true and false CNVs.

precision rate of 94% with an overall recall rate of 85% during the 10-fold cross-validation and achieved comparable performance when independently trained with CNVs at different training sample proportions (Supplemental Fig. S14A). The performance variability observed both during cross-validation and across the size ranges was comparable to the variability observed when microarray was used as the truth set (Supplemental Fig. S14B,C). Among the features used by CN-Learn to classify the CNVs, the relative importance of the mappability score was the highest, with GC content being the next important feature (Supplemental Fig. S14C). Although caller-specific features contributed to the discriminatory power of CNV classification when microarrays were used for validation, genomic features played a more prominent role when a sequence-based method was used for validation (Supplemental Fig. S15). These results show that the performance of CN-Learn is robust with minimal training data, at different size ranges, and even when orthogonal validations are not available.

CN-Learn recovers true CNVs that lack complete concordance among callers

To assess the ability of CN-Learn to correctly identify true CNVs that lack support from multiple callers, we analyzed the concordance profile of all CNVs classified by CN-Learn as true, based on microarray validations, before and after CN-Learn classification. Using a single random draw of 29 samples (10%) to train CN-Learn and 262 samples as the test set, we obtained predictions for 2245 CNVs with microarray validations (Fig. 4A). Among these predictions, only 122/2245 CNVs (5.4%) were supported by all four CNV callers prior to classification by CN-Learn. The strong concordance of the four methods for these CNV predictions was corroborated by a high microarray validation rate (116/122, 95%) (Fig. 4A). In contrast, CNVs that lacked support from one or more callers were less likely to intersect with microarray data. For example, only 41% (11/27) of the CNVs with support from
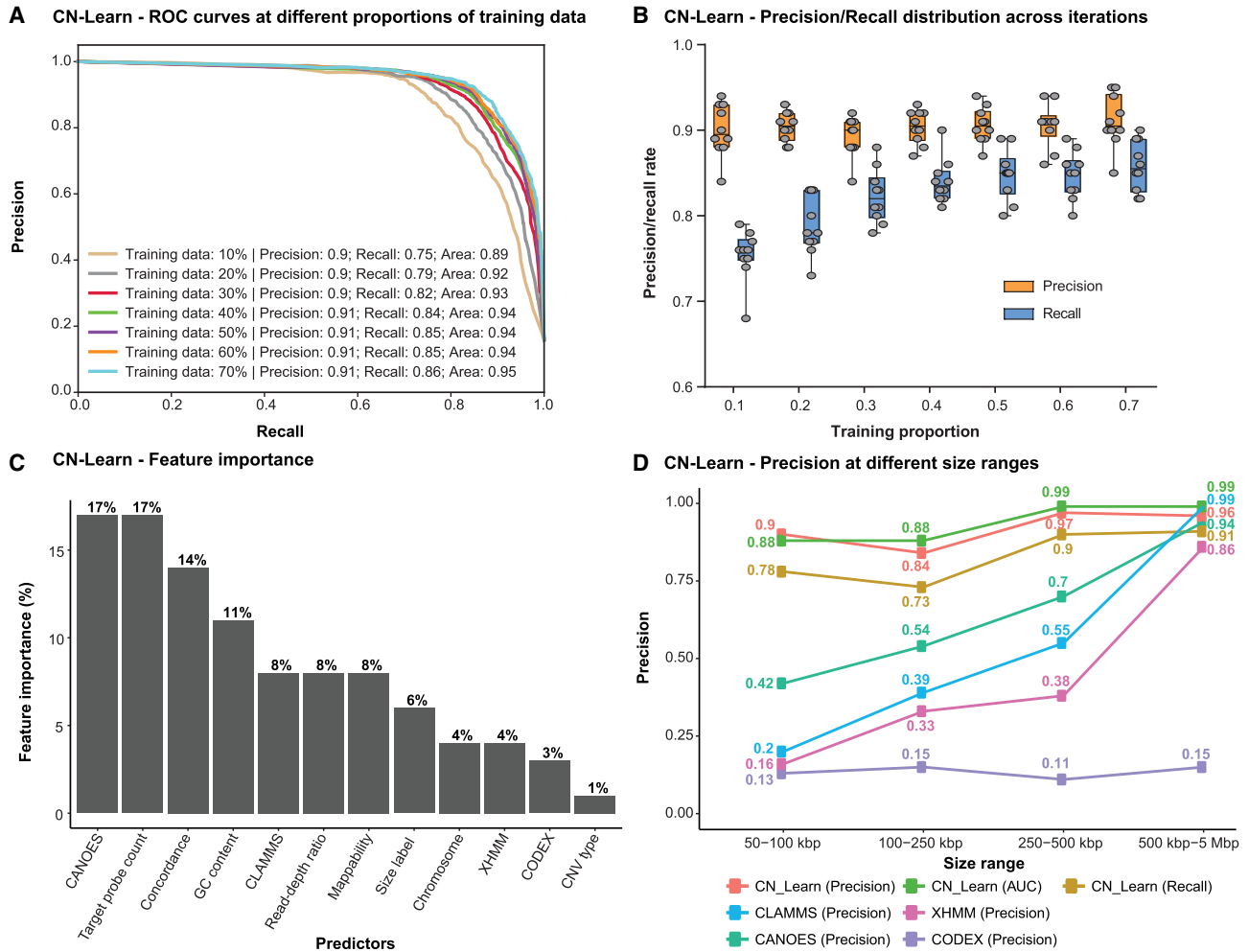
Figure 3. Characteristics of the CN-Learn binary Random Forest classifier. (*A*) Receiver operating characteristic (ROC) curves indicating the trade-off between the precision and recall rates when CN-Learn was trained as a Random Forest classifier are shown. Each curve represents the performance achieved when using different proportions of samples to train CN-Learn, starting from 10% up to 70% in increments of 10%. The results shown were from experiments aggregated across 10-fold cross-validation. (*B*) Variability observed in the precision and recall measures during the 10-fold cross-validation at various proportions of training data is shown. Both measures varied within ± 5% of their corresponding averages. (*C*) The relative importance of each genomic and caller-specific feature supplemented to CN-Learn is shown. Data shown here are the averages of the values obtained across 10-fold cross-validation after using 70% of the samples for training. (*D*) Precision rates for CNVs when CN-Learn was trained at four different size ranges compared to the precision rates of CNVs from individual callers are shown. Precision rates for CN-Learn were estimated as its classification accuracy (true positives/[true positives + false positives]), whereas the precision rates for the individual callers were calculated as the proportion of CNVs at each size range that were validated by the microarray calls.

