Polishing copy number variant calls on exome sequencing data via deep learning

Furkan Özden, Can Alkan and A. Ercüment Çiçek

Genome Res. 2022 32: 1170-1182 originally published online June 13, 2022
Access the most recent version at doi:10.1101/gr.274845.120

Method

Polishing copy number variant calls on exome sequencing data via deep learning

Furkan Özden,[1] Can Alkan,[1] and A. Ercüment Çiçek[1,2]

[1]Department of Computer Engineering, Bilkent University, 06800 Ankara, Turkey; [2]Computational Biology Department, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA

Accurate and efficient detection of copy number variants (CNVs) is of critical importance owing to their significant association with complex genetic diseases. Although algorithms that use whole-genome sequencing (WGS) data provide stable results with mostly valid statistical assumptions, copy number detection on whole-exome sequencing (WES) data shows comparatively lower accuracy. This is unfortunate as WES data are cost-efficient, compact, and relatively ubiquitous. The bottleneck is primarily due to the noncontiguous nature of the targeted capture: biases in targeted genomic hybridization, GC content, targeting probes, and sample batching during sequencing. Here, we present a novel deep learning model, *DECoNT*, which uses the matched WES and WGS data, and learns to correct the copy number variations reported by any off-the-shelf WES-based germline CNV caller. We train DECoNT on the 1000 Genomes Project data, and we show that we can efficiently triple the duplication call precision and double the deletion call precision of the state-of-the-art algorithms. We also show that our model consistently improves the performance independent of (1) sequencing technology, (2) exome capture kit, and (3) CNV caller. Using DECoNT as a universal exome CNV call polisher has the potential to improve the reliability of germline CNV detection on WES data sets.

[Supplemental material is available for this article.]

Gene copy number polymorphism in a population owing to deletions and duplications of genomic segments substantially drives genetic diversity (Iafrate et al. 2004; Sebat et al. 2004), affecting ~7% of the genome (Sudmant et al. 2015). This class of structural variations (SVs), called copy number variations (CNVs), have also been associated with several genetic diseases and disorders such as neurodevelopmental/neurodegenerative disorders (Heinzen et al. 2010; Cooper et al. 2011; Levy et al. 2011; Pankratz et al. 2011; Zarrei et al. 2019) and various cancers such as breast, ovary, and pancreas cancers (Kumaran et al. 2017; Hieronymus et al. 2018; Macintyre et al. 2018; Reid et al. 2019). Karyotyping and microarray analyses have been the standard clinical testing for disease-causing CNVs for many years (Trost et al. 2018), but high-throughput sequencing (HTS) has replaced these techniques with the ability to theoretically capture all forms of genomic variation. Numerous CNV detection algorithms have enjoyed success by analyzing whole-genome sequencing (WGS) data using different sequence signatures such as read depth, discordant paired-end read mappings, and split reads (Ho et al. 2020). WGS is a convenient resource for CNV callers as it provides a near-Poisson depth of coverage (Belkadi et al. 2015). On the other hand, accurate CNV detection on whole-exome sequencing (WES) data has mostly been lacking. The algorithms that call CNVs on the WES data have notoriously high false-discovery rates (FDRs) reaching up to ~60%, which renders them impractical for clinical use (Tan et al. 2014; Zare et al. 2017). This is mainly owing to several problems associated with the WES technology such as nonuniform read depth distribution among exons caused by biases in (1) sample batches, (2) GC content, and (3) targeting probes (Krumm et al. 2012; Kadalayil et al. 2015; Kebschull and Zador 2015). It is unfortunate as WES data size is 10 times smaller (i.e., ~10 GB vs. ~100 GB), and it costs three times less compared with WGS, which makes it highly abundant and a common choice for analyzing complex genetic disorders (De Rubeis et al. 2014; The Deciphering Developmental Disorders Study 2015; Singh et al. 2019; Satterstrom et al. 2020). For instance, the Genome Aggregation Database (gnomAD) contains around 125,000 WES samples as opposed to 70,000 WGS samples (Karczewski et al. 2020). Thus, currently, such a rich resource of large-scale WES data cannot be fully utilized to investigate the contribution of CNV to disease etiology. To address this gap in the literature, we present the first of its kind, an exome CNV call *polisher* named *Deep Exome Copy Number Tuner* (*DECoNT*). The aim of the study is to improve the performance of any off-the-shelf WES-based germline CNV detection algorithm. DECoNT achieves this by learning to correct the CNV calls made using the noisy WES read depth signal via the *higher-quality* calls made using the WGS signal.

Results

Overview of DECoNT

DECoNT is a deep learning–based method that uses matched WES and WGS samples present in the 1000 Genomes Project (The 1000 Genomes Project Consortium 2015) data set to learn the association between (1) calls made by any CNV caller that use WES data and (2) ground-truth calls generated from the WGS data for the same sample. Based on a bidirectional long short-term memory (Bi-LSTM)–based architecture, it uses only WES read depth along with the calls from a third-party caller and learns to correct noisy

Corresponding authors: calkan@cs.bilkent.edu.tr, cicek@cs.bilkent.edu.tr
Article published online before print. Article, supplemental material, and publication date are at https://www.genome.org/cgi/doi/10.1101/gr.274845.120.

32:1170–1182 Published by Cold Spring Harbor Laboratory Press; ISSN 1088-9051/22; www.genome.org

predictions (Fig. 1). DECoNT can work with CNV callers that output integer copy number predictions (such as copy number 0, 1, 2, and 3) and categorical predictions (i.e., deletion, duplication, or no call). As the training phase is offline, polishing procedure is memory- and time-efficient, and it takes only a few seconds on average per sample. The models learned are universal in the sense that they are independent of (1) sequencing platform, (2) target capture kit, and (3) CNV caller. For instance, using models learned on the 1000 Genomes Project data set that uses Illumina as the sequencing platform and various capture kits such as Agilent and NimbleGen, DECoNT can correct calls made by any of the state-of-the-art CNV caller for samples obtained from other capture kits (e.g., Illumina Nextera exome enrichment kit), or different sequencing platforms (e.g., Illumina HiSeq 4000, Illumina NovaSeq 6000, and MGI) that are "unseen" during training. Thus, DECoNT is highly flexible and scalable, and it makes exome-based CNV detection practical by boosting the performance of virtually any WES-based CNV caller algorithm.

Bi-LSTM-based neural network learns to correct false-positive germline WES CNV calls

A Bi-LSTM network (Hochreiter and Schmidhuber 1997) is a type of recurrent neural network (RNN), which learns a representation (i.e., embedding) of a sequence by processing it for each time-step (i.e., each read depth value in the CNV region in our case) in forward and backward directions. While doing so, it remembers a summary of the sequence observed to capture the context for each time-step. RNNs and LSTM-based architectures have been widely and successfully used in the natural language processing domain to process sequence data (Yu et al. 2019).

DECoNT uses a single hidden layered Bi-LSTM architecture with 128 hidden neurons in each direction to process the read depth signal (Methods). First, the WES-based germline CNV caller result is obtained along with the read depth signal within the putative CNV regions. The Bi-LSTM subnetwork learns a transformed representation for the read depth sequence (Fig. 1). These embedding and the corresponding CNV calls are then input to a fully connected (FC) layer feed forward neural network, and finally, the FC layers predict the polished result for the call. DECoNT makes use of the calls made on the WGS data of the same sample as the ground truth for the learning procedure. We downloaded matched WGS data to obtain the ground-truth calls for the CNV events called on 802 WES samples of the same individuals in the 1000 Genomes Project data (The 1000 Genomes Project Consortium 2015). For each tool to be polished, we used 90% of the calls made on these samples for training and the remaining 10% for testing, unless otherwise stated. This roughly corresponds to a test set size of 80 samples.

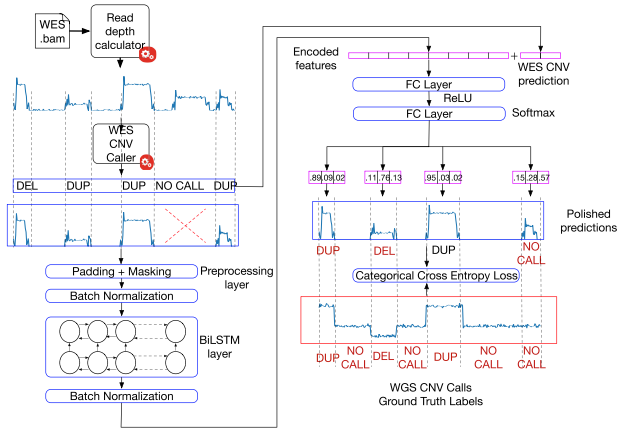

Figure 1. Learning workflow of DECoNT. First, BAM file that corresponds to a WES data set from the 1000 Genomes Project is used to calculate exome-wide read depth, which is input into a third-party WES-based CNV caller. The caller generates the calls for various regions that could be (1) a binary prediction like duplication, deletion (e.g., XHMM) (Fromer et al. 2012) as shown in the figure, or (2) an integer value that indicates the exact copy number (i.e., Control-FREEC) (Boeva et al. 2012). The read depth of the regions for which a call has been made is input to a Bi-LSTM model. Encoded features are passed from a series of fully connected (FC) layers along with the original prediction of the caller algorithm. Using the ground-truth calls from the WGS data of the same sample, the method learns to predict (correct) the calls using cross-entropy loss for the binary outputs (as shown in the figure) and using mean-squared loss for integral calls.

To evaluate DECoNT, we polished the call sets generated by several state-of-the-art WES-based germline CNV callers, which we group into two categories. The first type of algorithms makes discrete predictions for CNVs (i.e., deletion and duplication). We consider three methods in this category: (1) XHMM (Fromer et al. 2012), (2) CoNIFER (Krumm et al. 2012), and (3) CODEX2 (Jiang et al. 2018). The second type of algorithms predict the exact copy number as an integer value. The examples we consider of this type is Control-FREEC (Boeva et al. 2012) and CNVkit (Talevich et al. 2016). DECoNT architecture is flexible, and it can be easily modified to polish both types of algorithms (Methods). We trained a DECoNT model for every above-mentioned tool using three NVIDIA GeForce RTX 2080 Ti and one NVIDIA TITAN RTX GPU in parallel with training times ranging from ~1 to ~4 d (Methods).

