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], Santhosh Girirajan[1,3,4]

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

Keywords: Copy-number variants, Exome sequencing, Machine learning.

Correspondence to:
Santhosh Girirajan, MBBS, PhD
205A Life Sciences Building
Pennsylvania State University
University Park, PA 16802
Phone: 814-865-0674
E-mail: sxg47@psu.edu

ABSTRACT

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 due to the 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, as it misses potentially true calls that do not have consensus from multiple callers. Here, we present CN-Learn, a machine-learning framework (https://github.com/girirajanlab/CN_Learn) 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, about 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.

INTRODUCTION

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 (Yang et al. 2013; Stark et al. 2016; Tan et al. 2017; Retterer et al. 2016; de Ligt et al. 2013). 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 employed 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 (Backenroth et al. 2014; Jiang et al. 2015; Packer et al. 2015; Krumm et al. 2012; Fromer et al. 2012).

Several themes have emerged due to variations in the approaches employed 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, while 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*, while 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, while CANOES only requires as low 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 >10X across all

samples, while CODEX uses targets that are >20 bp long and with median coverage >20X. CODEX eliminates targets with mappability scores <0.9, while CLAMMS eliminates regions with scores <0.75 along with a custom list of "blacklisted" regions (Packer et al. 2015). *Fourth*, 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), while only a subset of CLAMMS calls with minor allele frequency <0.1 were validated using 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 utilize 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 (Kataoka et al. 2016; Priest et al. 2016; Bademci et al. 2016; Krumm et al. 2015). While using data from multiple callers minimizes false positive rates, this approach discards a large subset of non-concordant 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

106 compared to a Venn-diagram approach. Our study reiterates the limitations of existing CNV
107 detection and integration methods, and offers a better alternative that yields a set of high-quality
108 CNVs for application in clinical diagnostics.
109
110 **RESULTS**
111 We developed CN-Learn as a binary Random Forest classifier that can be trained to differentiate
112 true CNV predictions from false positives using a small subset of validated CNVs (**Figure 1**).
113 We identified twelve features that represented the extent of support from individual algorithms
114 and the genomic context for each CNV (see Methods). We detected statistically significant
115 correlations for several pairs of the quantitative features (**Figure 2; Supplemental Table 1**).
116 Based on a collection of decision trees built using the twelve features extracted from CNVs in
117 the training samples, CN-Learn estimates the probability of each CNV in the test sample to be
118 true (**Figure 2**). In addition to a Random Forest classifier, CN-Learn can also be built as a
119 Logistic Regression (LR) or Support Vector Machine (SVM) based classifier (**Supplemental**
120 **file**).
121
122 **CN-Learn detects high-confidence CNVs with high precision and recall rates**
123 To build the CN-Learn classifier, we first identified 41,791 CNV predictions from 503 samples
124 using four exome-based CNV callers (CANOES, Codex, CLAMMS, and XHMM). Using a read-
125 depth based method (**Supplemental Fig. S1)** to resolve breakpoint conflicts of overlapping CNV
126 predictions obtained from different callers, we identified 29,101 unique CNV events among the
127 503 samples (**Supplemental file**). An alternate approach to resolve breakpoint conflicts
128 (**Supplemental Fig. S2)** also provided the same number of unique CNV events. We selected
129 2,506 of these CNVs from 291 samples with microarray validations that were between 50 kbp
130 and 5 Mbp and spanned regions covered by microarray probes (**Supplemental Fig. S3)**. After
131 determining the proportion of CNVs that overlapped with microarray validations at different
132 thresholds (**Supplemental Table 2**), we labeled each of the selected CNVs as either "True" or
133 "False" based on a 10% reciprocal overlap threshold. We next used CNVs from 70% of the 291
134 samples to train CN-Learn as a Random Forest classifier and the remaining 30% of samples to
135 test its performance. Given the uneven distribution of the labels between the two CNV classes
136 (11% true vs. 89% false), we chose precision and recall as the measures of classifier performance

137 (**Supplemental Table 2**). We captured the aggregate performance of CN-Learn using ten random
138 draws of training data (10-fold cross validation), stratified by sample. CN-Learn used the twelve
139 predictors supplied with each CNV in the test set (see Methods) and classified the calls as either true
140 or false at 91% precision and 86% recall rates. The overall diagnostic ability of the binary classifier,
141 measured as the area under the Receiver Operating Characteristic (ROC) curve, was 95% (**Figure**
142 **3A**). Despite sampling different sets of training data for each iteration during cross-validation, the
143 performance of CN-Learn was consistent across all ten iterations for both precision (91±5%) and
144 recall rates (86±5%) (**Figure 3B**).
145 We also assessed the relative importance of each feature towards making accurate CNV
146 predictions by calculating the Gini index (defined as the total decrease in improvements to the node
147 purity for all splits on each feature averaged over all trees in the forest) (Breiman 2001). We found
148 that features such as the number of exome capture probes spanning a given call, the extent to which
149 CANOES agreed with a CNV prediction, concordance among the callers, and GC content provided
150 the most discriminatory power to the classifier (**Figure 3C**). Post-classification analysis of the
151 concordance profile indicated that only 34% of all CNVs classified as true had support from all the
152 four callers, while the remaining 66% lacked support from at least one caller (**Supplemental Fig.**
153 **S4**). We further assessed the performance of CN-Learn on an independent set of 90 samples from
154 the 1000 Genomes project (Jiang et al. 2015), and observed a precision rate of 93% and recall rate
155 of 86% when using 70% training data (**Supplemental Fig. S5-S6**). When we generated
156 precision/recall rates that are directly comparable with the concordance-based methods, we
157 observed a consistently superior performance for CN-Learn (**Supplemental File, Supplemental**
158 **Tables 3-4**, **Supplemental Fig. S7)**. Overall, these results highlight the ability of CN-Learn to look
159 beyond the single measure of concordance typically used in a Venn-diagram based approach, and to
160 utilize the discriminatory power of additional variables to identify high-confidence CNVs in a
161 systematic manner.
162
163 **Performance of CN-Learn is robust across varying CNV sizes, frequencies and training sets**
164 We independently trained CN-Learn using varying proportions of training data (between 10% and
165 70% in increments of 10%) and observed steady performance gains with increase in the number of
166 training samples (**Figure 3A**). Interestingly, even when the classifier was built using just 10% of the
167 total samples (n=29 samples), we obtained 90% precision and a recall rate of 75%, indicating the

168 robustness of the classifier when learning from minimal training data. We further trained CN-Learn
169 independently at four size ranges of CNVs and observed a modest increase in precision with
170 increase in CNV size (90% for 50-100 kbp CNV to 97% for 0.5-5 Mbp CNV) (**Figure 3D**). In fact,
171 the precision achieved by CN-Learn at each size interval was substantially higher than the precision
172 achieved by the individual CNV callers. We also observed comparable precision and recall rates
173 when CN-Learn was run on breakpoint-resolved CNVs obtained by merging overlapping
174 predictions (**Supplemental Fig. S8**). We then tested the performance of CN-Learn for different
175 classes of CNVs based on their frequency in the cohort (very rare, rare, common, very common),
176 and found precision (> 90%) and recall (> 80%) rates that were consistent across the CNV
177 frequency spectrum **(Supplemental File; Supplemental Fig. S9)**. In fact, when added as an
178 additional predictor to CN-Learn, CNV frequency showed the highest discriminatory power (16%)
179 of any predictor, but did not contribute to significant improvements in the performance of the
180 classifier (**Supplemental Fig. S10, S11)**. While the performance of the Random Forest classifier
181 was robust, Logistic Regression and SVM classifiers failed to match its performance
182 (**Supplemental File**; **Supplemental Figs. S12, S13**).
183 We next assessed the performance of CN-Learn by considering calls made by CLAMMS as
184 the truth set and classified CNVs obtained from predictions made by the other three callers
185 (CANOES, CODEX and XHMM). We categorized 25,019 breakpoint-resolved CNVs
186 (**Supplemental Fig. S3**) from all 503 samples as either "True" or "False" based on their intersection
187 (10% reciprocal overlap) with CNVs predicted by CLAMMS. Given the higher resolution of CNVs
188 detected from exome data relative to SNP microarrays, we used a total of 16,497 CNVs between 5
189 kbp and 5 Mbp in size to build CN-Learn (**Supplemental Fig. S3**). CN-Learn achieved an aggregate
190 precision rate of 94% with an overall recall rate of 85% during the 10-fold cross validation, and
191 achieved comparable performance when independently trained with CNVs at different training
192 sample proportions (**Supplemental Fig. S14A**). Performance variability observed both during cross-
193 validation and across the size ranges were comparable to the variability observed when microarray
194 was used as the truth set (**Supplemental Figs. S14B-C**). Among the features used by CN-Learn to
195 classify the CNVs, the relative importance of the mappability score was the highest, with GC
196 content being the next important feature (**Supplemental Fig. S14C**). While caller-specific features
197 contributed to the discriminatory power of CNV classification when microarray was used for
198 validation, genomic features played a more prominent role when a sequence-based method was used