XHMM, CODEX, and CLAMMS intersected with microarray data. After classification by CN-Learn, 84% (266/315) of all CNVs labeled as "True" intersected with microarray calls, indicating the high classification accuracy achieved by the classifier (Fig. 4A). All CNVs classified as true and false from this analysis are provided in Supplemental Tables 5, 6. Of the 266 CNVs validated by microarrays and correctly identified as true by the classifier, only 42% (112/266) were supported by all four CNV callers. The remaining 58% (154/266) were either singletons or calls that lacked support from at least one caller, and they would have been excluded if complete concordance was used as the only determinant for selecting high-confidence CNVs. For example, 13% of all true CNVs (35/266) recovered by CN-Learn were missed by CLAMMS but were identified by one or more of the other callers, reiterating the limitations of using a single exome-based CNV caller for variant predictions. Furthermore, CN-Learn managed to recover 97% (112/116)

of the true CNVs validated by microarrays that were supported by all of the four callers. Although these 112 CNVs could have been identified by a simple caller intersection approach, CN-Learn was uniquely able to recover CNVs that lacked support from at least one other caller. For example, CN-Learn classified 66 CNVs supported by CANOES, CODEX, and XHMM as true, of which 51% (34/66) were also validated by microarrays. This result is notable because, without using CN-Learn for classification, only 31% (46/148) of CNVs supported by CANOES, CODEX, and XHMM would intersect with the microarray calls, indicating the inherently high false-positive rate associated with simply intersecting calls from individual callers using a Venn diagram. Our results indicate that in addition to correctly identifying almost every true CNV reported by the four callers, CN-Learn overcame the limitations of the Venn diagram–based approach and recovered 154 additional high-confidence CNVs with suboptimal concordance, thereby
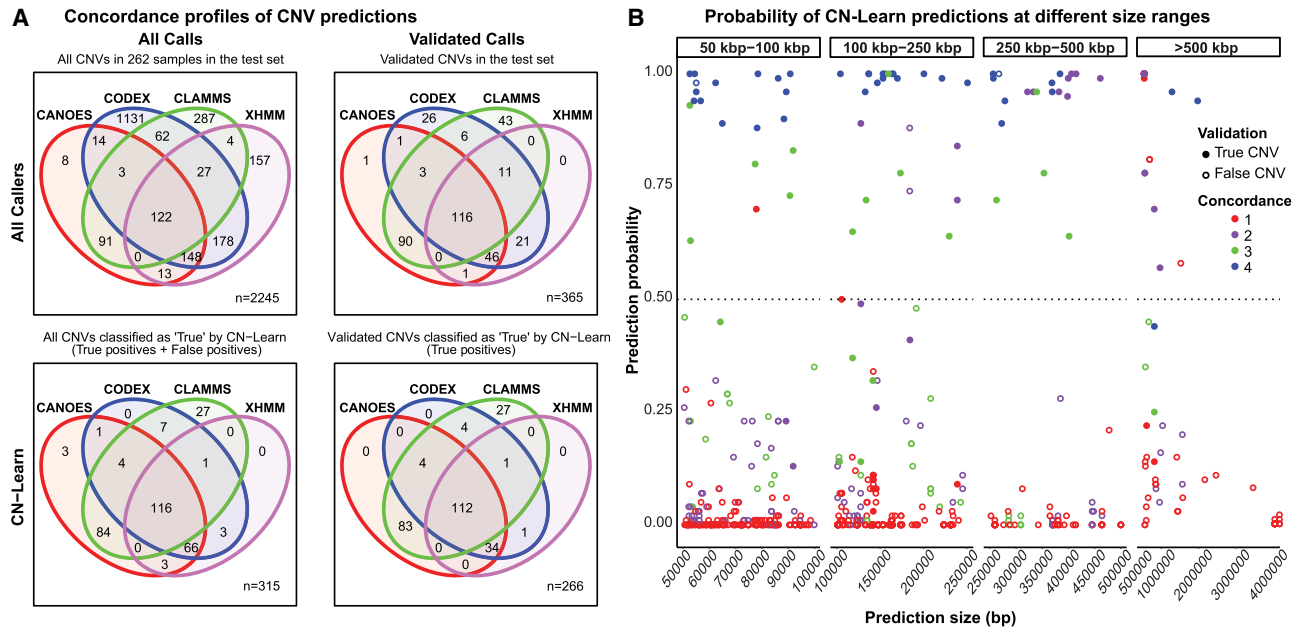
Figure 4. Concordance profiles of CNVs before and after classification by CN-Learn using microarray calls as the "gold standard" validation set. (*A*) Venn diagrams are shown for CNVs (≥50 kbp) identified from a random draw of 262 samples out of the total 291 samples before (*top*) and after (*bottom*) classification by CN-Learn. The Venn diagrams show the overlap of calls among the four callers (*top left*) versus those that were validated by microarrays (*top right*). Venn diagrams depicting the overlap of all true calls among the four callers after classification by CN-Learn (*bottom left*) and true calls that were also validated by microarrays are shown (*bottom right*). (*B*) The distribution of all calls within the 262 samples based on the probability scores (*y*-axis) predicted by CN-Learn across four size ranges (*x*-axis) is shown. The number of overlaps of each CNV with exome-based CNV callers is represented by different colors. CNVs that validated with microarrays are indicated by filled circles, and CNVs that did not validate with microarrays are represented by hollow circles.