We found that DECoNT is able to substantially improve the performance of all algorithms in almost all comparisons. For algorithms that make discrete predictions, we observed improved precision for calling both duplications and deletions (Fig. 2A). The largest gain in precision for calling duplications was seen in the CoNIFER call set, which was improved by threefold (i.e., 24.68%–75%). The largest gain in precision for calling deletions was again obtained for CoNIFER, which was improved by 1.5-fold (i.e., 45.45%–68.51%). Also, the overall precision was improved by 2.6-fold (i.e., 27.22%–71.11%) for CoNIFER. This improvement is especially striking as CoNIFER is relatively conservative compared with other algorithms, and it makes a small number of calls despite the relaxation of its parameters. For XHMM, we observed 1.4-, 1.7-, and 1.5-fold increases, which correspond to 20%, 29%, and 24% improvements in duplication, deletion, and overall precision, respectively. We saw a similar trend for CODEX2. Before polishing with DECoNT, CODEX2 achieved 12% precision for duplications, 45% for deletions, and 27% overall. DECoNT provided a 1.9-fold increase in precision for calling duplications (11% improvement), 1.5-fold increase in precision for calling deletions (23% improvement), and 1.75-fold increase in overall precision (20% improvement). Confusion matrices before and after polishing CNV calls generated with all tools are shown in Supplemental Figure 1. We note that these improvements are obtained in a very short time (seconds per sample). Increased precision is an important result for life scientists who work with these calls as the reliability of the calls are substantially increased as the number of false positives are substantially decreased.

As for Control-FREEC and CNVkit, which output exact copy number values, we evaluated their performance by calculating their absolute error (AE). For Control-FREEC, we considered 20,482 CNV calls (Methods) (Fig. 2B). DECoNT improved the AE

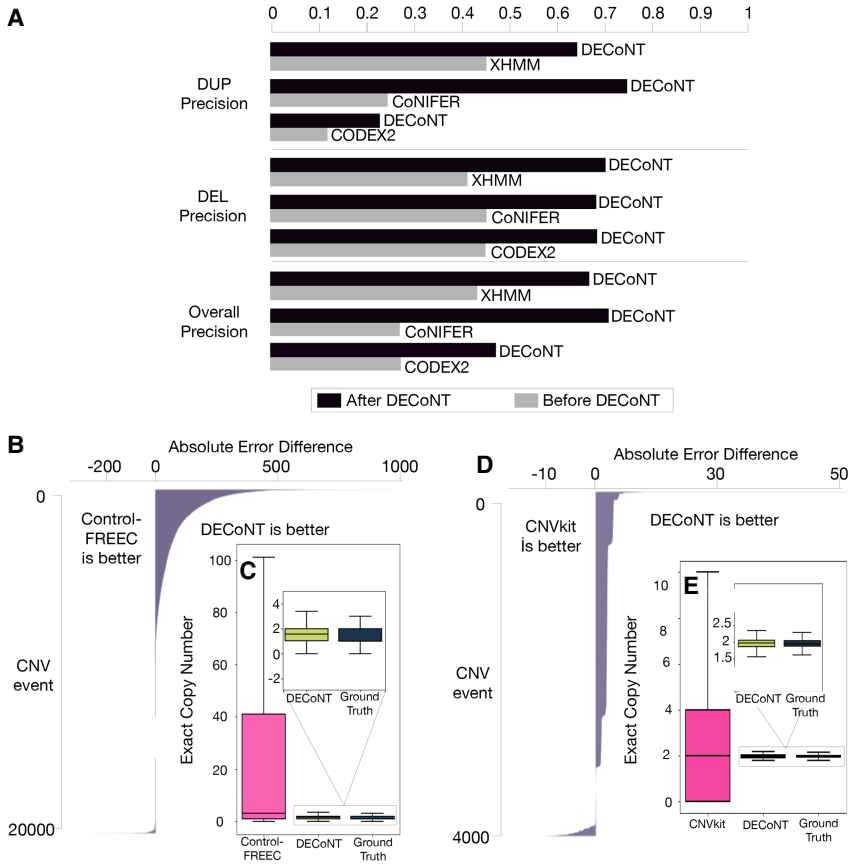

Figure 2. The performance comparison of the WES-based CNV callers before and after polishing with DECoNT. (*A*) For the tools that predict the existence of a CNV event (XHMM, CoNIFER and CODEX2) are evaluated with respect to *duplication call precision*, *deletion call precision*, and *overall precision*, DECoNT improves the performance for all tools in all settings and results in substantial improvements. Different shades of gray represent different tools, and the attached black bars represent the DECoNT-polished version of those tools. (*B*) In this panel, we compared Control-FREEC and the DECoNT-polished with respect to absolute error (AE) difference on each sample (i.e., events). Bars to the *right* indicate the magnitude of the improvement owing to polishing of DECoNT. For more than half of the samples, DECoNT results show improvement. (*C*) The distribution of the unpolished Control-FREEC predictions in the test samples (pink) is quite different than the ground-truth CNV distribution. On the other hand, DECoNT polished versions of the same events (dark blue) highly resemble the distribution of the ground-truth calls. Black lines across the boxes are median lines for the distributions. Black vertical lines are whiskers and 1.5× inter-quartile range is defined with the horizontal lines at the *top* and *bottom* of the whiskers. The 1000 Genomes Project WGS samples are used as ground-truth calls in all analyses. (*D*,*E*) The results for CNVkit, similar to *B* and *C*. For each polished tool, we used 90% of the calls made on 802 1000 Genomes Project samples for training and the remaining 10% of the calls for testing. This roughly corresponds to a test set size of 80 samples.

in 74.58% of the test samples for an average *AE* improvement of 47.39. On the other hand, DECoNT deteriorated the performance in 25.35% of the test samples for an average *AE* deterioration of only 1.2. Although unpolished Control-FREEC predictions had a Spearman's correlation coefficient of 0.227 with matched ground-truth copy numbers, DECoNT-polished predictions had a Spearman's correlation coefficient of 0.568 (Fig. 2C). Additionally, DECoNT-polished predictions highly resembled the distribution of the ground-truth calls. To mimic the discrete prediction case, we also discretized the CNV calls of Control-FREEC to *Deletion* (CN < 2), *Duplication* (CN > 2), and *No-Call* (CN = 2) categories to measure precision as defined in Methods. Again, DECoNT was able to improve the DEL and DUP precision up to threefold (for details, see Supplemental Note 1). We evaluated the performance of CNVkit (the other tool that outputs exact copy number values) on 3972 CNV calls (Methods) (Fig. 2D). DECoNT improved the AE in 86.78% of the test samples for an average *AE* improvement of 1.82 and deteriorated the performance in 13.21% of the test samples for an average *AE* deterioration of only 0.66. Raw CNVkit predictions had a Spearman's correlation coefficient of 0.0156, and DECoNT-polished predictions had a Spearman's correlation coefficient of 0.122 (Fig. 2E). Similar to Control-FREEC, DECoNT-polished CNVkit predictions highly resembled the distribution of the ground-truth calls. We observed similar precision improvements in CNVkit's classification performance after polishing when we discretized its predictions as we did for Control-FREEC (see Supplemental Table 2).

To show the need for an algorithm that uses a complex model like DECoNT in this application, we used standard machine learning algorithms for polishing and compared the performance. We used SVM and logistic regression for the discrete prediction case and polynomial regression for the exact prediction (rounded). We showed that these models actually deteriorate the baseline caller performance, and we need more complex models like DECoNT uses for this task. Details of our experiments are given in Supplemental Note 2. Moreover, we checked if the hidden Bi-LSTM encodings of DECoNT correlate with basic sequence features, such as mappability and GC content. For this, we obtained the 1000 Genomes Project WES test set CNV calls made by XHMM and visualized them using t-SNE (Supplemental Fig. 7). We did not observe any clustering patterns with respect to these features. This indicates that DECoNT predictions do not directly depend on these features, and it cannot be replaced by a simpler method that uses GC content and mappability.

We also investigated DECoNT's polishing performance on a consensus WES-based germline CNV caller, CNLearn (Pounraja et al. 2019). CNLearn first runs four different WES-based callers (CANOES, CODEX, CLAMMS, XHMM), and then using a random forest classifier, it learns to aggregate the results of these programs. We obtained 39 CNV predictions of CNLearn on four samples from the 1000 Genomes Project (default settings are used) (S. Girirajan, pers. comm.). The list of these samples is given in Supplemental Note 4. Using the CNVnator calls obtained on the WGS data of the samples as the ground truth, we observed that CNLearn achieved a precision of 0.79. Using the DECoNT model trained using XHMM calls, we polished the results of CNLearn and improved the precision to 0.889. Note that CNLearn requires more computation as it uses many models, yet DECoNT was able to improve the performance even when using a cross-model

polisher. More specifically, CNLearn and polished-CNLearn did not agree on eight calls, out of which the polished version was correct in four, the unpolished version was correct in two, and both were incorrect in two. The list of these calls is also given in Supplemental Table 3.

Polishing performance on a validated CNV call set

To further test the polishing performance of DECoNT, we also used a highly validated CNV call set published by Chaisson et al. (2019). This data set contains the WGS-based CNV calls of nine genomes selected from the 1000 Genomes Project samples, for which a consensus call set was obtained using 15 different WGS-based CNV callers with comparisons against high-quality SVs generated using long-read Pacific Biosciences (PacBio) data with a single-base-pair breakpoint resolution (Methods). We used WGS data from eight samples that have matched WES data available.

Using the same models explained in the subsection "Overview of DECoNT," we corrected the CNV calls made on WES data generated from eight samples made by XHMM, CoNIFER, and CODEX2. Note that none of the DECoNT models were trained with the data of these individuals. Table 1 summarizes the prediction performances before and after polishing with DECoNT, with respect to validated WGS-based calls.

Similar to the analysis above, DECoNT improved the performance of all three algorithms in all comparisons. The most substantial improvements were observed for CoNIFER; 7%, 31.4%, and 16% improvements were observed for duplication, deletion, and overall precision, respectively. It is noteworthy to mention that although CoNIFER did not report any deletion events, DECoNT was able to correct false duplication calls as true deletion calls and thus increase the precision to 31.4% in this category. Supplemental Figure 2 shows the confusion matrices obtained before and after polishing by DECoNT. For XHMM and CODEX2, we observed consistent improvements reaching up to nearly twofold for CODEX2.