199 for validation (**Supplemental Fig. S15**). These results show that the performance of CN-Learn is
200 robust with minimal training data, at different size ranges and even when orthogonal validations are
201 not available.
202
203 **CN-Learn recovers true CNVs that lack complete concordance among callers**
204 To assess the ability of CN-Learn to correctly identify true CNVs that lack support from multiple
205 callers, we analyzed the concordance profile of all CNVs classified by CN-Learn as true, based on
206 microarray validations, before and after CN-Learn classification. Using a single random draw of 29
207 samples (10%) to train CN-Learn and 262 samples as the test set, we obtained predictions for 2,245
208 CNVs with microarray validations (**Figure 4A)**. Among these predictions, only 122/2,245 CNVs
209 (5.4%) were supported by all four CNV callers prior to classification by CN-Learn. The strong
210 concordance of the four methods for these CNV predictions was corroborated by a high microarray
211 validation rate (116/122, 95%) (**Figure 4A**). In contrast, CNVs that lacked support from one or
212 more callers were less likely to intersect with microarray data. For example, only 41% (11/27) of the
213 CNVs with support from XHMM, CODEX and CLAMMS intersected with microarray data. After
214 classification by CN-Learn, 84% (266/315) of all CNVs labeled as "True" intersected with
215 microarray calls, indicating the high classification accuracy achieved by the classifier (**Figure 4A**).
216 All CNVs classified as true and false from this analysis are provided in **Supplemental Tables 5 and**
217 **6**. Of the 266 CNVs validated by microarrays and correctly identified as true by the classifier, only
218 42% (112/266) were supported by all four CNV callers. The remaining 58% (154/266) were either
219 singletons or calls that lacked support from at least one caller, and would have been excluded if
220 complete concordance was used as the only determinant for selecting high confidence CNVs. For
221 example, 13% of all true CNVs (35/266) recovered by CN-Learn were missed by CLAMMS but
222 were identified by one or more of the other callers, reiterating the limitations of using a single
223 exome-based CNV caller for variant predictions. Further, CN-Learn managed to recover 97%
224 (112/116) of the true CNVs validated by microarrays that were supported by all of the four callers.
225 Although these 112 CNVs could have been identified by a simple caller intersection approach, CN-
226 Learn was uniquely able to recover CNVs that lacked support from at least one other caller. For
227 example, CN-Learn classified 66 CNVs supported by CANOES, CODEX and XHMM as true, of
228 which 51% (34/66) were also validated by microarrays. This result is notable because, without using
229 CN-Learn for classification, only 31% (46/148) of CNVs supported by CANOES, CODEX and

230 XHMM would intersect with the microarray calls, indicating the inherently high false positive rate
231 associated with simply intersecting calls from individual callers using a Venn diagram. Our results
232 indicate that in addition to correctly identifying almost every true CNV reported by the four callers,
233 CN-Learn overcame the limitations of the Venn-diagram based approach and recovered 154
234 additional high-confidence CNVs with sub-optimal concordance, thereby improving the CNV yield
235 by 2.37-fold (266/112). Further, the true positive calls recovered by CN-Learn spanned the entire
236 spectrum of CNV size range (**Figure 4B**). We also obtained comparable improvements in CNV
237 yield when CNVs predicted by CLAMMS were used as the truth set (**Supplemental file**;
238 **Supplemental Figs. S16-S17, Supplemental Tables 5 and 6**).
239
240 **DISCUSSION**
241 Exome sequencing is a cost-effective assay to accurately identify single base pair mutations and
242 small insertions or deletions within protein-coding genes, which are more likely to cause disease
243 than mutations in other regions of the genome (Bamshad et al. 2011). Even with the emergence of
244 whole genome sequencing (WGS) techniques, exome sequencing has continued to flourish,
245 resulting in an increase in the availability of datasets for analysis. This widespread data availability
246 has allowed for repurposing data generated for single nucleotide variant detection to identify CNVs
247 as well (Koboldt et al. 2013). Given the high false positive rates among calls reported by individual
248 exome-based callers and platform-specific biases (Tan et al. 2014), existing CNV identification
249 pipelines that leverage more than one calling algorithm have depended on naïve Venn diagram-
250 based integration approaches to identify high-confidence CNVs. One reason for the overwhelming
251 trust in such approaches could be the fact that the self-reported performance measures of each
252 individual algorithm are typically high. Therefore, there is little reason to doubt that performance
253 variations among the callers could affect the precision of Venn diagram-based integration
254 approaches. While we found that the likelihood of a CNV prediction to be true increased with an
255 increase in the number of callers supporting it (**Supplemental Fig. S18**), there are two key
256 limitations of this approach. *First*, even among completely concordant predictions, the observed
257 false positive rates were not zero. Inflated false positives pose a large hurdle for researchers
258 interested in identifying clinically relevant CNVs, as it is time consuming to validate a large number
259 of false positive CNVs using orthogonal methods. *Second*, the Venn diagram approach failed to
260 identify a large subset of non-concordant or singleton CNVs supported by microarray validations. In

261 fact, approximately one true CNV (>50 kbp) per individual in our cohort would have been discarded
262 under a Venn-diagram based approach. This evidence reiterates that Venn diagram-based
263 approaches do not have the required precision for usage in both clinical and research settings.
264 The utility of alternate methods for CNV detection hence rests on the ability to both
265 eliminate false positives among completely concordant predictions and recover true CNVs that lack
266 adequate support from multiple callers. Therefore, instead of addressing the shortcomings of
267 existing methods by developing yet another CNV detection tool, our study focused on offering a
268 reliable integrative approach. In this study, we demonstrate a machine-learning approach that
269 leverages caller-specific and genomic contexts from a subset of validated calls to identify high-
270 confidence CNVs more thoroughly than individual callers on their own or Venn diagram-based
271 approaches. CN-Learn achieved precision as high as 94% while doubling the CNV yield, showing
272 its ability to capture singletons that would have been missed by other approaches. Importantly, the
273 precision of CN-Learn is robust to variation in CNV size, ratio of training to testing samples, and
274 validation method/type, indicating the utility of CN-Learn in a variety of clinical or research
275 contexts. The use of multiple variables that capture the genomic context of each CNV in addition to
276 caller concordance is a major reason for the high recovery and precision rates achieved by CN-
277 Learn. In fact, GC content and mappability scores were some of the most useful features in
278 predicting true CNVs.
279 One of the limitations of CN-Learn is its dependency on a small set of biologically validated
280 CNVs. Our study leverages microarray validations to evaluate the predictions of each caller and to
281 label the breakpoint-resolved CNV predictions as either true or false. While microarrays may not be
282 considered as a "gold standard" for CNV detection, we were able to use microarray calls as an
283 orthogonal validation to demonstrate the utility of CN-Learn. Our study also serves as a proof-of-
284 principle for future studies that could utilize CN-Learn with "gold-standard" CNVs curated from
285 multiple genomic technologies, such as PacBio SMRT sequencing (John et al. 2009), Illumina long-
286 read sequencing (Voskoboynik et al. 2013), 10X linked-read sequencing (Zheng et al. 2016),
287 BioNano Genomics genome mapping (Lam et al. 2012; Mak et al. 2016) or PCR. Another limitation
288 of our study is the time and computational capacity required to run four different CNV callers. Each
289 CNV calling algorithm extracts read-depth information for each sample using tools such as
290 Samtools (Li et al. 2009), Bedtools (Quinlan and Hall 2010) or Genome Analysis Toolkit (GATK)
291 (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. While we presented two strategies to resolve breakpoints of concordant CNVs (see Methods), future studies could explore more effective strategies before using CN-Learn. As population-scale projects continue to generate large exome-sequencing datasets, 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.