improving the CNV yield by 2.37-fold (266/112). Furthermore, the true positive calls recovered by CN-Learn spanned the entire spectrum of CNV size range (Fig. 4B). We also obtained comparable improvements in CNV yield when CNVs predicted by CLAMMS were used as the truth set (Supplemental Figs. S16, S17; Supplemental Tables 5, 6; Supplemental Methods and Results).

Discussion

Exome sequencing is a cost-effective assay to accurately identify single base pair mutations and small insertions or deletions within protein-coding genes, which are more likely to cause disease than mutations in other regions of the genome (Bamshad et al. 2011). Even with the emergence of whole-genome sequencing (WGS) techniques, exome sequencing has continued to flourish, resulting in an increase in the availability of data sets for analysis. This widespread data availability has allowed for repurposing data generated for single-nucleotide variant detection to identify CNVs as well (Koboldt et al. 2013). Given the high false-positive rates among calls reported by individual exome-based callers and platform-specific biases (Tan et al. 2014), existing CNV identification pipelines that leverage more than one calling algorithm have depended on naive Venn diagram–based integration approaches to identify high-confidence CNVs. One reason for the overwhelming trust in such approaches could be that the self-reported performance measures of each individual algorithm are typically high. Therefore, there is little reason to doubt that performance variations among the callers could affect the precision of Venn diagram–based integration approaches. Although we found that the likelihood of a CNV prediction to be true increased with an increase in the number of callers supporting it (Supplemental Fig.

S18), there are two key limitations of this approach. First, even among completely concordant predictions, the observed false-positive rates were not zero. Inflated false positives pose a large hurdle for researchers interested in identifying clinically relevant CNVs, because it is time consuming to validate a large number of false-positive CNVs using orthogonal methods. Second, the Venn diagram approach failed to identify a large subset of nonconcordant or singleton CNVs supported by microarray validations. In fact, approximately one true CNV (>50 kbp) per individual in our cohort would have been discarded under a Venn diagram–based approach. This evidence reiterates that Venn diagram–based approaches do not have the required precision for usage in both clinical and research settings.

The utility of alternate methods for CNV detection hence rests on the ability to both eliminate false positives among completely concordant predictions and recover true CNVs that lack adequate support from multiple callers. Therefore, instead of addressing the shortcomings of existing methods by developing yet another CNV detection tool, our study focused on offering a reliable integrative approach. In this study, we demonstrate a machine-learning approach that leverages caller-specific and genomic contexts from a subset of validated calls to identify high-confidence CNVs more thoroughly than individual callers on their own or Venn diagram–based approaches. CN-Learn achieved precision as high as 94% while doubling the CNV yield, showing its ability to capture singletons that would have been missed by other approaches. Moreover, the precision of CN-Learn is robust to variation in CNV size, ratio of training to testing samples, and validation method/type, indicating the utility of CN-Learn in a variety of clinical or research contexts. The use of multiple variables that capture the genomic context of each CNV in addition to caller concordance is a

major reason for the high recovery and precision rates achieved by CN-Learn. In fact, GC content and mappability scores were some of the most useful features in predicting true CNVs.