Polishing performance generalizes to other sequencing platforms

We obtained the training data from the 1000 Genomes Project, in which the WES component was produced using Illumina Genome Analyzer II and Illumina HiSeq 2000. Although data from these platforms are abundant and sufficient training data set size can be found, for users using other sequencing platforms, it might not be possible to train DECoNT owing to the lack of matched WES and WGS samples. We therefore evaluated whether models trained on the available 1000 Genomes Project data can be used to polish CNV calls made on WES samples obtained using other sequencing platforms or capture kits that were not used while training DECoNT models (Methods).

We obtained the WES data generated from the genome of NA12878, sequenced using four different platforms—Illumina NovaSeq 6000; (2) Illumina HiSeq 4000; (3) BGISEQ-500; and (4) MGISEQ-2000—to test DECoNT's efficacy across sequencing technologies. We used WES-based CNV callers that we considered previously to call CNV events on these four WES samples.

Even though DECoNT was not trained using the read depth information or the CNV events generated by these sequencing platforms, it still can generalize from the training on the 1000 Genomes Project data, and it substantially improved the performances of XHMM, CoNIFER, and CODEX2 (Table 2; Supplemental Fig. 3). We observed improvements in 34 out of 36 tests.

The most substantial improvement was observed for CODEX2 that corresponded to an average 2.6-fold increase in performance. This even exceeded testing performance on the same platform as training (i.e., approximately twofold improvement) that we reported above. For XHMM, the performance was improved for 10 out of 12 tests, doubling the performance in overall precision performance for BGISEQ and MGISEQ platforms. For NovaSeq 6000 and HiSeq 4000, the performance deteriorated in duplication precision. However, XHMM made a few duplication calls: three and two, respectively. Although DECoNT kept the true positives, it added a few false positives, and this resulted in the performance decrease in these settings. CoNIFER did not report any events on the data generated by the NovaSeq 6000 platform despite tuning its parameters to more relaxed settings. On BGISEQ-500 and MGISEQ-2000 platforms, even though CoNIFER did not report any duplication calls, DECoNT returned duplication calls and increased the duplication precision from 0 to 1%. Although CoNIFER did not report any duplication calls on the data generated by the HiSeq 4000 platform, DECoNT was able to increase the precision to 50% by reporting one true-positive and a false-positive call. The trend in deletion precision performance is similar. Finally, overall precision performance consistently improved in all tests, and the improvement ranged from 0.3% to 2.3%.

For Control-FREEC, ~65% to ~74% of the CNV calls have been improved as opposed to only ~7% to ~8% of the calls that have been deteriorated by DECoNT. We observed a decrease in average AE after polishing in all four platforms, which ranges from 0.94 to 1.0 (Table 3).

We note that the improvements provided by DECoNT on the BGI and MGI platforms are important as these systems were developed by a different manufacturer and use sequencing chemistry different from that of Illumina. Because these platforms are expected to have different systematic biases in read depth distributions compared with the DECoNT's training data, we would also expect a lower performance. Yet, DECoNT was able to generalize well and consistently proved to be useful across a diverse set of technologies. Overall, the performance was on par with the tests obtained on data generated by Illumina Genome Analyzer II and Illumina HiSeq 2000. Polishing procedure consistently improved the performance in a platform-independent manner.

Table 1. The performances of the WES-based CNV caller algorithms before and after polishing (DEL, DUP, and overall precision)

	DUP precision		DEL precision		Overall precision	
Tool	Default	Polished	Default	Polished	Default	Polished
XHMM	0.064	**0.071**	0.257	**0.387**	0.135	**0.170**
CoNIFER	0.090	**0.160**	0.0[a]	**0.314**	0.090	**0.250**
CODEX2	0.027	**0.046**	0.387	**0.685**	0.185	**0.350**

Validated WGS CNV call set of Chaisson et al. (2019) is used as the ground-truth CNV call set. We first use matched WES reads to call WES CNVs using CoNIFER, CODEX2, and XHMM. Then, we use DECoNT to polish obtained CNV calls. Table shows the DEL, DUP, and overall precision of the methods. Ninety percent of the calls made on the 1000 Genomes Project data are used for training the models, and nine samples from Chaisson et al. are used for validation of these results. Bold indicates the best result in each category.
[a]CoNIFER does not report any deletion events on this set of WES samples.

Polishing performance on calls from other CNV callers

Ideally, a distinct DECoNT model is trained for every WES-based germline CNV caller. This makes sense as the CNV regions and

Table 2. The performance of discrete germline WES-based CNV callers on NA12878 data before and after being polished by DECoNT

		DUP precision		DEL precision		Overall precision	
Platform	Tool	Default	Polished	Default	Polished	Default	Polished
NovaSeq 6000	XHMM	**0.660**	0.330	0.078	**0.111**	0.097	**0.133**
	CoNIFER	NA[a]	NA[a]	NA[a]	NA[a]	NA[a]	NA[a]
	CODEX2	0.043	**0.139**	0.198	**0.398**	0.112	**0.266**
HiSeq 4000	XHMM	**0.500**	0.125	0.093	**0.156**	0.100	**0.152**
	CoNIFER	0.0[b]	**0.500**	0.191	**0.192**	0.191	**0.214**
	CODEX2	0.032	**0.075**	0.188	**0.389**	0.099	**0.212**
BGISEQ-500	XHMM	0.045	**0.076**	0.157	**0.176**	0.088	**0.200**
	CoNIFER	0.0[b]	**0.010**	0.052	**0.082**	0.052	**0.055**
	CODEX2	0.051	**0.156**	0.214	**0.492**	0.125	**0.364**
MGISEQ-2000	XHMM	0.045	**0.076**	0.157	**0.176**	0.088	**0.200**
	CoNIFER	0.0[b]	**0.010**	0.052	**0.082**	0.052	**0.055**
	CODEX2	0.051	**0.156**	0.214	**0.492**	0.125	**0.364**

We evaluated caller performance on NA12878 data obtained using four different sequencing platforms: (1) NovaSeq 6000, (2) HiSeq 4000, (3) BGISEQ-500, and (4) MGISEQ-2000. Note that DECoNT models did not train on the data sequenced with any of these sequencing platforms or on NA12878 sequencing data of any form. Ninety percent of the calls made on the 1000 Genomes Project data are used for training the models. DUP precision, DEL precision, and overall precision results are shown. In all comparisons, DECoNT provides substantial improvements showing the generalizability of our models trained on the 1000 Genomes Project data. Bold indicates the best result in each category.
[a]CoNIFER does not report any CNV calls on NA12878 WES data sequenced with NovaSeq 6000. For that reason, DECoNT has no input to correct, and thus, that comparison is not applicable.
[b]CoNIFER does not report any duplication events in the unpolished case. The 1000 Genomes Project WGS samples are used as ground-truth calls.

copy numbers may substantially differ among algorithms in their recommended settings (e.g., CODEX2 calls 10× more events than XHMM). We next aimed to understand whether it is still reliable to use a DECoNT model trained using calls made by one algorithm can be applied to polish the calls made by others in the absence of a trained model (e.g., owing to time and data constraints in training).

We used the same DECoNT models trained for XHMM, CoNIFER, and CODEX2 on the 1000 Genomes Project data. For each tool-specific DECoNT model, we polished the calls made by other callers on samples that are not used for training. For instance, we polished the calls made by CODEX2 using the DECoNT model trained on XHMM calls. This experiment resulted in six pairwise tests (i.e., for two-way comparison among every tool pair). We measured the performance of the polishing procedure using duplication call precision, deletion call precision, and overall precision to obtain 18 performance results in total (Methods).

We observed that DECoNT improved the performance metric in 10 out of the 18 comparisons, XHMM-trained DECoNT consistently improved the other tools' performance in all metrics, except DEL precision when polishing calls reported by CoNIFER, ranging from 2% to 13% (Fig. 3; Supplemental Fig. 1). Duplication precision was improved in most of the cases with the exception of CoNIFER-trained and CODEX2-trained DECoNT models that deteriorated the performance of XHMM by 11% and 8%, respectively. For deletion precision, this was not the case, as the deletion precision was improved for CODEX2 for both DECoNT models. However, for CoNIFER, deletion precision deteriorated by 13% and 45% when polished with XHMM-trained and CODEX2-trained DECoNT models, respectively. This is because of very limited number of deletion CNV predictions of CoNIFER as even a small perturbation to the true positives of deletion calls yields large differences in precision. Also, CoNIFER-trained DECoNT model very slightly deteriorated deletion precision of XHMM calls by 5%. Although XHMM improved overall precision for other methods, in half of the overall precision comparisons, the performance was decreased.

Overall, DECoNT showed to be still effective despite being trained using a different call set. The training process uses the read depth information for the event regions, which enables DECoNT to generalize to polish other tools. Although, arguably, it can be used to polish calls generated by other tools, we suggest that a DECoNT model trained on the calls of the to-be-polished WES-based caller should be used, as the improvements in the discussed performance metrics are higher in this case.

Table 3. The performance of Control-FREEC on NA12878 data before and after being polished by DECoNT

Platform	No. of events	% of improved events	% of deteriorated events	Mean absolute error (MAE) difference decreased by
NovaSeq 6000	329	**73.85%**	7.59%	**0.9392**
HiSeq 4000	437	**70.94%**	6.86%	**1.0022**
BGISEQ-500	367	**64.57%**	8.17%	**0.9809**
MGISEQ-2000	367	**64.57%**	8.17%	**0.9809**

We evaluate caller performance on NA12878 data obtained using four different sequencing platforms: (1) NovaSeq 6000, (2) HiSeq 4000, (3) BGISEQ-500, and (4) MGISEQ-2000. Note that DECoNT models did not train on the data sequenced with any of these sequencing platforms or on NA12878 sequencing data of any form. Ninety percent of the calls made on the 1000 Genomes Project data are used for training the models. Table shows the number of CNVs reported on each sample, the percentage of improved and deteriorated events, and the average decrease in absolute error after being polished by DECoNT. In all comparisons, DECoNT provides substantial improvements, showing the generalizability of our models trained on the 1000 Genomes Project data. Bold indicates the best result in each category.