309 **METHODS**

310 **Samples**

311 We obtained exome sequencing data from 503 individuals in the Simons Variation in Individual
312 Project (SVIP) which were generated using the Agilent Sure-Select Human All Exon v2.0 capture
313 kit containing 182,430 autosomal probes (Spiro and Chung 2012). Overall, 187 individuals within
314 this cohort carry the 16p11.2 deletion, and 143 individuals carry the reciprocal duplication. Single
315 nucleotide polymorphism (SNP) based microarray data were available for 291 samples, and CNVs
316 >100 kbp were confirmed experimentally using array CGH (Duyzend et al. 2016). We also used 90
317 samples from the 1000 Genomes project (Durbin et al. 2010), of which 46 samples were sequenced
318 at the Washington University Genome Sequencing Center using the Agilent SureSelect All Exon
319 v2.0 capture kit and 44 samples were sequenced at Baylor College of Medicine using the Roche
320 HGSC VCRome capture kit. We used 186,065 autosomal probes from the consensus probe set
321 (ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/reference/exome_pull_down_targets/20130108.e
322 xome.targets.bed) to make CNV predictions.

323

324 **Exome CNV callers**

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

336

337 **Resolving breakpoint conflicts for overlapping predictions**

338 Multiple CNV callers can make predictions that overlap with each other in a given genomic region.
339 Treating such overlapping predictions with different breakpoints as separate CNVs would result in

340 double-counting the calls for the same CNV event. Therefore, it is important to merge concordant
341 predictions and represent them as a single event for downstream analyses. We developed a five-step
342 procedure that uses fluctuations in local read depth to resolve breakpoint conflicts among
343 overlapping predictions by identifying the most likely start and end coordinates of the underlying
344 event**.** A detailed explanation of this strategy is presented in the **Supplemental file (Supplemental**
345 **Figs. S1 & S20)**. As an alternate strategy, we resolved breakpoint conflicts by simply selecting the
346 smallest and largest coordinates among the overlapping predictions as the start and end coordinates
347 of the underlying CNV event. This strategy is also described in the **Supplemental file**
348 **(Supplemental Fig. S2)**. Applying both methods to the 41,791 CNVs detected in the SVIP samples
349 yielded 8,382 unique breakpoint-resolved CNV events, in addition to the 20,719 singleton calls
350 (**Supplemental Fig. S3**).
351
352 **CNV validations**
353 SNP microarray data were generated by the SVIP consortium (Duyzend et al. 2016) using Illumina
354 OmniExpress-12 microarrays for 272 samples and OmniExpress-14 for 19 samples. PennCNV
355 (v.1.0.3) was used to identify CNVs from microarray data for all 291 individuals using standard
356 parameters (Wang et al. 2007). Individual and family-based (trios and quads) CNV calls were
357 combined for autosomal chromosomes, while CNVs on chromosome X were called only at the
358 individual level. CNV calls with ≥1 bp overlap or gaps <20% of the total CNV length and <50 kbp
359 were merged. CNVs ≥50 kbp in length and containing ≥5 SNP target probes were subsequently
360 considered for further analyses.
361 For the 1000 Genomes samples, we pooled the validated calls from three sources (Altshuler
362 et al. 2010; Conrad et al. 2012; McCarroll et al. 2008) that were used to measure the performance of
363 CODEX (Jiang et al. 2015). Merging the overlaps among the 10,235 CNVs resulted in 7,302 CNVs
364 (**Supplemental Table 4**) that were used to label the CNVs predicted for the 90 samples from the
365 1000 Genomes project.
366
367 **Feature selection for CN-Learn classifier**
368 We identified twelve features that represent the extent of support provided by the four individual
369 callers and genomic context as predictors of true CNVs to the CN-Learn Random Forest classifier.
370 Since we used four callers in our study, the extent of overlap calculated during the breakpoint-

371 resolution process (**Supplemental file** & **Supplemental Fig. S1**) served as individual predictors.
372 Concordance count and read-depth ratio (RD_{ratio}) for both breakpoint-resolved concordant calls and
373 singletons were also supplied as features. As individual algorithms use different GC and repeat
374 content (mappability) thresholds to classify CNV predictions as outliers, CNVs with extreme GC
375 content or low mappability could be predicted by one caller but discarded as an outlier by the other
376 callers. To take this into account, we extracted GC content data using the "nuc" option in bedtools
377 (Quinlan and Hall 2010) and mappability scores (Derrien et al. 2012) using the
378 "bigWigAverageOverBed" option in kentUtils (Kent et al. 2010) for use as predictors. Similarly, the
379 efficacy of CNV detection can vary across chromosomes, size ranges, and CNV type
380 (duplication/deletion). To take these variations into account, we used chromosome number, CNV
381 size, CNV type and the number of exome capture probes as the final set of CN-Learn features.
382
383 **Probability estimation and classification using CN-Learn**
384 CN-Learn leverages the features extracted for the CNVs in the training samples to build a Random
385 Forest (Breiman 2001) classifier with hundreds of decision trees and estimates the probability of
386 each CNV in the test samples to be true (**Figure 2A**). Decision trees have been shown to perform
387 well when the distribution of observations is unbalanced between the classes of interest (Cieslak and
388 Chawla 2008). Given the high false positive rates of CNV callers, an uneven split between the
389 number of true and false predictions is likely to occur in clinical samples. In the samples we
390 analyzed, 10% of the CNVs overlapped with the truth sets derived from microarray calls at a 10%
391 reciprocal overlap threshold (**Supplemental Table 2**). To address the imbalance between the
392 number of true and false CNVs, we stratified the training data by sample to accurately reflect the
393 clinical setting as follows: CNVs in p% of the samples were used for training, and the remaining (1-
394 p)% were used as the testing set. For a random forest built with "N" trees, "M" predictors and "C"
395 classes, the probability of an observation "o" belonging to the class "c" (true or false) can be
396 expressed as $p_{o,c} = Pr(Y = c \mid X = x_i)$, where x_i is a vector that captures the values for each of the 12
397 predictors and Y is the outcome variable. The probability $p_{o,true}$ of the CNV "o" being true in the test
398 set was then measured as the proportion of trees in the forest that assigned it to the true class.
399 Specifically, the probability of a CNV prediction can be represented as:

400 $$P_{o,true} = Pr(Y = True \mid X = x_i)$$

401 $$= \frac{\text{Number of trees that identify the CNV as True}}{\text{Number of trees in the forest}}$$

402 where X represents the values of the following features for the observation "o":

403 x_1 = Overlap proportion with CANOES

404 x_2 = Overlap proportion with CODEX

405 x_3 = Overlap proportion with CLAMMS

406 x_4 = Overlap proportion with XHMM

407 x_5 = Concordance among callers

408 x_6 = Read depth ratio

409 x_7 = Chromosome number

410 x_8 = CNV type (Duplication/Deletion)

411 x_9 = CNV size

412 x_{10} = Target probe count

413 x_{11} = GC content

414 x_{12} = Mappability

415

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

422

423 **Statistical analysis**

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

427

428 **Code availability**

429 CN-Learn is available as an open-source software at *https://github.com/girirajanlab/CN_Learn* and
430 provided as a supplemental file (**Supplemental_Code.zip**). In addition to the scripts necessary to
431 run CN-Learn, we also provide simplified and easily parallelizable pipelines for each of the original
432 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 the supplemental file (**Supplemental_Code.zip**) and at the landing page for the software on GitHub.