One of the limitations of CN-Learn is its dependency on a small set of biologically validated CNVs. Our study leverages microarray validations to evaluate the predictions of each caller and to label the breakpoint-resolved CNV predictions as either true or false. Although microarrays may not be considered as a "gold standard" for CNV detection, we were able to use microarray calls as an orthogonal validation to demonstrate the utility of CN-Learn. Our study also serves as a proof-of-principle for future studies that could utilize CN-Learn with "gold standard" CNVs curated from multiple genomic technologies, such as Pacific Biosciences (PacBio) SMRT sequencing (John et al. 2009), Illumina long-read sequencing (Voskoboynik et al. 2013), 10x Genomics linked-read sequencing (Zheng et al. 2016), BioNano Genomics genome mapping (Lam et al. 2012; Mak et al. 2016), or PCR. Another limitation of our study is the time and computational capacity required to run four different CNV callers. Each CNV calling algorithm extracts read-depth information for each sample using tools such as SAMtools (Li et al. 2009), BEDTools (Quinlan and Hall 2010) or Genome Analysis Toolkit (GATK) (McKenna et al. 2010), which are often the rate-limiting steps for each pipeline. Future studies could simplify this data extraction layer by using a single read-depth tool without adversely impacting the results of the individual callers. Finally, we also acknowledge the complexity associated with resolving breakpoint conflicts from multiple callers that arise during data integration. Although we presented two strategies to resolve breakpoints of concordant CNVs (Methods), future studies could explore more effective strategies before using CN-Learn. Because population-scale projects continue to generate large exome-sequencing data sets, the need and importance of robust CNV integration approaches such as CN-Learn is apparent.

Overall, CN-Learn integrates predictions from multiple CNV callers and overcomes the limitations of existing integration approaches, even when the availability of samples with biological validations is limited. Although we chose a set of four CNV calling algorithms with microarray validations, CN-Learn framework can be extended to use different sets of CNV callers or validation types to identify high-confidence CNVs, making our framework easy to adopt and customize. This can allow clinicians and researchers to use their preferred callers and validation methods to detect CNVs from exomes. Our results suggest that a small set of high-quality validated CNVs and an objective machine-learning method can help alleviate several shortcomings of existing integration approaches to generate an informed set of clinically relevant CNVs.

Methods

Samples

We obtained exome-sequencing data (BAM files) from 503 individuals in the Simons Variation in Individual Project (Simons VIP) from the Simons Foundation Autism Research Initiative (SFARI) via SFARI Base following appropriate approvals (https://www .sfari.org/resource/sfari-base/). Exome-sequencing reads were generated using the Agilent SureSelect Human All Exon v2.0 capture kit containing 182,430 autosomal probes (Simons VIP Consortium 2012) and were aligned to the hg19 reference genome. Overall, 187 individuals within this cohort carry the

16p11.2 deletion, and 143 individuals carry the reciprocal duplication. Single-nucleotide polymorphism (SNP)-based microarray data were available for 291 samples, and CNVs >100 kbp were confirmed experimentally using array CGH (Duyzend et al. 2016). We also obtained 90 samples from the ftp site of the 1000 Genomes Project (The 1000 Genomes Project Consortium 2010) data collection (ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase3/data), of which 46 samples were sequenced at the Washington University Genome Sequencing Center using the Agilent SureSelect All Exon v2.0 capture kit, and 44 samples were sequenced at Baylor College of Medicine using the Roche HGSC VCRome capture kit. We used 186,065 autosomal probes from the consensus probe set (ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/reference/exome_pull_down_targets/20130108.exome.targets.bed) to make CNV predictions. Realigning the reads to hg38 (GRCh38) would not significantly affect the conclusions of the paper, because the CNV predictions are restricted to sequences captured by the exome capture probes.

Exome CNV callers

We chose four exome CNV callers to obtain the initial sets of CNV calls: CANOES, CLAMMS (v1.1), CODEX (v0.2.2), and XHMM (Fromer et al. 2012; Backenroth et al. 2014; Jiang et al. 2015; Packer et al. 2015). As the Simons VIP samples were sequenced in two sets (312 and 191 samples), each set of samples was treated as individual batches for running the CNV calling pipelines. XHMM, CANOES, and CODEX were run using the default parameters. CLAMMS models were built with the assumption that the samples were independent without accounting for batch effects. All of the parameters used to make CNV calls using the four callers for this study can be found in the pipelines provided as part of the CN-Learn distribution (https://github.com/girirajanlab/CN_Learn) (Supplemental_Code.zip). Running the four CNV callers yielded 41,791 calls of varying sizes (Supplemental Fig. S19). These original CNV calls (deletions and duplications) were then characterized based on their level of concordance among the four callers (Supplemental Methods and Results).

Resolving breakpoint conflicts for overlapping predictions

Multiple CNV callers can make predictions that overlap with each other in a given genomic region. Treating such overlapping predictions with different breakpoints as separate CNVs would result in double counting the calls for the same CNV event. Therefore, it is important to merge concordant predictions and represent them as a single event for downstream analyses. We developed a five-step procedure that uses fluctuations in local read depth to resolve breakpoint conflicts among overlapping predictions by identifying the most likely start and end coordinates of the underlying event. A detailed explanation of this strategy is presented in the Supplemental Methods and Results (Supplemental Figs. S1, S20). As an alternate strategy, we resolved breakpoint conflicts by simply selecting the smallest and largest coordinates among the overlapping predictions as the start and end coordinates of the underlying CNV event. This strategy is also described in the Supplemental Methods and Results (Supplemental Fig. S2). Applying both methods to the 41,791 CNVs detected in the Simons VIP samples yielded 8382 unique breakpoint-resolved CNV events, in addition to the 20,719 singleton calls (Supplemental Fig. S3).