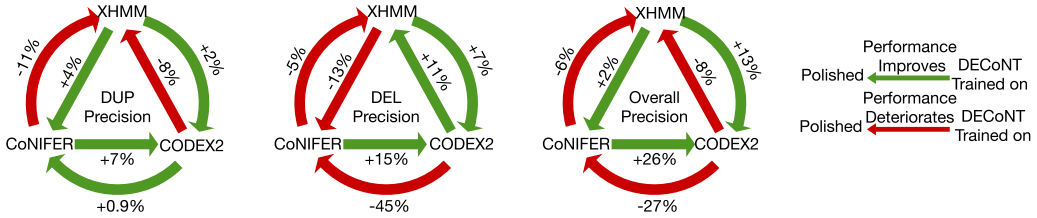

Figure 3. Performance of DECoNT when polishing calls from unseen CNV callers. DECoNT learns a different set of weights and a different model for each WES-based CNV caller. To show the cross-model performance, we used DECoNT to correct CNV calls made by tools other than the ones used for training. We try every pair combination. Tools being pointed by an arrow are call-generating tools (i.e., being corrected). Tools at source of the arrow are the tools that are used to train the DECoNT model. Green arrows indicate improvement, and red arrows indicates deterioration in the corresponding performance metric. For each polished tool, we used 90% of the calls made on 802 1000 Genomes Project samples for training and the remaining 10% of the calls for testing. This roughly corresponds to a test set size of 80 samples.

Minimizing the false calls made by DECoNT

Despite substantial improvements over the precision of the CNV callers, one could argue that for downstream usage in the clinic one needs to further limit the false calls, which is typically achieved through focusing only on the calls with high confidence. We investigated the effect of this on the precision of (1) raw XHMM calls and (2) DECoNT-corrected XHMM calls. For this, we gradually adjusted both XHMM and DECoNT to be more conservative in making calls to limit false positives. That is, we set a confidence threshold at which each algorithm made a call only if an event's likelihood (e.g., DUP likelihood) exceeds this threshold. For instance, DECoNT in the default mode would call a DEL if DEL likelihood is 0.34 where DUP and NO-CALL likelihoods are both 0.33. Now, if we set the confidence threshold as 0.5, a call is made only if the likelihood of the event is >0.5.

As seen in Figure 4, forcing XHMM to be more conservative increased its precision. However, we observed that this cannot replace the polishing procedure. In all categories, DECoNT correction consistently dominated the precision performance of raw XHMM calls by a large margin (by ~20%). Second, we observed that it was possible to achieve >~95% precision by keeping only very high confidence calls (i.e., confidence threshold ~= 1). We also observed that such high precision was not obtained by sacrificing recall (i.e., making only a very small number of calls) as about 1000 calls were still kept in the most conservative setting out of 6832 (see the last panel in Fig. 4).

We then further checked if DECoNT acts as a basic thresholding scheme for XHMM or if it can correct calls that pass quality thresholds of XHMM. We observe that DECoNT is successful in true-negative corrections, meaning converting false DEL/DUP calls made by XHMM into a NO-CALL even with high-confidence SQ (some deletion probability parameter) scores such as greater than 60. See Supplemental Figure 8. We also observe one true-positive correction. These indicate that common filtering of XHMM is not sufficient as we observe DECoNT-corrected calls at all XHMM quality levels.

The effect of polishing on pathogenic CNVs for clinical use

To better understand the benefits of using a call polished in the clinical setting, we tested whether DECoNT is able to reduce clinically relevant false calls predicted in the genomes of healthy individuals. For this purpose, we used the 1000 Genomes Project test

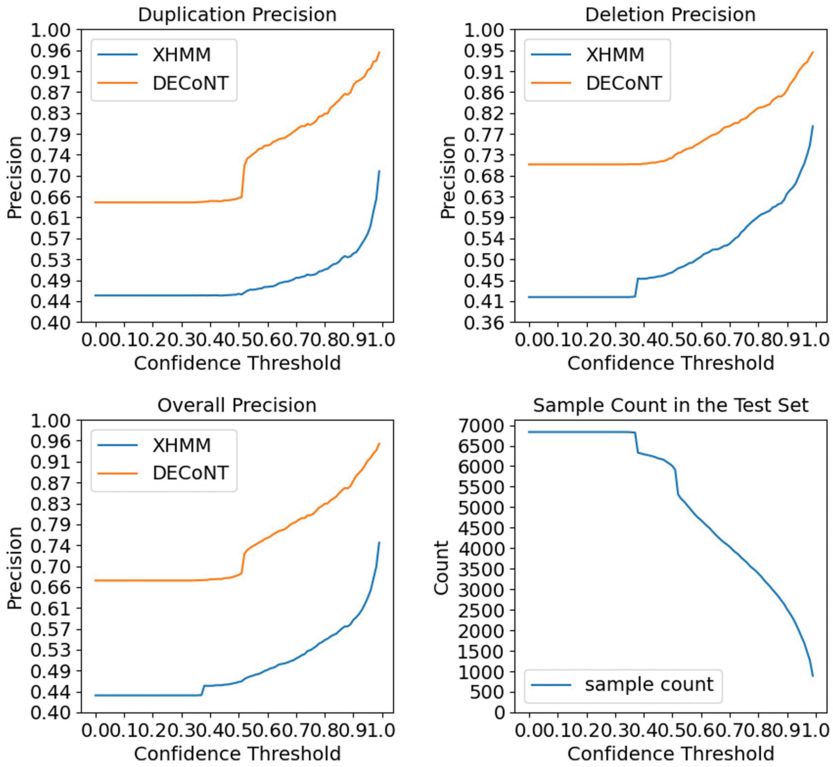

Figure 4. Precision of raw XHMM and DECoNT-corrected XHMM calls are shown with respect to varied confidence threshold (DEL, DUP, and overall precision); 6832 CNV calls made by XHMM on our test data set (the 1000 Genomes Project WES data set test samples) are used. The ground truth is the CNV calls made by CNVnator on the corresponding WGS samples. The last panel shows the number of calls remaining as the threshold gets more conservative. For each polished tool, we used 90% of the calls made on 802 1000 Genomes Project samples for training and the remaining 10% of the calls for testing. This roughly corresponds to a test set size of 80 samples.

samples, because the individuals represented in the 1000 Genomes Project do not carry any known disease of genomic origin (The 1000 Genomes Project Consortium 2015). We reasoned that if a genome in the 1000 Genomes Project is predicted to contain a pathogenic CNV by any CNV caller, the predicted variant should be a false positive. We evaluated the XHMM calls in this analysis, and we used the XCNV tool (Zhang et al. 2021) to assign pathogenicity score (*MVP pathogenicity*). We found 123 pathogenic and likely-pathogenic CNVs (MVP > 0.47) that were listed in the XHMM call set. DECoNT polishing modified 47 of the calls, in which 40 were changed to *no-call* (32%). Out of the remaining seven CNVs, six were deletion calls that were converted to duplication calls. We reanalyzed the polished set using XCNV, and among those "newly updated" duplications, five were classified as "likely benign" and one was classified as "uncertain." The last modified call was a duplication that was updated to a deletion by DECoNT; its pathogenicity score was downgraded to "likely pathogenic." In summary, 37% of the calls that were initially classified as risky were removed from the clinically important list after polishing.

We also analyzed the WES data of 16 bladder cancer patients studied by Guo et al. (2013) (accessed from the NCBI Sequence Read Archive [SRA; https://www.ncbi.nlm.nih.gov/sra] under accession number SRP017787). The study reports the following pathogenic and real-time qPCR–validated CNVs in regions that encompass genes that are associated with bladder cancer in the literature (Mhawech-Fauceglia et al. 2006):

1. A deletion in Chromosome 9 between *20,305,364* and *24,115,910* for patients B63, B80-0, and B112. This region contains the *CDKN2A* and *CDKN2B* genes. This is a commonly reported variant in bladder cancer (Williamson et al. 1995) and is the most significantly altered region in this study cohort.
2. A duplication in Chromosome 11 between *68.8* million and *69.8* million that contains the *CCND1* gene for patients B37 and B103.
3. A duplication in Chromosome 5 between *79.9* million and *80* million that contains the *DHFR* gene for patients B15, B18, B19, B24, B34 and B50.
4. A duplication in Chromosome 17 between *35* million and *35.2* million that contains the *ERBB2* gene for patients B9, B23, B80, B80-5, and B86.

We tested whether XHMM can detect these validated and pathogenic CNVs on the WES data of these samples accurately and whether DECoNT polishing could help the clinical assessment by polishing potentially incorrect calls. We ran XHMM on these samples using the relaxed setting mentioned in Supplemental Note 3 in the batch mode. We also used matching samples taken from normal tissues of these patients. We then used DECoNT to polish the calls of XHMM and focused on the calls in the above-mentioned validated regions. For the list of resulting calls, see Supplemental Table 4.

For regions 2, 3, and 4, XHMM did not return any calls in spite of the relaxed parameter setting. Hence, DECoNT did not have any calls to polish for the 13 patients related to these regions. For B112, XHMM returned two deletion calls of lengths ~2.75 Mbp and ~72.5 kbp in the validated region that were confirmed by DECoNT consistently with the ground truth. Similarly, for B63, a deletion call in the validated region of length ~800 kbp was confirmed by DECoNT. For B80-0, XHMM returned two overlapping and similar deletion calls of length ~450 kbp, which DECoNT again confirmed consistently with the ground truth. However, in the normal tissue sample from the same individual, it returned a major (~550-kbp) duplication event that covers the deletion call region in the tumor. This is extremely unlikely, and DECoNT was able to convert this to a *no-call*, which shows that DECoNT is also useful in a clinical setting despite being a germline CNV caller.

In depth analysis of the polishing procedure

Next, we investigated the decisions made by DECoNT on the 6832 calls made by XHMM on the 1000 Genomes Project test set to understand whether polishing performs better or not depending on the underlying sequence properties. First, we evaluated DECoNT's calls made on pseudoautosomal regions (PAR1 and PAR2) of Chromosome X. We observe that the performance of XHMM is the lowest in these regions. Yet, DECoNT was still able to improve the performance in eight out of nine categories as shown in Supplemental Table 1.