440 **DECLARATIONS**

441 **Ethics approval and consent to participate**

442 As these data were de-identified, all of our samples were exempt from IRB review and conformed to
443 the Helsinki Declaration. No other approvals were needed for the study.

444

445 **Consent for publication**

446 All authors agree and consent for publication of the manuscript.

447

448 **Availability of data and materials**

449 BAM files for all samples, aligned to the hg19 reference genome, were obtained from the Simons
450 Foundation Autism Research Initiative (SFARI) via SFARI Base following appropriate approvals
451 (*https://www.sfari.org/resource/sfari-base/*) and from the ftp site of the 1000 Genomes data
452 collection (ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase3/data). Realigning the reads to hg38
453 (GRCh38) would not significantly affect the conclusions of the manuscript, as the CNV predictions
454 are restricted to sequences captured by the exome capture probes. The script used to generate the
455 plots are provided in the supplemental file (**Supplemental_Code.zip**) and in the "research"
456 directory of CN-Learn repository. Any intermediate datasets generated during this study are
457 available from the corresponding author upon request.

458

459 **Competing interests**

460 The authors declare that no competing interests exist in relation to this work.

461

462 **Funding**

463 This work was supported by R01-MH107431, R01-GM121907, SFARI Pilot Grant 399894, and
464 resources from the Huck Institutes of the Life Sciences (to SG). This work was funded partly by the
465 Big Data to Knowledge (BD2K) pre-doctoral training program (T32LM012415) from the National
466 Institutes of Health (to VK). The funding bodies had no role in data collection, analysis, and
467 interpretation.

468

Authors' contributions

VK and SG conceived the project. VK performed the analyses, generated the plots/images, and wrote and revised the manuscript; MJ organized the images for publication and assisted with revision to the manuscript; GJ and NK collected data and performed parts of the analyses; SG supervised the research and wrote and revised the manuscript. All authors read and approved the final draft of the manuscript.

Acknowledgements

We thank Naomi Altman (Penn State), Dajiang Liu (Penn State), Arjun Krishnan (Michigan State), Cooduvalli Shashikant (Penn State), Lucilla Pizzo (Penn State), and Shaun Mahony (Penn State) for constructive comments on the manuscript.

481 **REFERENCES**

482 Altshuler DM, Gibbs RA, Peltonen L, Schaffner SF, Yu F, Dermitzakis E, Bonnen PE, De Bakker
483 PIW, Deloukas P, Gabriel SB, et al. 2010. Integrating common and rare genetic variation in
484 diverse human populations. *Nature* **467**: 52–58.

485 Backenroth D, Homsy J, Murillo LR, Glessner J, Lin E, Brueckner M, Lifton R, Goldmuntz E,
486 Chung WK, Shen Y. 2014. CANOES: Detecting rare copy number variants from whole exome
487 sequencing data. *Nucleic Acids Res* **42**: 1–9.

488 Bademci G, Mahdieh N, Bonyadi M, Duman D, Cengiz FB, Menendez I, Horta OD, Shirkavand A,
489 Zeinali S, Subasioglu A, et al. 2016. Comprehensive Analysis via Exome Sequencing
490 Uncovers Genetic Etiology in Autosomal Recessive Non-Syndromic Deafness in a Large
491 Multiethnic Cohort. *Genet Med* **18**: 364–371.

492 Bamshad MJ, Ng SB, Bigham AW, Tabor HK, Emond MJ, Nickerson DA, Shendure J. 2011.
493 Exome sequencing as a tool for Mendelian disease gene discovery. *Nat Rev Genet* **12**: 745–
494 755.

495 Breiman L. 2001. Random forests. *Mach Learn* **45**: 5–32.

496 Cieslak DA, Chawla N V. 2008. Learning Decision Trees for Unbalanced Data. *Mach Learn Knowl*
497 *Discov Databases* **5211**: 241–256.

498 Conrad DF, Pinto D, Redon R, Feuk L, Gokcumen O, Zhang Y, Aerts J, Andrews TD, Barnes C,
499 Campbell P, et al. 2012. Origins and functional impact of copy number variation in the human
500 genome. *Nature* **464**: 704–712.

501 Coughlin CR 2nd, Scharer GH, Shaikh TH. 2012. Clinical impact of copy number variation analysis
502 using high-resolution microarray technologies: advantages, limitations and concerns. *Genome*
503 *Med* **4**: 80.

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

507 Derrien T, Estellé J, Sola SM, Knowles DG, Raineri E, Guigó R, Ribeca P. 2012. Fast computation
508 and applications of genome mappability. *PLoS One* **7**.

509 Durbin RM, Altshuler DL, Durbin RM, Abecasis GR, Bentley DR, Chakravarti A, Clark AG,
510 Collins FS, De La Vega FM, Donnelly P, et al. 2010. A map of human genome variation from
511 population-scale sequencing. *Nature* **467**: 1061–1073.

512 Duyzend MH, Nuttle X, Coe BP, Baker C, Nickerson DA, Bernier R, Eichler EE. 2016. Maternal
513 Modifiers and Parent-of-Origin Bias of the Autism-Associated 16p11.2 CNV. *Am J Hum Genet*
514 **98**: 45–57.
515 Epilepsy Phenome/Genome Project & Epi4K Consortium. 2015. Copy number variant analysis from
516 exome data in 349 patients with epileptic encephalopathy. *Ann Neurol* **78**: 323–328.
517 Fromer M, Moran JL, Chambert K, Banks E, Bergen SE, Ruderfer DM, Handsaker RE, McCarroll
518 SA, O'Donovan MC, Owen MJ, et al. 2012. Discovery and statistical genotyping of copy-
519 number variation from whole-exome sequencing depth. *Am J Hum Genet* **91**: 597–607.
520 Girirajan S, Campbell CD, Eichler EE. 2011. Human Copy Number Variation and Complex Genetic
521 Disease. *Annu Rev Genet* **45**: 203–226.
522 Hong CS, Singh LN, Mullikin JC, Biesecker LG. 2016. Assessing the reproducibility of exome copy
523 number variations predictions. *Genome Med* **8**: 1–11.
524 Jiang Y, Oldridge DA, Diskin SJ, Zhang NR. 2015. CODEX: A normalization and copy number
525 variation detection method for whole exome sequencing. *Nucleic Acids Res* **43**: e39.
526 John E, Adrian F, Jeremy G, Khai L, John L, Geoff O, Paul P, Rank D, Baybayan P, Bettman B, et
527 al. 2009. Real-Time DNA Sequencing from Single Polymerase Molecules. *Science (80-)* **323**:
528 133–138.
529 Kataoka M, Matoba N, Sawada T, Kazuno AA, Ishiwata M, Fujii K, Matsuo K, Takata A, Kato T.
530 2016. Exome sequencing for bipolar disorder points to roles of de novo loss-of-function and
531 protein-altering mutations. *Mol Psychiatry* **21**: 885–893.
532 Kent WJ, Zweig AS, Barber G, Hinrichs AS, Karolchik D. 2010. BigWig and BigBed: Enabling
533 browsing of large distributed datasets. *Bioinformatics* **26**: 2204–2207.
534 Koboldt DC, Steinberg KM, Larson DE, Wilson RK, Mardis ER. 2013. The next-generation
535 sequencing revolution and its impact on genomics. *Cell* **155**: 27–38.
536 Koboldt DC, Zhang Q, Larson DE, Shen D, McLellan MD, Lin L, Miller CA, Mardis ER, Ding L,
537 Wilson RK. 2012. VarScan 2: Somatic mutation and copy number alteration discovery in
538 cancer by exome sequencing. *Genome Res* **22**: 568–576.
539 Krumm N, O'Roak BJ, Karakoc E, Mohajeri K, Nelson B, Vives L, Jacquemont S, Munson J,
540 Bernier R, Eichler EE. 2013. Transmission disequilibrium of small CNVs in simplex autism.
541 *Am J Hum Genet* **93**: 595–606.
542 Krumm N, Sudmant P, Ko A. 2012. Copy number variation detection and genotyping from exome