CNV validations

SNP microarray data were generated by the Simons VIP Consortium (Duyzend et al. 2016) using Illumina OmniExpress-12 microarrays for 272 samples and OmniExpress-14 for 19

samples. PennCNV (v.1.0.3) was used to identify CNVs from microarray data for all 291 individuals using standard parameters (Wang et al. 2007). Individual and family-based (trios and quads) CNV calls were combined for autosomal chromosomes, whereas CNVs on Chromosome X were called only at the individual level. CNV calls with ≥1 bp overlap or gaps <20% of the total CNV length and <50 kbp were merged. CNVs ≥50 kbp in length and containing ≥5 SNP target probes were subsequently considered for further analyses.

For the 1000 Genomes samples, we pooled the validated calls from three sources (McCarroll et al. 2008; The International HapMap 3 Consortium 2010; Conrad et al. 2010) that were used to measure the performance of CODEX (Jiang et al. 2015). Merging the overlaps among the 10,235 CNVs resulted in 7302 CNVs (Supplemental Table 4) that were used to label the CNVs predicted for the 90 samples from the 1000 Genomes Project.

Feature selection for CN-Learn classifier

We identified 12 features that represent the extent of support provided by the four individual callers and genomic context as predictors of true CNVs to the CN-Learn Random Forest classifier. Because we used four callers in our study, the extent of overlap calculated during the breakpoint-resolution process (Supplemental Fig. S1; Supplemental Methods and Results) served as individual predictors. Concordance count and read-depth ratio (RD_{ratio}) for both breakpoint-resolved concordant calls and singletons were also supplied as features. Because individual algorithms use different GC and repeat content (mappability) thresholds to classify CNV predictions as outliers, CNVs with extreme GC content or low mappability could be predicted by one caller but discarded as an outlier by the other callers. To take this into account, we extracted GC content data using the "nuc" option in BEDTools (Quinlan and Hall 2010) and mappability scores (Derrien et al. 2012) using the "bigWigAverageOverBed" option in kentUtils (Kent et al. 2010) for use as predictors. Similarly, the efficacy of CNV detection can vary across chromosomes, size ranges, and CNV type (duplication/deletion). To take these variations into account, we used chromosome number, CNV size, CNV type, and the number of exome capture probes as the final set of CN-Learn features.

Probability estimation and classification using CN-Learn

CN-Learn leverages the features extracted for the CNVs in the training samples to build a Random Forest (Breiman 2001) classifier with hundreds of decision trees and estimates the probability of each CNV in the test samples to be true (Fig. 2A). Decision trees have been shown to perform well when the distribution of observations is unbalanced between the classes of interest (Cieslak and Chawla 2008). Given the high false-positive rates of CNV callers, an uneven split between the number of true and false predictions is likely to occur in clinical samples. In the samples we analyzed, 10% of the CNVs overlapped with the truth sets derived from microarray calls at a 10% reciprocal overlap threshold (Supplemental Table 2). To address the imbalance between the number of true and false CNVs, we stratified the training data by sample to accurately reflect the clinical setting as follows: CNVs in p% of the samples were used for training, and the remaining $(1-p)$% were used as the testing set. For a random forest built with "N" trees, "M" predictors, and "C" classes, the probability of an observation "o" belonging to the class "c" (true or false) can be expressed as $p_{o,c} = Pr(Y = c | X = x_i)$, where x_i is a vector that captures the values for each of the 12 predictors, and Y is the outcome variable. The probability $p_{o,true}$ of the CNV "o" being true in the test set was then measured as the proportion of trees in the forest that assigned it to the true class.

Specifically, the probability of a CNV prediction can be represented as

$$P_{o,true} = Pr(Y = True | X = x_i)$$

$$= \frac{\text{number of trees that identify the CNV as true}}{\text{number of trees in the forest}},$$

where X represents the values of the following features for the observation "o": $x_1 =$ overlap proportion with CANOES; $x_2 =$ overlap proportion with CODEX; $x_3 =$ overlap proportion with CLAMMS; $x_4 =$ overlap proportion with XHMM; $x_5 =$ concordance among callers; $x_6 =$ read-depth ratio; $x_7 =$ chromosome number; $x_8 =$ CNV type (duplication/deletion); $x_9 =$ CNV size; $x_{10} =$ target probe count; $x_{11} =$ GC content; and $x_{12} =$ mappability.

CNV calls with predicted probability scores >0.5 were then classified as true. We selected this cutoff based on the distribution of validated (true) and invalidated (false) CNVs across the predicted probability scores (ranging between 0 and 1). For both duplications and deletions, the proportion of true calls compared to false calls was higher for CNVs with probability scores >0.5 (Fig. 2C). This indicated that at a 0.5 threshold, the classifier recovers as many true CNVs (recall) as possible without compromising on the false-positive rate (precision).