We also analyzed the size distribution of the true-positive calls, that is, either the original XHMM call or the DECoNT-polished version of the call matches with the ground truth. In Supplemental Figure 4, we show that the length distribution of the calls in which the DECoNT-polished version of the call remains unchanged had a large variance (up to 1 Mbp). Yet, for the calls in which only DECoNT-corrected versions match with the ground-truth calls, sizes are up to 500 kbp, showing that XHMM seems to require less polishing for larger CNVs.

We then investigated whether the importance of DECoNT polishing varies across chromosomes. Figure 5 shows the chromosome-wise stratification of the calls in which each dot represents a call made by XHMM, colored by one of the four possibilities: (1) DECoNT-polished call is correct (i.e., matching WGS call, semi ground-truth) and XHMM call is incorrect; (2) XHMM call is correct and DECoNT agrees, (3) both polished and original calls are incorrect; and (4) XHMM call is correct and DECoNT-polished call is incorrect. We observed that 80% of the XHMM calls made on Chromosome 8 are changed and corrected by DECoNT. This number is 84% for Chromosome Y. This indicates that researchers focusing on these regions of the genome should definitely use polishing to obtain better calls. Forty-four percent of the XHMM calls on Chromosome 20 are corrected, and this is the best-case scenario for XHMM, yet nearly half of the calls needed correction. We observed that there was a large variance in the length of the calls made on Chromosome Y, and polishing helped regardless of the size (i.e., both short and long calls are corrected). On Chromosome 13, most calls were close to the median size with low variance, and almost all DECoNT-corrected calls were relatively short. This indicates that DECoNT is useful when the calls on a chromosome have both low and high variance in size, and it can correct both short and long calls. We observe that DECoNT made a small number of mistakes that were uniformly distributed across chromosomes.

Next, we investigated whether the calls corrected by DECoNT differed with respect to the GC content of the region. Supplemental Figure 5 shows the GC distribution of the call regions, stratified with respect to the four categories in Figure 5. DECoNT-polished calls tend to have a larger GC content. Correctly polished calls had a mean GC% of ~53%, and correct-and-agreed XHMM calls had a mean GC% of ~48%. XHMM had no correct calls in both GC-rich (>65%) and GC-poor areas (<35%), and DECoNT was able to correct many calls in these biased regions.

Finally, we checked whether the number of probes in a CNV region introduces a bias. We observed that a high-probe count leads XHMM to make more erroneous calls. As shown in

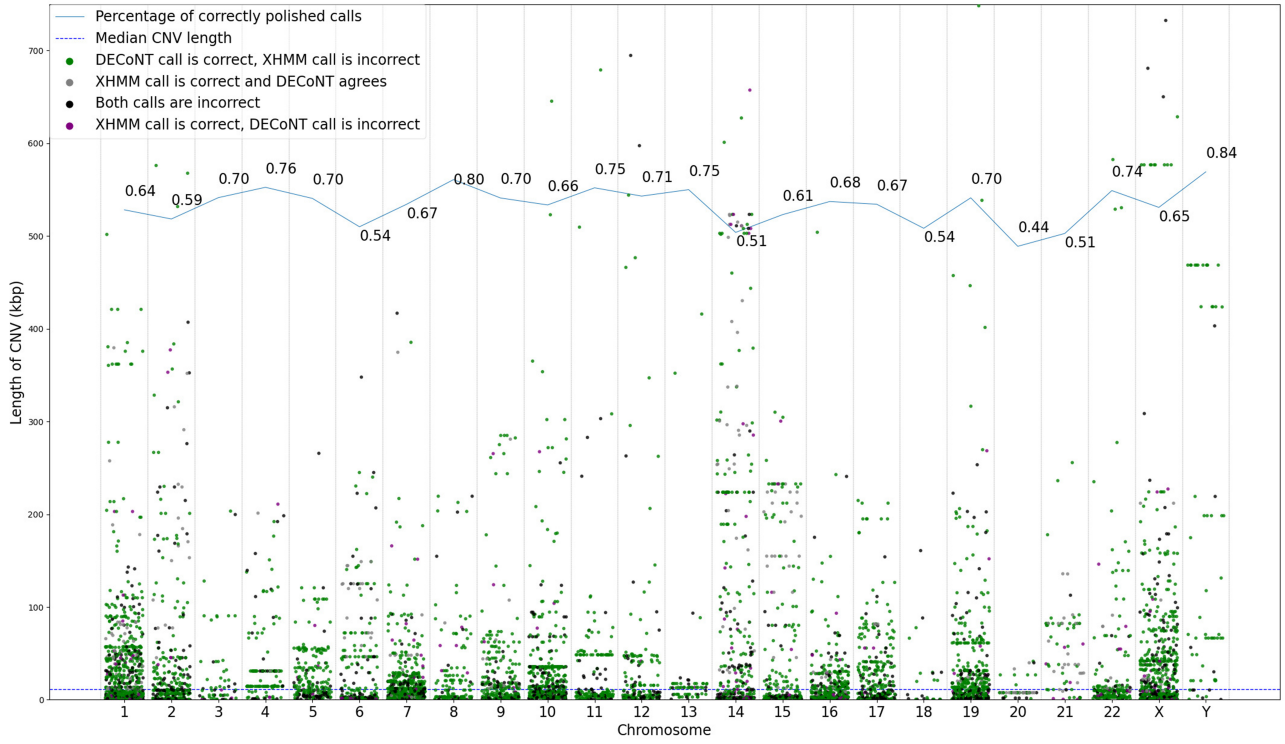

Figure 5. Each dot in this figure corresponds to one of the 6832 CNV calls made by XHMM on our test data set (the 1000 Genomes Project WES data set test samples). The ground truth is the CNV calls made by CNVnator on the corresponding WGS samples. The calls are stratified w.r.t. their chromosomes. The *y*-axis of the plot represents the length of calls. A gray dot indicates that the XHMM call is not changed by DECoNT and the prediction matches the ground truth (correct), whereas a black dot indicates that both DECoNT's and XHMM's decisions do not match the ground truth (incorrect). A green dot indicates the call is corrected by DECoNT and the XHMM call was incorrect. Finally, a purple dot indicates that DECoNT changes the prediction of XHMM and both the original and the changed calls are incorrect. For each chromosome, a random jitter is added to the *x*-axis for better visualization. The solid line on the *top* of the figure shows the ratio of the number of DECoNT-corrected calls and the number of all XHMM calls in that chromosome. The dashed line indicates the median CNV call length across chromosomes. For each polished tool, we used 90% of the calls made on 802 1000 Genomes Project samples for training and the remaining 10% of the calls for testing. This roughly corresponds to a test set size of 80 samples.

Supplemental Figure 6, all calls with a probe count larger than around 50 were incorrect. DECoNT was able to correct all XHMM calls with probe count higher than 60. We conclude that DECoNT polishing is essential especially in regions with systematic biases.

Hyperparameter selection for the base callers

One issue with polishing is to come up with a recipe to set the parameters of the CNV caller to achieve the best performance. We mostly used the suggested parameters and had to relax CoNIFER's parameters as it did not return many calls. One other option is to run the CNV caller in the most relaxed setting to improve sensitivity and to let DECoNT correct the likely higher number of false positives. We tested if this is feasible using XHMM, which is the best-performing algorithm in our benchmarks. As detailed in Supplemental Note 3, we used XHMM using two more settings: one with a relaxed and one with a conservative parameter set in addition to our original test. We then polished the calls using DECoNT. The precision improvement was stable at ~22% in all three settings. The polished precision of the relaxed setting was 10% lower compared with the suggested setting. Conservative and suggested setting precision values differed by only ~1%. Thus, we suggest using the *suggested* parameter settings of the CNV caller unless it makes a very small number calls, which is insufficient for training.

Discussion

HTS platforms are now the dominant source of data generation for biological and medical research and are on their way to be routinely used for diagnosis and treatment guidance. Although the cost of human WGS is now at the $1000 mark, WES will likely remain the main workhorse in clinical settings owing to (1) its lower cost, (2) its ability to capture almost all actionable genetic defects within exons, and (3) a smaller data size that reduces the computational burden for analysis. However, the main drawback of WES has been the low accuracy in the discovery and genotyping of CNVs, which has two main reasons. First, depth of coverage is not uniform among exons, making it very difficult to apply read depth–based methods. Second, the read pairs and reads often do not span CNV breakpoints, which is needed for read pair–based and split read–based approaches, respectively. Therefore, it is often necessary to complement WES studies with alternative approaches such as array comparative genomic hybridization or quantitative RT-PCR.

We specifically designed our new algorithm, DECoNT, to address this limitation as a CNV call *polisher*. Using a deep learning approach, we were able to boost the precision of several widely used state-of-the-art algorithms that use WES data for CNV discovery. Although we trained DECoNT using matched WGS and WES samples from the 1000 Genomes Project, we also showed that

the performance gain is independent from the training data, the capture kit, and the sequencing platform. The trained models are portable and can be used off the shelf. That is, the users can directly download our models released on GitHub and feed the results of the WES-based callers along with the WES read depth to obtain polished CNV calls. It is not necessary to retrain DECoNT.

CNV is an important cause of genetic diseases that may be difficult to characterize in clinical settings without specific assays. WES is a powerful method to genotype small mutations, but so far, it has been unsuccessful to discover large CNVs that have a more direct effect in gene losses. DECoNT aims to help ameliorate high FDR problems related to CNV characterization using WES, including integer copy number prediction. Therefore, DECoNT adds an important type of genomic variation discovery to the capabilities of WES and enhances the genome analysis arsenal in the clinic.