543 sequence data. *Genome Res* **22**: 1525–1532.
544 Krumm N, Turner TN, Baker C, Vives L, Mohajeri K, Witherspoon K, Raja A, Coe BP, Stessman
545 HA, He ZX, et al. 2015. Excess of rare, inherited truncating mutations in autism. *Nat Genet* **47**:
546 582–588.
547 Lam ET, Hastie A, Lin C, Ehrlich D, Das SK, Austin MD, Deshpande P, Cao H, Nagarajan N, Xiao
548 M, et al. 2012. Genome mapping on nanochannel arrays for structural variation analysis and
549 sequence assembly. *Nat Biotechnol* **30**: 771–776.
550 Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R.
551 2009. The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**: 2078–2079.
552 Mak ACY, Lai YYY, Lam ET, Kwok TP, Leung AKY, Poon A, Mostovoy Y, Hastie AR, Stedman
553 W, Anantharaman T, et al. 2016. Genome-wide structural variation detection by genome
554 mapping on nanochannel arrays. *Genetics* **202**: 351–362.
555 McCarroll SA, Kuruvilla FG, Korn JM, Cawley S, Nemesh J, Wysoker A, Shapero MH, De Bakker
556 PIW, Maller JB, Kirby A, et al. 2008. Integrated detection and population-genetic analysis of
557 SNPs and copy number variation. *Nat Genet* **40**: 1166–1174.
558 McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler
559 D, Gabriel S, Daly M, et al. 2010. The Genome Analysis Toolkit: A MapReduce framework
560 for analyzing next-generation DNA sequencing data. *Genome Res* **20**: 254–260.
561 Miller DT, Adam MP, Aradhya S, Biesecker LG, Brothman AR, Carter NP, Church DM, Crolla JA,
562 Eichler EE, Epstein CJ, et al. 2010. Consensus Statement: Chromosomal Microarray Is a First-
563 Tier Clinical Diagnostic Test for Individuals with Developmental Disabilities or Congenital
564 Anomalies. *Am J Hum Genet* **86**: 749–764.
565 Packer JS, Maxwell EK, O'Dushlaine C, Lopez AE, Dewey FE, Chernomorsky R, Baras A, Overton
566 JD, Habegger L, Reid JG. 2015. CLAMMS: A scalable algorithm for calling common and rare
567 copy number variants from exome sequencing data. *Bioinformatics* **32**: 133–135.
568 Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P,
569 Weiss R, Dubourg V, et al. 2012. Scikit-learn: Machine Learning in Python. *J Mach Learn Res*
570 **12**: 2825–2830.
571 Perry G, Yang F, Marques-Bonet T, Murphy C. 2008. Copy number variation and evolution in
572 humans and chimpanzees. *Genome Res* 1698–1710.
573 Poultney CS, Goldberg AP, Drapeau E, Kou Y, Harony-Nicolas H, Kajiwara Y, De Rubeis S,

574 Durand S, Stevens C, Rehnström K, et al. 2013. Identification of small exonic CNV from
575 whole-exome sequence data and application to autism spectrum disorder. *Am J Hum Genet* **93**:
576 607–619.
577 Priest JR, Osoegawa K, Mohammed N, Nanda V, Kundu R, Schultz K, Lammer EJ, Girirajan S,
578 Scheetz T, Waggott D, et al. 2016. De Novo and Rare Variants at Multiple Loci Support the
579 Oligogenic Origins of Atrioventricular Septal Heart Defects. *PLoS Genet* **12**: 1–25.
580 Quinlan AR, Hall IM. 2010. BEDTools: A flexible suite of utilities for comparing genomic features.
581 *Bioinformatics* **26**: 841–842.
582 Redon R, Ishikawa S, Fitch KR, Feuk L, Perry GH, Andrews TD, Fiegler H, Shapero MH, Carson
583 AR, Chen W, et al. 2006. Global variation in copy number in the human genome. *Nature* **444**:
584 444–454.
585 Retterer K, Juusola J, Cho MT, Vitazka P, Millan F, Gibellini F, Vertino-Bell A, Smaoui N, Neidich
586 J, Monaghan KG, et al. 2016. Clinical application of whole-exome sequencing across clinical
587 indications. *Genet Med* **18**: 696–704.
588 Sathirapongsasuti JF, Lee H, Horst BAJ, Brunner G, Cochran AJ, Binder S, Quackenbush J, Nelson
589 SF. 2011. Exome sequencing-based copy-number variation and loss of heterozygosity
590 detection: ExomeCNV. *Bioinformatics* **27**: 2648–2654.
591 Sebat J, Lakshmi B, Troge J, Alexander J, Young J, Lundin P, Månér S, Massa H, Walker M, Chi
592 M, et al. 2004. Large-scale copy number polymorphism in the human genome. *Science (80-)*
593 **305**: 525–8.
594 Spiro JE, Chung WK. 2012. Simons Variation in Individuals Project (Simons VIP): A Genetics-
595 First Approach to Studying Autism Spectrum and Related Neurodevelopmental Disorders.
596 *Neuron* **73**: 1063–1067.
597 Stark Z, Tan TY, Chong B, Brett GR, Yap P, Walsh M, Yeung A, Peters H, Mordaunt D, Cowie S,
598 et al. 2016. A prospective evaluation of whole-exome sequencing as a first-tier molecular test
599 in infants with suspected monogenic disorders. *Genet Med* **18**: 1090–1096.
600 Tan R, Wang Y, Kleinstein SE, Liu Y, Zhu X, Guo H, Jiang Q, Allen AS, Zhu M. 2014. An
601 Evaluation of Copy Number Variation Detection Tools from Whole-Exome Sequencing Data.
602 *Hum Mutat* **35**: 899–907.
603 Tan TY, Dillon OJ, Stark Z, Schofield D, Alam K, Shrestha R, Chong B, Phelan D, Brett GR, Creed
604 E, et al. 2017. Diagnostic impact and cost-effectiveness of whole-exome sequencing for

605 ambulant children with suspected monogenic conditions. *JAMA Pediatr* **171**: 855–862.
606 Voskoboynik A, Neff NF, Sahoo D, Newman AM, Pushkarev D, Koh W, Passarelli B, Fan HC,
607 Mantalas GL, Palmeri KJ, et al. 2013. The genome sequence of the colonial chordate, Botryllus
608 schlosseri. *Elife* **2013**: 1–24.
609 Wang K, Li M, Hadley D, Liu R, Glessner J, Grant SFA, Hakonarson H, Bucan M. 2007.
610 PennCNV : An integrated hidden Markov model designed for high-resolution copy number
611 variation detection in whole-genome SNP genotyping data PennCNV : An integrated hidden
612 Markov model designed for high-resolution copy number variation detection in whole-.
613 *Genome Res* 1665–1674.
614 Yang Y, Muzny DM, Reid JG, Bainbridge MN, Willis A, Ward PA, Braxton A, Beuten J, Xia F,
615 Niu Z, et al. 2013. Clinical Whole-Exome Sequencing for the Diagnosis of Mendelian
616 Disorders. *N Engl J Med* **369**: 1502–1511.
617 Yao R, Zhang C, Yu T, Li N, Hu X, Wang X, Wang J, Shen Y. 2017. Evaluation of three read-depth
618 based CNV detection tools using whole-exome sequencing data. *Mol Cytogenet* **10**: 1–7.
619 Zheng GXY, Lau BT, Schnall-Levin M, Jarosz M, Bell JM, Hindson CM, Kyriazopoulou-
620 Panagiotopoulou S, Masquelier DA, Merrill L, Terry JM, et al. 2016. Haplotyping germline
621 and cancer genomes with high-throughput linked-read sequencing. *Nat Biotechnol* **34**: 303–
622 311.
623 SFARI Base. *https://www.sfari.org/resource/sfari-base/*.
624
625