Statistical analysis

All statistical analyses, including the calculation of precision-recall rates, feature importance, and ROC areas, were performed using the Python library scikit-learn (v 0.18.1) (Pedregosa et al. 2012). Plots were generated using the R package ggplot2 (Wickham 2016) and the Python library Matplotlib (Hunter 2007).

Software availability

CN-Learn is available as an open-source software at https://github.com/girirajanlab/CN_Learn and provided as Supplemental_Code.zip. In addition to the scripts necessary to run CN-Learn, we also provided simplified and easily parallelizable pipelines for each of the original CNV calling algorithms (CANOES, CLAMMS, CODEX, and XHMM) used in this study. Furthermore, in order to simplify installation and avoid software version incompatibilities, we provide a Docker image with all the required tools and software packages preinstalled to run CN-Learn. Users can download the image using the command "docker pull girirajanlab/cnlearn" on any Linux platform to run CN-Learn. Detailed instructions to install and use Docker and CN-Learn are provided in the README.md file in Supplemental_Code.zip and at the landing page for the software on GitHub. The script used to generate the plots is also provided in Supplemental_Code.zip and in the "research" directory of the CN-Learn GitHub repository.

References

The 1000 Genomes Project Consortium. 2010. A map of human genome variation from population-scale sequencing. *Nature* **467:** 1061–1073. doi:10.1038/nature09534

Backenroth D, Homsy J, Murillo LR, Glessner J, Lin E, Brueckner M, Lifton R, Goldmuntz E, Chung WK, Shen Y. 2014. CANOES: detecting rare copy number variants from whole exome sequencing data. *Nucleic Acids Res* **42:** e97. doi:10.1093/nar/gku345

Bademci G, Mahdieh N, Bonyadi M, Duman D, Cengiz FB, Menendez I, Horta OD, Shirkavand A, Zeinali S, Subasioglu A, et al. 2016. Comprehensive analysis via exome sequencing uncovers genetic etiology in autosomal recessive non-syndromic deafness in a large multiethnic cohort. *Genet Med* **18:** 364–371. doi:10.1038/gim.2015.89

Bamshad MJ, Ng SB, Bigham AW, Tabor HK, Emond MJ, Nickerson DA, Shendure J. 2011. Exome sequencing as a tool for Mendelian disease gene discovery. *Nat Rev Genet* **12:** 745–755. doi:10.1038/nrg3031

Breiman L. 2001. Random forests. *Mach Learn* **45:** 5–32. doi:10.1023/A:1010933404324

Cieslak DA, Chawla N V. 2008. Learning decision trees for unbalanced data. *Mach Learn Knowl Discov Databases* **5211:** 241–256. doi:10.1007/978-3-540-87479-9_34

Conrad DF, Pinto D, Redon R, Feuk L, Gokcumen O, Zhang Y, Aerts J, Andrews TD, Barnes C, Campbell P, et al. 2010. Origins and functional impact of copy number variation in the human genome. *Nature* **464:** 704–712. doi:10.1038/nature08516

Coughlin CR II, Scharer GH, Shaikh TH. 2012. Clinical impact of copy number variation analysis using high-resolution microarray technologies: advantages, limitations and concerns. *Genome Med* **4:** 80. doi:10.1186/gm381

de Ligt J, Boone PM, Pfundt R, Vissers LELM, Richmond T, Geoghegan J, O'Moore K, de Leeuw N, Shaw C, Brunner HG, et al. 2013. Detection of clinically relevant copy number variants with whole-exome sequencing. *Hum Mutat* **34:** 1439–1448. doi:10.1002/humu.22387

Derrien T, Estellé J, Sola SM, Knowles DG, Raineri E, Guigó R, Ribeca P. 2012. Fast computation and applications of genome mappability. *PLoS One* **7:** e30377. doi:10.1371/journal.pone.0030377

Duyzend MH, Nuttle X, Coe BP, Baker C, Nickerson DA, Bernier R, Eichler EE. 2016. Maternal modifiers and parent-of-origin bias of the autism-associated 16p11.2 CNV. *Am J Hum Genet* **98:** 45–57. doi:10.1016/j.ajhg.2015.11.017

Epilepsy Phenome/Genome Project Epi4K Consortium. 2015. Copy number variant analysis from exome data in 349 patients with epileptic encephalopathy. *Ann Neurol* **78:** 323–328. doi:10.1002/ana.24457

Fromer M, Moran JL, Chambert K, Banks E, Bergen SE, Ruderfer DM, Handsaker RE, McCarroll SA, O'Donovan MC, Owen MJ, et al. 2012. Discovery and statistical genotyping of copy-number variation from whole-exome sequencing depth. *Am J Hum Genet* **91:** 597–607. doi:10.1016/j.ajhg.2012.08.005

Girirajan S, Campbell CD, Eichler EE. 2011. Human copy number variation and complex genetic disease. *Annu Rev Genet* **45:** 203–226. doi:10.1146/annurev-genet-102209-163544

Hong CS, Singh LN, Mullikin JC, Biesecker LG. 2016. Assessing the reproducibility of exome copy number variations predictions. *Genome Med* **8:** 82. doi:10.1186/s13073-016-0336-6