DECoNT uses WGS-derived CNV calls as labels for training. Note that these labels cannot serve as the ground truth but rather as the semi-ground truth. Unfortunately, there exists no sufficiently large hand-curated labeled data for training a model. Chaisson et al. (2019) provide hand-curated CNV calls on nine samples which let us perform only validation. Although a larger sample set was used, the latest release by HGSVC (Byrska-Bishop et al. 2021) contains only 674 CNVs, which is a very small number for training DECoNT. We attempted training a model with this data set but failed. For comparison, the model we used and presented in this study to polish XHMM calls used about 68,000 CNV calls during training. The HGSVC CNV call set can also be regarded as a consensus call set (i.e., result of a consensus caller) because it is generated by using three different calling pipelines and supported by long-read sequencing, StrandSeq, and optical mapping analysis of a subset of these genomes. Although these hand-curated high-quality data sets and consensus callers are certainly going to help DECoNT to achieve higher precision, they are quite limited in size, which currently prohibits training. We foresee that with increasing size of future call sets, DECoNT's performance will also increase. Yet, it is evident that CNV calling on WGS data is more accurate compared with CNV calling on WES data even when only one WGS-based CNV caller is used. Thus, DECoNT transfers these higher-confidence labels into the WES domain, and it is limited by the precision of the underlying WGS-based CNV caller (CNVnator in this case) (Abyzov et al. 2011). Abyzov et al. (2011) report that CNVnator has high sensitivity (86%–96%) and low FDR (3%–20%). This corresponds to a precision range of 80% to 97% (mean 11.8%). Note that Abyzov et al. (2011) obtain these performance results on two high-coverage (20×–32×) trios in the 1000 Genomes Pilot Project data set (The 1000 Genomes Project Consortium 2010), whereas DECoNT uses the newer 1000 Genomes Project data set at 30× coverage generated using NovaSeq (Byrska-Bishop et al. 2021). Thus, we expect the accuracy of CNVnator calls we use in this study to be higher than the above-mentioned values.

The next challenge will be relieving DECoNT from the dependence on existing variation callers and making it a standalone, highly accurate CNV discovery tool using WES data. One possible such direction for DECoNT could be redesigning the architecture to work with bin-level data. That is, most baseline callers first analyze read depth in small bins, and then adjacent bins are smoothed/combined via a segmentation algorithm and are returned as final set of calls if sufficient evidence exists along multiple neighboring bins. Currently DECoNT works with this final call set, which is limiting. Working with bin-level data will require an architecture that can handle one-base-pair resolution, whereas now DECoNT works with kilobase-sized windows that are averaged.

Methods

Data set

For training and testing of DECoNT, we used 802 samples from the 1000 Genomes Project that have CNVNator calls available (i.e., HG00096 to HG02356, when sample IDs are alphabetically ordered) (Byrska-Bishop et al. 2021; ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/1000_genomes_project/data/). For these samples, we obtain both WES and WGS data. WES samples were captured using the NimbleGen SeqCap EZ Exome v3 as a capture kit and sequenced to an average of 50× depth with Illumina Genome Analyzer II and Illumina HiSeq 2000 platforms. The average read length is 76 bp. Reads were aligned to the GRCh38 using the BWA-MEM aligner (Li 2013). WGS samples were also sequenced using the same platforms with an average read length of 100 bp. Average depth coverage for this set is 30×. For XHMM, CoNIFER, and CODEX2, the ground-truth CNV calls are obtained using the CNVnator (Abyzov et al. 2011) tool call set (ftp://ftp.1000genomes.ebi.ac.uk/vol1/ftp/data_collections/1000G_2504_high_coverage_SV/working/20190825_Yale_CNVnator/). For Control-FREEC and CNVkit, the ground-truth exact CNV events are obtained using mrCaNaVaR (Alkan et al. 2009).

WES reads for NA12878 sample were obtained from SRA with accession codes SRX5191370, SRX5191369, SRX5180030, and SRX5180221 for NovaSeq 6000, HiSeq 4000, BGISEQ-500, and MGISEQ-2000, respectively.

For tools that output a categorical prediction of a CNV, we also use a highly validated CNV call set published by Chaisson et al. (2019) as another validation source. The WGS CNV calls in this call set are thoroughly validated. That is, they were obtained via a consensus of 15 different WGS CNV callers with comparisons against high-quality PB-SVs that have a single-base breakpoint resolution. We obtain WGS CNV calls for these nine samples from the 1000 Genomes Project data set (i.e., HG00512, HG00513, HG00514, HG00731, HG00732, HG00733, NA19238, NA19239, NA19240). We also obtained aligned WES reads of these samples, with the exception of HG00514, for which no WES data were available. This data set is only used for testing. The calls regarding this data set are released in the NCBI dbVar (https://www.ncbi.nlm.nih.gov/dbvar/) under accession nstd152.

DECoNT model

Problem formulation

Let X denote the set of CNV events detected on the WES data set by a WES-based CNV caller, and $X^{(i)}$ denote the i^{th} event. F_i denotes the set of features we use for $X^{(i)}$, which contains the following information: (1) the chromosome in which the CNV event occurred ($X^{(i)}_{chr}$), (2) the start coordinate of the CNV event ($X^{(i)}_{start}$), (3) the end coordinate of the CNV event, (4) the type (e.g., deletion) of the called event ($X^{(i)}_{call}$), and (5) the read depth vector between $X^{(i)}_{start}$ to $X^{(i)}_{end}$ ($X^{(i)}_{RDSeq}$). Let $Y^{(i)}_{gt}$ denote the ground-truth label obtained from the WGS CNV call for $X^{(i)}$. There are two cases: (1) for the tools that predict the existence of an event $X^{(i)}_{gt} \in \{0, 1, 2\}$, denoting no call, deletion, or duplication, respectively; and (2) for the tools that predict the copy number $Y^{(i)}_{gt} \in Z^{\geq}$. Then, the problem at hand is formulated as a classification task for 1, and as a regression task for 2. That is, our goal is to learn a function $f(F_1, \cdots, F_n) \rightarrow (Y^{(1)}_{pr}, \cdots, Y^{(n)}_{pr})$ such that the

difference between $(Y_{pr}^{(1)}, \cdots, Y_{pr}^{(n)})$ and $(Y_{gt}^{(1)}, \cdots, Y_{gt}^{(n)})$ is minimized with respect to a loss function. Here, $n = |X|$ and $Y_{pr}^{(i)}$ is the predicted label for $X^{(i)}$, and it is in the same domain as $Y_{gt}^{(i)}$ in respective tasks.

DECoNT architecture

DECoNT is an end-to-end multi-input neural network designed for polishing and improving the performance of the WES-based germline CNV callers. It is capable of improving accuracy of WES CNV calling for both exact CNV prediction (i.e., integer) and categorical CNV prediction cases (i.e., deletion, duplication, or no call). For each CNV caller, a distinct network is trained. DECoNT's pipeline for the categorical CNV prediction case can be divided into three main building blocks. First, a data preprocessing step extracts the read depth for genomic regions of interest (i.e., CNV call regions made by the CNV caller). It also normalizes the read depth sequence and acts as a regularizer for the model. The resulting read depth information is −1 padded to the length of the longest call sequence and masked. Second, a bidirectional LSTM network (BiLSTM) inputs the read depth sequence and extracts the required encoded features (i.e., embeddings). This subnetwork has 128 neurons in each direction and is followed by a batch normalization layer. Third, a two-layered FC neural network inputs the embedding calculated by Bi-LSTM, concatenated with the prior CNV prediction of the CNV caller (a one-hot-encoded vector). The first FC layer has 100 neurons and uses ReLU activation. The output layer has three neurons, and it calculates the posterior probability of each event via softmax activation: no call, deletion, or duplication. We use weighted cross-entropy as the loss function. This architecture has a total of 160,351 parameters, 159,837 of which are trainable. The rest are the batch normalization parameters. For a training data set of N samples, the formulation of DECoNT can be summarized as follows:

$$X_{encoding1}^{(1:N)} = \text{BatchNorm}(\text{BiLSTM}^{(128)}(\text{Batchnorm}(\text{Mask}(X_{RDSeq}^{(1:N)})))) \tag{1}$$

$$X_{encoding2}^{(1:N)} = \text{CAT}(X_{encoding1}^{(1:N)}, X_{call}^{(1:N)}) \tag{2}$$

$$X_{encoding3}^{(1:N)} = \text{ReLU}(\text{FC}^{(100)}(X_{encoding2}^{(1:N)})) \tag{3}$$

$$Y_{pr}^{(1:N)} = \text{Softmax}(\text{FC}^{(3)}(X_{encoding3}^{(1:N)})) \tag{4}$$

where $\text{BiLSTM}^{(\cdot)}$ represents the bidirecitonal LSTM layer with $\cdot$ hidden units in each direction. Similarly, $\text{FC}^{(\cdot)}$represents a dense layer with $\cdot$ neurons. ReLU and BatchNorm stand for rectified linear unit activation function and batch normalization, respectively.

Using $Y_{pr}^{(1:N)}$ and $Y_{gt}^{(1:N)}$ training phase minimizes the categorical cross-entropy loss. We use Adam optimizer (Kingma and Ba 2014) with a minibatch size of 128 samples. All weights in the network are initialized using Xavier initialization (Glorot and Bengio 2010).

DECoNT's pipeline for the exact (i.e., integer) CNV prediction is almost the same as the one described above. The first difference is instead of taking the one-hot encoded version of the CNV call, it inputs an integer value representing the called copy number. The second difference is at the output layer. Instead of three neurons with softmax activation, this version has a single neuron with ReLU activation to perform regression instead of classification. It has a total of 160,149 parameters, 159,635 of which are trainable. Again, the rest are the batch normalization parameters. So, the last layer in the formulation above (Equation 4) is replaced by the following layer, and in this case, $Y_{pr}^{(1:N)} \in Z^{\geq}$.

$$X_{pr}^{(1:N)} = \text{ReLU}(\text{FC}^{(1)}(X_{encoding3}^{(1:N)})) \tag{5}$$

Using the $Y_{pr}^{(1:N)}$ and $Y_{gt}^{(1:N)}$ training phase now minimizes the mean AE loss. Again, we use Adam optimizer with a minibatch size of 128 samples, and we use Xavier initialization for weights.