626 **FIGURE LEGENDS**

627

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

639

640 **Figure 2: Illustration of the Random Forest model used to build CN-Learn. (A)** The inner
641 workings of the random forest model used for training CN-Learn is shown. Twelve features were
642 used to grow 100 trees with different subsamples of predictors and training data to classify each
643 CNV in the test set as either true or false. If the predicted probability score was greater than 0.5, the
644 CNV call was classified as true. Calls with predicted probability score <0.5 were labeled as false.
645 (**B)** A Spearman rank correlation between pairs of quantitative predictors used by the CN-Learn
646 classifier is shown. The color of the circles indicates the direction of the correlation, while the size
647 of the circles indicates the strength of the correlation. The correlation scores are provided in the
648 supplemental file (**Supplemental Table 1**). (**C)** The frequency of microarray validated and
649 invalidated CNVs, distributed across 20 bins of increasing predicted probability scores, is shown.
650 For the probability bins less than 0.5, the proportion of CNVs that were validated was higher than
651 the proportion of CNVs that was not validated. This indicated that classification score of 0.5 is an
652 appropriate threshold for distinguishing true and false CNVs.

653

654 **Figure 3: Characteristics of the CN-Learn binary Random Forest classifier. (A)** Receiver
655 operating characteristic (ROC) curves indicating the trade-off between the precision and recall rates
656 when CN-Learn was trained as a Random Forest classifier are shown. Each curve represents the

657 performance achieved when using different proportions of samples to train CN-Learn, starting from
658 10% up to 70% in increments of 10%. The results shown were from experiments aggregated across
659 10-fold cross validation. (**B)** Variability observed in the precision and recall measures during the 10-
660 fold cross validation at various proportions of training data is shown. Both measures varied within
661 ±5% of their corresponding averages. (**C)** The relative importance of each genomic and caller-
662 specific feature supplemented to CN-Learn is shown. Data shown here are the averages of the
663 values obtained across 10-fold cross-validation after using 70% of the samples for training. **(D)**
664 Precision rates for CNVs when CN-Learn was trained at four different size ranges compared to the
665 precision rates of CNVs from individual callers are shown. Precision rates for CN-Learn were
666 estimated as its classification accuracy (True positives/ [True positives + False positives]), while the
667 precision rates for the individual callers were calculated as the proportion of CNVs at each size
668 range that were validated by the microarray calls.
669
670 **Figure 4: Concordance profile of CNVs before and after classification by CN-Learn using**
671 **microarray calls as the "gold standard" validation set. (A)** Venn diagrams are shown for CNVs
672 (≥50 kbp) identified from a random draw of 262 samples out of the total 291 samples before (**Top**
673 **panels**) and after (**Bottom panels**) classification by CN-Learn. The Venn diagrams show the
674 overlap of calls among the four callers (**Top left**) versus those that were validated by microarrays
675 (**Top right**). Venn diagrams depicting the overlap of all true calls among the four callers after
676 classification by CN-Learn (**Bottom left**) and true calls that were also validated by microarrays are
677 shown (**Bottom right**). (**B)** The distribution of all calls within the 262 samples based on the
678 probability scores (Y-axis) predicted by CN-Learn across four size ranges (X-axis) is shown. The
679 number of overlaps of each CNV with exome-based CNV callers is represented by different colors.
680 CNVs that validated with microarrays are indicated by filled circles, while CNVs that did not
681 validate with microarrays are represented by hollow circles.