Hunter JD. 2007. Matplotlib: a 2D graphics environment. *Comput Sci Eng* **9:** 90–95. doi:10.1109/MCSE.2007.55

The International HapMap 3 Consortium. 2010. Integrating common and rare genetic variation in diverse human populations. *Nature* **467:** 52–58. doi:10.1038/nature09298

Jiang Y, Oldridge DA, Diskin SJ, Zhang NR. 2015. CODEX: a normalization and copy number variation detection method for whole exome sequencing. *Nucleic Acids Res* **43:** e39. doi:10.1093/nar/gku1363

John E, Adrian F, Jeremy G, Khai L, John L, Geoff O, Paul P, Rank D, Baybayan P, Bettman B, et al. 2009. Real-time DNA sequencing from single polymerase molecules. *Science* **323:** 133–138. doi:10.1126/science.1162986

Kataoka M, Matoba N, Sawada T, Kazuno AA, Ishiwata M, Fujii K, Matsuo K, Takata A, Kato T. 2016. Exome sequencing for bipolar disorder points to roles of *de novo* loss-of-function and protein-altering mutations. *Mol Psychiatry* **21:** 885–893. doi:10.1038/mp.2016.69

Kent WJ, Zweig AS, Barber G, Hinrichs AS, Karolchik D. 2010. BigWig and BigBed: enabling browsing of large distributed datasets. *Bioinformatics* **26:** 2204–2207. doi:10.1093/bioinformatics/btq351

Koboldt DC, Zhang Q, Larson DE, Shen D, McLellan MD, Lin L, Miller CA, Mardis ER, Ding L, Wilson RK. 2012. VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Res* **22:** 568–576. doi:10.1101/gr.129684.111

Koboldt DC, Steinberg KM, Larson DE, Wilson RK, Mardis ER. 2013. The next-generation sequencing revolution and its impact on genomics. *Cell* **155:** 27–38. doi:10.1016/j.cell.2013.09.006

Krumm N, Sudmant P, Ko A. 2012. Copy number variation detection and genotyping from exome sequence data. *Genome Res* **22:** 1525–1532. doi:10.1101/gr.138115.112

Krumm N, O'Roak BJ, Karakoc E, Mohajeri K, Nelson B, Vives L, Jacquemont S, Munson J, Bernier R, Eichler EE. 2013. Transmission disequilibrium of small CNVs in simplex autism. *Am J Hum Genet* **93:** 595–606. doi:10.1016/j.ajhg.2013.07.024

Krumm N, Turner TN, Baker C, Vives L, Mohajeri K, Witherspoon K, Raja A, Coe BP, Stessman HA, He ZX, et al. 2015. Excess of rare, inherited truncating mutations in autism. *Nat Genet* **47:** 582–588. doi:10.1038/ng.3303

Lam ET, Hastie A, Lin C, Ehrlich D, Das SK, Austin MD, Deshpande P, Cao H, Nagarajan N, Xiao M, et al. 2012. Genome mapping on nanochannel arrays for structural variation analysis and sequence assembly. *Nat Biotechnol* **30:** 771–776. doi:10.1038/nbt.2303

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R. 2009. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25:** 2078–2079. doi:10.1093/bioinformatics/btp352

Mak AC, Lai YY, Lam ET, Kwok TP, Leung AK, Poon A, Mostovoy Y, Hastie AR, Stedman W, Anantharaman T, et al. 2016. Genome-wide structural variation detection by genome mapping on nanochannel arrays. *Genetics* **202:** 351–362. doi:10.1534/genetics.115.183483

McCarroll SA, Kuruvilla FG, Korn JM, Cawley S, Nemesh J, Wysoker A, Shapero MH, De Bakker PIW, Maller JB, Kirby A, et al. 2008. Integrated detection and population-genetic analysis of SNPs and copy number variation. *Nat Genet* **40:** 1166–1174. doi:10.1038/ng.238

McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M, et al. 2010. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res* **20:** 1297–1303. doi:10.1101/gr.107524.110

Miller DT, Adam MP, Aradhya S, Biesecker LG, Brothman AR, Carter NP, Church DM, Crolla JA, Eichler EE, Epstein CJ, et al. 2010. Consensus statement: Chromosomal microarray is a first-tier clinical diagnostic test for individuals with developmental disabilities or congenital anomalies. *Am J Hum Genet* **86:** 749–764. doi:10.1016/j.ajhg.2010.04.006

Packer JS, Maxwell EK, O'Dushlaine C, Lopez AE, Dewey FE, Chernomorsky R, Baras A, Overton JD, Habegger L, Reid JG. 2015. CLAMMS: a scalable algorithm for calling common and rare copy number variants from exome sequencing data. *Bioinformatics* **32:** 133–135. doi:10.1093/bioinformatics/btv547

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, et al. 2012. Scikit-learn: machine learning in Python. *J Mach Learn Res* **12:** 2825–2830.