Polishing the state-of-the-art WES-based germline CNV callers

We polish the CNV calls made by four state-of-the-art WES-based germline CNV callers: (1) XHMM (Fromer et al. 2012), (2) CoNIFER (Krumm et al. 2012), (3) CODEX2 (Jiang et al. 2018), (4) Control-FREEC (Boeva et al. 2012), and (5) CNVkit (Talevich et al. 2016). Tools XHMM, CoNIFER, and CODEX2 perform categorical CNV prediction, and Control-FREEC and CNVkit perform exact CNV prediction. We use calls made on the WGS samples by CNVnator (Abyzov et al. 2011) as the ground-truth call set for discrete predictions and use the exact copy number predictions made by mrCaNaVaR as the ground-truth call set for integral prediction (i.e., Control-FREEC). First, DECoNT obtains the results of these tools (XHMM, CoNIFER, CODEX2, and Control-FREEC). Then, it learns to correct these calls on a portion of the 1000 Genomes Project data set using ground-truth calls. Finally, on the left-out test portion of the data, we compare the performance of the CNV callers before and after polishing by DECoNT.

Settings for the WES-based CNV callers

We follow the recommended settings for the WES-based callers. For XHMM, the parameters are set as follows: (1) $Pr(\text{start DEL}) = Pr(\text{start DUP}) = 1 \times 10^{-8}$, (2) mean number of targets in CNV (geometric distribution) = 6, (3) mean distance between targets within CNV (exponential decay) = 70 kbp, and (4) DEL, diploid (i.e., no event), and DUP read depth distributions are modeled as $\sim \mathcal{N}(-3,1)$, $\sim \mathcal{N}(0,1)$, and $\sim \mathcal{N}(3,1)$, respectively. Also, for XHMM nBins parameter is set as 200, which is the default setting. We performed the PCA normalization step of XHMM in all results reported. For CODEX2, minimum read coverage of 20 was enforced at the filtering step. Then, the algorithm automatically chooses its parameter, K, using a Bayesian information criterion (BIC) and Akaike information criterion (AIC). CoNIFER performs SVD on the data matrix and then removes n singular vectors with n largest singular values. We set n to six. Control-FREEC has 45 parameters, which were all set to default values as previously described (Boeva et al. 2012). CNVkit uses a rolling median technique to recenter each on- or off-target bin with other bins of similar GC content, repetitiveness, target size or distance from other targets, independently of genomic location (Talevich et al. 2016). We used recommended settings for CNVkit as well where log_2 read depth below threshold = −5, above threshold = 1.0.

Training settings for DECoNT

We train a DECoNT model for each of the above-mentioned tools. The set X of CNV calls per tool is shuffled and divided into training, validation, and testing sets that contain 70%, 20%, and 10% of the data, respectively. The number of events in the test sets are 6832 (3,102,221 no-calls, 2098 duplications, 1633 deletions), 81,761 (67,885 no-calls, 3042 duplications, 10,834 deletions), 180 (85 no-calls, 43 duplications, 52 deletions), 20,482 (minimum copy number is zero, maximum copy number is 585), and 39,720 (minimum copy number is 34, maximum copy number is zero) for XHMM, CODEX2, CoNIFER, Control-FREEC, and CNVkit, respectively. The second input of the algorithm is the read depth for the CNV-associated regions on the WES data. We calculate it using the

Sambamba tool (Tarasov et al. 2015). For all tools other than CODEX2, DECoNT is trained up to 30 epochs with early stopping by checking the loss on the validation fold. Training for CODEX2 has a maximum epoch number 60. For training, DECoNT uses final CNV calls (i.e., concatenated bin-level calls) made by the CNV callers.

Performance metrics

The tools XHMM, CoNIFER, and CODEX2 predict CNVs either as deletion or duplication. The main problem of these callers are FDRs (Tan et al. 2014; Zare et al. 2017). Given a deletion or duplication call by XHMM, CoNIFER, and CODEX2, DECoNT outputs a probability for the call to be deletion, duplication, or no call (i.e., false discovery). The option with the highest probability is returned as the prediction.

To assess the performance of XHMM, CoNIFER, and CODEX2 before and after being polished, we calculate the following performance metrics using $Y_{pr}^{(1:N)}$ and $Y_{gt}^{(1:N)}$: (1) duplication call precision; (2) deletion call precision, and (3) overall precision. We first define the following variables: $TP_1 :=$ number of duplications correctly identified; $TP_2 :=$ number of deletions correctly identified; $FP_1 :=$ number of duplications incorrectly identified; and $FP_2 :=$ number of deletions incorrectly identified.

Then, the performance metrics are defined as follows:

$$\text{Duplication call precision} = \frac{TP_1}{TP_1 + FP_1} \tag{6}$$

$$\text{Deletion call precision} = \frac{TP_2}{TP_2 + FP_2} \tag{7}$$

$$\text{Overall precision} = \frac{TP_1 + TP_2}{TP_1 + TP_2 + FP_1 + FP_2} \tag{8}$$

To test DECoNT's performance on exact CNV prediction problem, which is a regression task, we use *AE* between the predicted and ground-truth copy number values. For an event X_i, $AE^{(i)}$ is defined as follows:

$$AE^{(i)} = |Y_{pr}^{(i)} - Y_{gt}^{(i)}| \tag{9}$$

Time performance

All models are trained on a SuperMicro SuperServer 4029GP-TRT with two Intel Xeon gold 6140 processors (2.3 GHz, 24.75 M cache), 251 GB RAM, three NVIDIA GeForce RTX 2080 Ti (11 GB, 352 bit) and one NVIDIA TITAN RTX GPU (24 GB, 384 Bit). We used 4 GPUs in parallel to train all five models, and total training times were approximately as follows: ~70, 12, 95, 50, and 20 h for XHMM, CoNIFER, CODEX2, Control-FREEC, and CNVkit, respectively. Note that training is performed offline. The average polishing time per sample is in the order of seconds, for all models.

Polishing samples from other sequencing platforms

The training data we use is obtained using Illumina Genome Analyzer II and Illumina HiSeq 2000 machines. We check if models trained on the 1000 Genomes Project data can be used to polish CNV calls made on WES samples obtained using other sequencing platforms or capture kits that have not been seen by DECoNT.

We obtain the WES data for the sample NA12878, sequenced using four different platforms: (1) Illumina NovaSeq 6000; (2) Illumina HiSeq 4000; (3) BGISEQ-500; and (4) MGISEQ-2000. Reads are aligned to the reference genome (GRCh38) using BWA (Li 2013) with *-mem* option and default parameters. Average depth coverage for these samples is 241×, 395×, 328×, and 129×, respectively. We use these four samples only for testing. All considered WES-based CNV callers are used to call CNV events on these four WES samples with default parameters. Using the CNVnator calls obtained on the WGS sample for NA12878 as the ground truth, we measure the performance of the CNV callers before and after polishing with DECoNT. Note that NA12878 data are not included in the training data set in any form.

Polishing other WES-based CNV caller algorithms

In our framework, a separate DECoNT model is trained for every WES-based germline CNV caller. We check if a DECoNT model trained using calls made by one algorithm can be used to polish the calls made by others in the absence of a trained model.

We use the same three models trained with the settings described in the Results subsection "Overview of DECoNT" for XHMM, CoNIFER, and CODEX2. For each tool-specific DECoNT model, we polish the calls made by others. Here the training and testing folds are again exclusive. For testing, we use the same test folds for each tool as described in "Overview of DECoNT." This experiment results in six tests (i.e., for two-way comparison among every tool pair). We measure the performance of the polishing procedure using duplication precision, deletion precision, and overall precision to obtain 18 performance results in total.

Data access

The input of the models we use to train the models are (1) CNV calls made by third-party CNV callers and (2) the calculated read depth data. These are available at Zenodo (https://zenodo.org/record/6539897#.YnwL4y8RpMN) inside the respective folders of the analysis.

DECoNT is implemented and released at GitHub (https://github.com/ciceklab/DECoNT) under a CC-BY-NC-ND 4.0 international license. All custom Python scripts that were used to generate matched WGS and WES CNV data are also available on the GitHub page. The scripts used to generate the data for all figures and tables in the manuscript and the source code are provided at Zenodo (https://zenodo.org/record/6539897#.YnwL4y8RpMN) and as Supplemental Code.

Competing interest statement

The authors declare no competing interests.

Acknowledgments

We thank Vijay Kumar and Dr. Santhosh Girirajan for their help with obtaining results on CNLearn. We also thank Dr. Girirajan for the feedback on our manuscript. A.E.C. acknowledges the funding from the Türkiye Bilimler Akademisi Üstün Başarılı Genç Bilim İnsanı Ödülleri Programı (TUBA GEBIP), Bilim Akademisi–Genç Bilim İnsanları Ödül Programı, and TUSEB Aziz Sancar Research Incentive awards.

Author contributions: A.E.C. and C.A. designed and supervised the study. A.E.C. and F.O. designed the model. F.O. implemented the software and performed the experiments. A.E.C., C.A., and F.O. wrote the manuscript.

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

The 1000 Genomes Project Consortium. 2015. A global reference for human genetic variation. *Nature* **526:** 68–74. doi:10.1038/nature15393

Abyzov A, Urban AE, Snyder M, Gerstein M. 2011. CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome Res* **21:** 974–984. doi:10.1101/gr.114876.110

Alkan C, Kidd JM, Marques-Bonet T, Aksay G, Antonacci F, Hormozdiari F, Kitzman JO, Baker C, Malig M, Mutlu O, et al. 2009. Personalized copy number and segmental duplication maps using next-generation sequencing. *Nat Genet* **41:** 1061–1067. doi:10.1038/ng.437

Belkadi A, Bolze A, Itan Y, Cobat A, Vincent QB, Antipenko A, Shang L, Boisson B, Casanova J-L, Abel L, et al. 2015. Whole-genome sequencing is more powerful than whole-exome sequencing for detecting exome variants. *Proc Natl Acad Sci* **112:** 5473–5478. doi:10.1073/pnas.1418631112

Boeva V, Popova T, Bleakley K, Chiche P, Cappo J, Schleiermacher G, Janoueix-Lerosey I, Delattre O, Barillot E. 2012. Control-FREEC: a tool for assessing copy number and allelic content using next-generation sequencing data. *Bioinformatics* **28:** 423–425. doi:10.1093/bioinformatics/btr670

Byrska-Bishop M, Evani US, Zhao X, Basile AO, Abel HJ, Regier AA, Corvelo A, Clarke WE, Musunuri R, Nagulapalli K, et al. 2021. High coverage whole genome sequencing of the expanded 1000 genomes project cohort including 602 trios. bioRxiv doi:10.1101/2021.02.06.430068

Chaisson MJ, Sanders AD, Zhao X, Malhotra A, Porubsky D, Rausch T, Gardner EJ, Rodriguez OL, Guo L, Collins RL, et al. 2019. Multi-platform discovery of haplotype-resolved structural variation in human genomes. *Nat Commun* **10:** 1784. doi:10.1038/s41467-018-08148-z

Cooper GM, Coe BP, Girirajan S, Rosenfeld JA, Vu TH, Baker C, Williams C, Stalker H, Hamid R, Hannig V, et al. 2011. A copy number variation morbidity map of developmental delay. *Nat Genet* **43:** 838–846. doi:10.1038/ng.909

The Deciphering Developmental Disorders Study. 2015. Large-scale discovery of novel genetic causes of developmental disorders. *Nature* **519:** 223–228. doi:10.1038/nature14135

De Rubeis S, He X, Goldberg AP, Poultney CS, Samocha K, Cicek AE, Kou Y, Liu L, Fromer M, Walker S, et al. 2014. Synaptic, transcriptional and chromatin genes disrupted in autism. *Nature* **515:** 209–215. doi:10.1038/nature13772

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

Glorot X, Bengio Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics* (ed. Teh YW, Titterington M), Vol. 9, pp. 249–256. PMLR, Chia Laguna Resort, Sardinia, Italy.