Figure 1

CN-Learn - Conceptual workflow

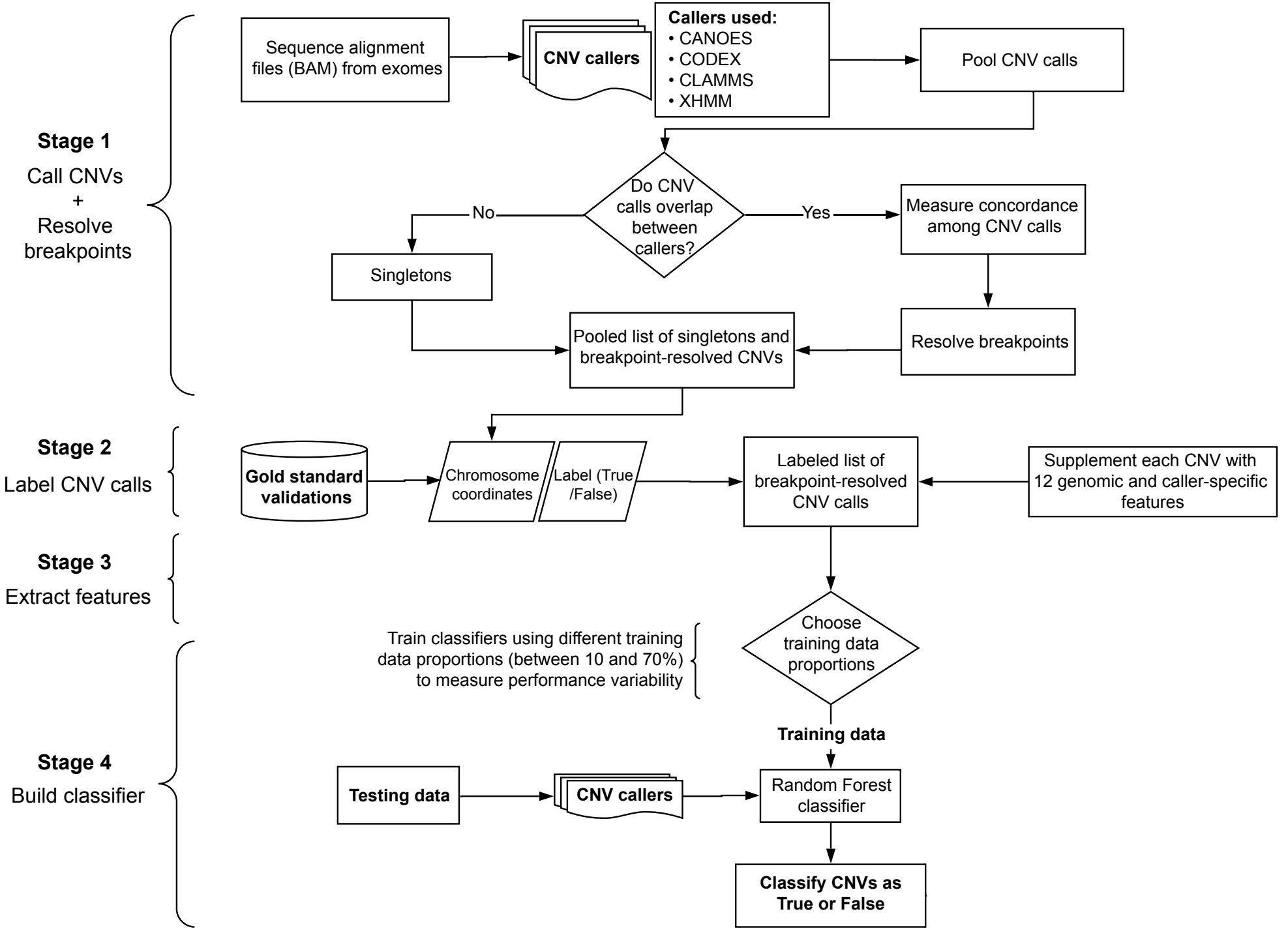


Figure 2

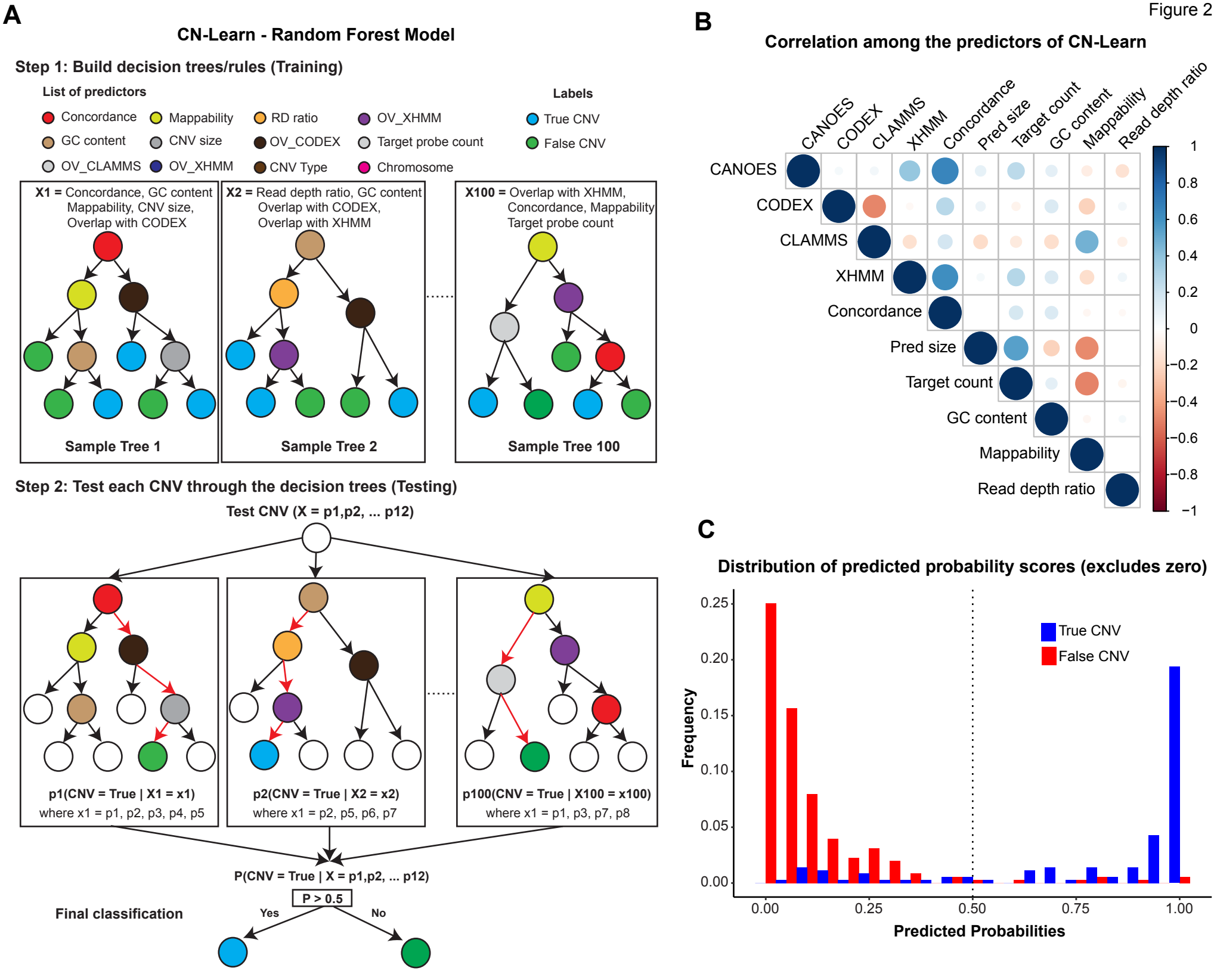

A
CN-Learn - Random Forest Model
Step 1: Build decision trees/rules (Training)
List of predictors
Concordance
Mappability
RD ratio
OV_XHMM
GC content
CNV size
OV_CODEX
Target probe count
OV_CLAMMS
OV_XHMM
CNV Type
Chromosome
Labels
True CNV
False CNV
X1 = Concordance, GC content Mappability, CNV size, Overlap with CODEX
X2 = Read depth ratio, GC content Overlap with CODEX, Overlap with XHMM
X100 = Overlap with XHMM, Concordance, Mappability Target probe count
Sample Tree 1
Sample Tree 2
Sample Tree 100
Step 2: Test each CNV through the decision trees (Testing)
Test CNV (X = p1,p2, ... p12)
p1(CNV = True | X1 = x1)
where x1 = p1, p2, p3, p4, p5
p2(CNV = True | X2 = x2)
where x1 = p2, p5, p6, p7
p100(CNV = True | X100 = x100)
where x1 = p1, p3, p7, p8
P(CNV = True | X = p1,p2, ... p12)
P > 0.5
Yes
No
Final classification
B
Correlation among the predictors of CN-Learn
CANOES
CODEX
CLAMMS
XHMM
Concordance
Pred size
Target count
GC content
Mappability
Read depth ratio
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1
C
Distribution of predicted probability scores (excludes zero)
Frequency
0.25
0.20
0.15
0.10
0.05
0.00
True CNV
False CNV
0.00
0.25
0.50
0.75
1.00
Predicted Probabilities