Perry G, Yang F, Marques-Bonet T, Murphy C. 2008. Copy number variation and evolution in humans and chimpanzees. *Genome Res* **18:** 1698–1710. doi:10.1101/gr.082016.108

Poultney CS, Goldberg AP, Drapeau E, Kou Y, Harony-Nicolas H, Kajiwara Y, De Rubeis S, Durand S, Stevens C, Rehnström K, et al. 2013. Identification of small exonic CNV from whole-exome sequence data and application to autism spectrum disorder. *Am J Hum Genet* **93:** 607–619. doi:10.1016/j.ajhg.2013.09.001

Priest JR, Osoegawa K, Mohammed N, Nanda V, Kundu R, Schultz K, Lammer EJ, Girirajan S, Scheetz T, Waggott D, et al. 2016. *De novo* and rare variants at multiple loci support the oligogenic origins of atrioventricular septal heart defects. *PLoS Genet* **12:** e1005963. doi:10.1371/journal.pgen.1005963

Quinlan AR, Hall IM. 2010. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26:** 841–842. doi:10.1093/bioinformatics/btq033

Redon R, Ishikawa S, Fitch KR, Feuk L, Perry GH, Andrews TD, Fiegler H, Shapero MH, Carson AR, Chen W, et al. 2006. Global variation in copy number in the human genome. *Nature* **444:** 444–454. doi:10.1038/nature05329

Retterer K, Juusola J, Cho MT, Vitazka P, Millan F, Gibellini F, Vertino-Bell A, Smaoui N, Neidich J, Monaghan KG, et al. 2016. Clinical application of whole-exome sequencing across clinical indications. *Genet Med* **18:** 696–704. doi:10.1038/gim.2015.148

Sathirapongsasuti JF, Lee H, Horst BA, Brunner G, Cochran AJ, Binder S, Quackenbush J, Nelson SF. 2011. Exome sequencing-based copy-number variation and loss of heterozygosity detection: ExomeCNV. *Bioinformatics* **27:** 2648–2654. doi:10.1093/bioinformatics/btr462

Sebat J, Lakshmi B, Troge J, Alexander J, Young J, Lundin P, Månér S, Massa H, Walker M, Chi M, et al. 2004. Large-scale copy number polymorphism in the human genome. *Science* **305:** 525–528. doi:10.1126/science.1098918

Simons VIP Consortium. 2012. Simons Variation in Individuals Project (Simons VIP): a genetics-first approach to studying autism spectrum and related neurodevelopmental disorders. *Neuron* **73:** 1063–1067. doi:10.1016/j.neuron.2012.02.014

Stark Z, Tan TY, Chong B, Brett GR, Yap P, Walsh M, Yeung A, Peters H, Mordaunt D, Cowie S, et al. 2016. A prospective evaluation of whole-exome sequencing as a first-tier molecular test in infants with suspected monogenic disorders. *Genet Med* **18:** 1090–1096. doi:10.1038/gim.2016.1

Tan R, Wang Y, Kleinstein SE, Liu Y, Zhu X, Guo H, Jiang Q, Allen AS, Zhu M. 2014. An evaluation of copy number variation detection tools from whole-exome sequencing data. *Hum Mutat* **35:** 899–907. doi:10.1002/humu.22537

Tan TY, Dillon OJ, Stark Z, Schofield D, Alam K, Shrestha R, Chong B, Phelan D, Brett GR, Creed E, et al. 2017. Diagnostic impact and cost-effectiveness of whole-exome sequencing for ambulant children with suspected monogenic conditions. *JAMA Pediatr* **171:** 855–862. doi:10.1001/jamapediatrics.2017.1755

Voskoboynik A, Neff NF, Sahoo D, Newman AM, Pushkarev D, Koh W, Passarelli B, Fan HC, Mantalas GL, Palmeri KJ, et al. 2013. The genome sequence of the colonial chordate, *Botryllus schlosseri*. *eLife* **2:** e00569. doi:10.7554/eLife.00569

Wang K, Li M, Hadley D, Liu R, Glessner J, Grant SFA, Hakonarson H, Bucan M. 2007. PennCNV: an integrated hidden Markov model designed for high-resolution copy number variation detection in whole-genome SNP genotyping data. *Genome Res* **17:** 1665–1674. doi:10.1101/gr.6861907

Wickham H. 2016. *ggplot2: elegant graphics for data analysis*. Springer-Verlag, New York.

Yang Y, Muzny DM, Reid JG, Bainbridge MN, Willis A, Ward PA, Braxton A, Beuten J, Xia F, Niu Z, et al. 2013. Clinical whole-exome sequencing for the diagnosis of mendelian disorders. *N Engl J Med* **369:** 1502–1511. doi:10.1056/NEJMoa1306555

Yao R, Zhang C, Yu T, Li N, Hu X, Wang X, Wang J, Shen Y. 2017. Evaluation of three read-depth based CNV detection tools using whole-exome sequencing data. *Mol Cytogenet* **10:** 30. doi:10.1186/s13039-017-0333-5

Zheng GX, Lau BT, Schnall-Levin M, Jarosz M, Bell JM, Hindson CM, Kyriazopoulou-Panagiotopoulou S, Masquelier DA, Merrill L, Terry JM, et al. 2016. Haplotyping germline and cancer genomes with high-throughput linked-read sequencing. *Nat Biotechnol* **34:** 303–311. doi:10.1038/nbt.3432