Guo G, Sun X, Chen C, Wu S, Huang P, Li Z, Dean M, Huang Y, Jia W, Zhou Q, et al. 2013. Whole-genome and whole-exome sequencing of bladder cancer identifies frequent alterations in genes involved in sister chromatid cohesion and segregation. *Nat Genet* **45:** 1459–1463. doi:10.1038/ng.2798

Heinzen EL, Need AC, Hayden KM, Chiba-Falek O, Roses AD, Strittmatter WJ, Burke JR, Hulette CM, Welsh-Bohmer KA, Goldstein DB, et al. 2010. Genome-wide scan of copy number variation in late-onset Alzheimer's disease. *J Alzheimers Dis* **19:** 69–77. doi:10.3233/JAD-2010-1212

Hieronymus H, Murali R, Tin A, Yadav K, Abida W, Moller H, Berney D, Scher H, Carver B, Scardino P, et al. 2018. Tumor copy number alteration burden is a pan-cancer prognostic factor associated with recurrence and death. *eLife* **7:** e37294. doi:10.7554/eLife.37294

Ho SS, Urban AE, Mills RE. 2020. Structural variation in the sequencing era. *Nat Rev Genet* **21:** 171–189. doi:10.1038/s41576-019-0180-9

Hochreiter S, Schmidhuber J. 1997. Long short-term memory. *Neural Comput* **9:** 1735–1780. doi:10.1162/neco.1997.9.8.1735

Iafrate AJ, Feuk L, Rivera MN, Listewnik ML, Donahoe PK, Qi Y, Scherer SW, Lee C. 2004. Detection of large-scale variation in the human genome. *Nat Genet* **36:** 949–951. doi:10.1038/ng1416

Jiang Y, Wang R, Urrutia E, Anastopoulos IN, Nathanson KL, Zhang NR. 2018. CODEX2: full-spectrum copy number variation detection by high-throughput DNA sequencing. *Genome Biol* **19:** 202. doi:10.1186/s13059-018-1578-y

Kadalayil L, Rafiq S, Rose-Zerilli MJ, Pengelly RJ, Parker H, Oscier D, Strefford JC, Tapper WJ, Gibson J, Ennis S, et al. 2015. Exome sequence read depth methods for identifying copy number changes. *Brief Bioinformatics* **16:** 380–392. doi:10.1093/bib/bbu027

Karczewski KJ, Francioli LC, Tiao G, Cummings BB, Alföldi J, Wang Q, Collins RL, Laricchia KM, Ganna A, Birnbaum DP, et al. 2020. The mutational constraint spectrum quantified from variation in 141,456 humans. *Nature* **581:** 434–443. doi:10.1038/s41586-020-2308-7

Kebschull JM, Zador AM. 2015. Sources of PCR-induced distortions in high-throughput sequencing data sets. *Nucleic Acids Res* **43:** e143. doi:10.1093/nar/gkv717

Kingma DP, Ba J. 2014. Adam: a method for stochastic optimization. arXiv:1412.6980 [cs.LG]. doi:10.48550/arXiv.1412.6980

Krumm N, Sudmant PH, Ko A, O'Roak BJ, Malig M, Coe BP, NHLBI Exome Sequencing Project, Quinlan AR, Nickerson DA, Eichler EE. 2012. Copy number variation detection and genotyping from exome sequence data. *Genome Res* **22:** 1525–1532. doi:10.1101/gr.138115.112

Kumaran M, Cass CE, Graham K, Mackey JR, Hubaux R, Lam W, Yasui Y, Damaraju S. 2017. Germline copy number variations are associated with breast cancer risk and prognosis. *Sci Rep* **7:** 14621. doi:10.1038/s41598-017-14799-7

Levy D, Ronemus M, Yamrom B, Lee Y-H, Leotta A, Kendall J, Marks S, Lakshmi B, Pai D, Ye K, et al. 2011. Rare de novo and transmitted copy-number variation in autistic spectrum disorders. *Neuron* **70:** 886–897. doi:10.1016/j.neuron.2011.05.015

Li H. 2013. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv:1303.3997 [q-bio.GN].

Macintyre G, Goranova TE, De Silva D, Ennis D, Piskorz AM, Eldridge M, Sie D, Lewsley L-A, Hanif A, Wilson C, et al. 2018. Copy number signatures and mutational processes in ovarian carcinoma. *Nat Genet* **50:** 1262–1270. doi:10.1038/s41588-018-0179-8

Mhawech-Fauceglia P, Cheney RT, Schwaller J. 2006. Genetic alterations in urothelial bladder carcinoma: an updated review. *Cancer* **106:** 1205–1216. doi:10.1002/cncr.21743

Pankratz N, Dumitriu A, Hetrick KN, Sun M, Latourelle JC, Wilk JB, Halter C, Doheny KF, Gusella JF, Nichols WC, et al. 2011. Copy number variation in familial Parkinson disease. *PLoS One* **6:** e20988. doi:10.1371/journal.pone.0020988

Pounraja VK, Jayakar G, Jensen M, Kelkar N, Girirajan S. 2019. A machine-learning approach for accurate detection of copy number variants from exome sequencing. *Genome Res* **29:** 1134–1143. doi:10.1101/gr.245928.118

Reid BM, Permuth JB, Chen YA, Fridley BL, Iversen ES, Chen Z, Jim H, Vierkant RA, Cunningham JM, Barnholtz-Sloan JS, et al. 2019. Genome-wide analysis of common copy number variation and epithelial ovarian cancer risk. *Cancer Epidemiol Biomarkers Prev* **28:** 1117–1126. doi:10.1158/1055-9965.EPI-18-0833

Satterstrom FK, Kosmicki JA, Wang J, Breen MS, De Rubeis S, An J-Y, Peng M, Collins R, Grove J, Klei L, et al. 2020. Large-scale exome sequencing study implicates both developmental and functional changes in the neurobiology of autism. *Cell* **180:** 568–584.e23. doi:10.1016/j.cell.2019.12.036

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

Singh T, Neale B, Daly M, Schizophrenia Exome Meta-Analysis Consortium. 2019. Initial results from the meta-analysis of the whole-exomes of over 20,000 schizophrenia cases and 45,000 controls. *Eur Neuropsychopharmacol* **29:** S813–S814. doi:10.1016/j.euroneuro.2017.08.057

Sudmant PH, Mallick S, Nelson BJ, Hormozdiari F, Krumm N, Huddleston J, Coe BP, Baker C, Nordenfelt S, Bamshad M, et al. 2015. Global diversity, population stratification, and selection of human copy-number variation. *Science* **349:** aab3761. doi:10.1126/science.aab3761

Talevich E, Shain AH, Botton T, Bastian BC. 2016. CNVkit: genome-wide copy number detection and visualization from targeted DNA sequencing. *PLoS Comput Biol* **12:** e1004873. doi:10.1371/journal.pcbi.1004873

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

Tarasov A, Vilella AJ, Cuppen E, Nijman IJ, Prins P. 2015. Sambamba: fast processing of NGS alignment formats. *Bioinformatics* **31:** 2032–2034. doi:10.1093/bioinformatics/btv098

Trost B, Walker S, Wang Z, Thiruvahindrapuram B, MacDonald JR, Sung WW, Pereira SL, Whitney J, Chan AJ, Pellecchia G, et al. 2018. A comprehensive workflow for read depth-based identification of copy-number

variation from whole-genome sequence data. *Am J Hum Genet* **102:** 142–155. doi:10.1016/j.ajhg.2017.12.007

Williamson MP, Elder PA, Shaw ME, Devlin J, Knowles MA. 1995. *P16 (cdkn2)* is a major deletion target at 9p21 in bladder cancer. *Hum Mol Genet* **4:** 1569–1577. doi:10.1093/hmg/4.9.1569

Yu Y, Si X, Hu C, Zhang J. 2019. A review of recurrent neural networks: LSTM cells and network architectures. *Neural Comput* **31:** 1235–1270. doi:10.1162/neco_a_01199

Zare F, Dow M, Monteleone N, Hosny A, Nabavi S. 2017. An evaluation of copy number variation detection tools for cancer using whole exome sequencing data. *BMC Bioinformatics* **18:** 286. doi:10.1186/s12859-017-1705-x

Zarrei M, Burton CL, Engchuan W, Young EJ, Higginbotham EJ, MacDonald JR, Trost B, Chan AJ, Walker S, Lamoureux S, et al. 2019. A large data resource of genomic copy number variation across neurodevelopmental disorders. *NPJ Genom Med* **4:** 26. doi:10.1038/s41525-019-0098-3

Zhang L, Shi J, Ouyang J, Zhang R, Tao Y, Yuan D, Lv C, Wang R, Ning B, Roberts R, et al. 2021. X-CNV: genome-wide prediction of the pathogenicity of copy number variations. *Genome Med* **13:** 132. doi:10.1186/s13073-021-00945-4

Received December 2, 2020; accepted in revised form May 13, 2022.