Figure 3

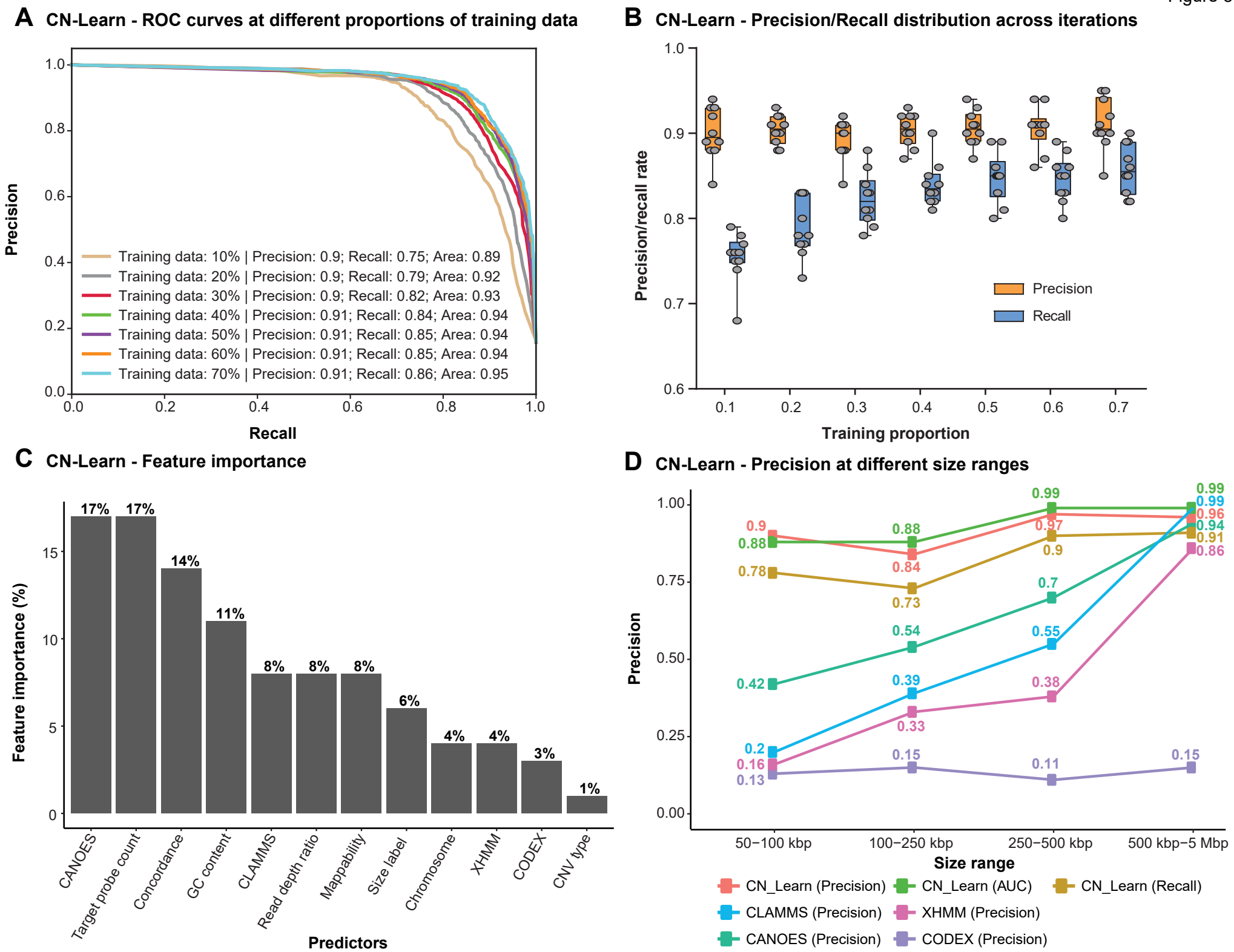
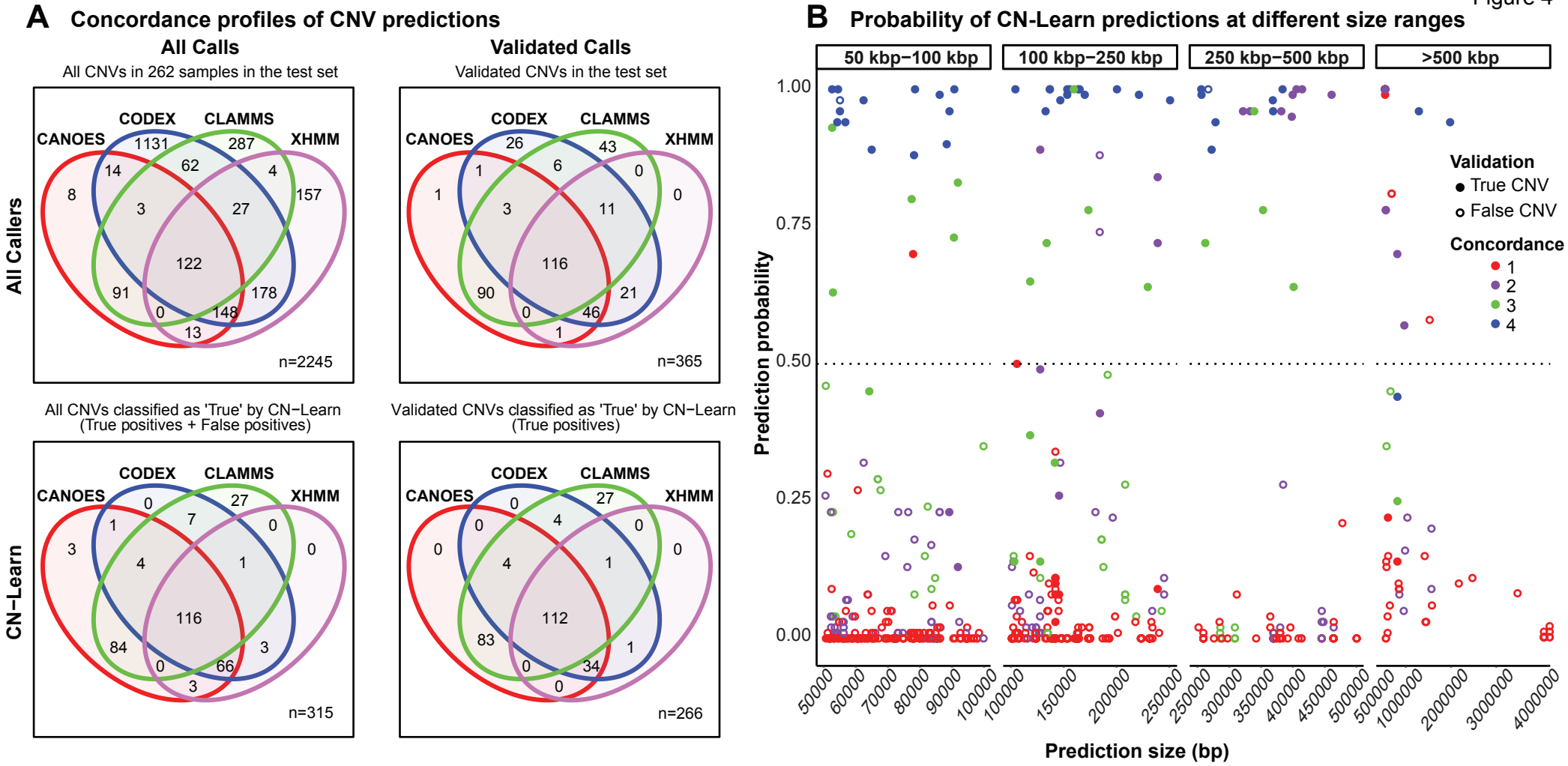

A
CN-Learn - ROC curves at different proportions of training data
Precision
Recall
Training data: 10% | Precision: 0.9; Recall: 0.75; Area: 0.89
Training data: 20% | Precision: 0.9; Recall: 0.79; Area: 0.92
Training data: 30% | Precision: 0.9; Recall: 0.82; Area: 0.93
Training data: 40% | Precision: 0.91; Recall: 0.84; Area: 0.94
Training data: 50% | Precision: 0.91; Recall: 0.85; Area: 0.94
Training data: 60% | Precision: 0.91; Recall: 0.85; Area: 0.94
Training data: 70% | Precision: 0.91; Recall: 0.86; Area: 0.95
B
CN-Learn - Precision/Recall distribution across iterations
Precision/recall rate
Training proportion
Precision
Recall
C
CN-Learn - Feature importance
Feature importance (%)
Predictors
CANOES
Target probe count
Concordance
GC content
CLAMMS
Read depth ratio
Mappability
Size label
Chromosome
XHMM
CODEX
CNV type
17%
17%
14%
11%
8%
8%
8%
6%
4%
4%
3%
1%
D
CN-Learn - Precision at different size ranges
Precision
Size range
50−100 kbp
100−250 kbp
250−500 kbp
500 kbp−5 Mbp
CN_Learn (Precision)
CN_Learn (AUC)
CN_Learn (Recall)
CLAMMS (Precision)
XHMM (Precision)
CANOES (Precision)
CODEX (Precision)

Figure 4


A
Concordance profiles of CNV predictions
All Calls
All CNVs in 262 samples in the test set
Validated Calls
Validated CNVs in the test set
All Callers
CANOES
CODEX
CLAMMS
XHMM
8
14
1131
62
287
4
157
3
27
122
91
178
0
148
13
n=2245
1
26
6
43
0
0
3
11
116
90
21
0
46
1
n=365
All CNVs classified as 'True' by CN−Learn
(True positives + False positives)
Validated CNVs classified as 'True' by CN−Learn
(True positives)
CN−Learn
3
1
0
7
27
0
0
4
1
116
84
3
0
66
3
n=315
0
0
4
27
0
0
4
1
112
83
1
0
34
0
n=266
B
Probability of CN-Learn predictions at different size ranges
50 kbp−100 kbp
100 kbp−250 kbp
250 kbp−500 kbp
>500 kbp
Prediction probability
1.00
0.75
0.50
0.25
0.00
Validation
True CNV
False CNV
Concordance
1
2
3
4
50000
60000
70000
80000
90000
100000
100000
150000
200000
250000
250000
300000
350000
400000
450000
500000
500000
1000000
2000000
3000000
4000000
Prediction size (bp)