CNV-P: A machine-learning framework for predicting high confident copy number variations

Taifu Wang [Equal first author, 1], **Jinghua Sun** [Equal first author, 1, 2], **Xiuqing Zhang** [1, 2, 3], **Wen-Jing Wang** [Corresp., 1], **Qing Zhou** [Corresp. 1]

[1] BGI-Shenzhen, Shenzhen 518083, China

[2] College of Life Sciences, University of Chinese Academy of Sciences, Beijing 100049, China

[3] Guangdong Enterprise Key Laboratory of Human Disease Genomics, Beishan Industrial Zone, Shenzhen, 518083, China

Corresponding Authors: Wen-Jing Wang, Qing Zhou
Email address: wangwenjing@genomics.cn, zhouqing1@genomics.cn

Background: Copy-number variants (CNVs) have been recognized as one of the major causes of genetic disorders. Reliable detection of CNVs from genome sequencing data has been a strong demand for disease research. However, current software for detecting CNVs has high false-positive rates, which needs further improvement.

Methods: Here, we proposed a novel and post-processing approach for CNVs prediction (CNV-P), a machine-learning framework that could efficiently remove false-positive fragments from results of CNVs detecting tools. A series of CNVs signals such as read depth (RD), split reads (SR) and read pair (RP) around the putative CNV fragments were defined as features to train a classifier.

Results: The prediction results on several real biological datasets showed that our models could accurately classify the CNVs at over 90% precision rate and 85% recall rate, which greatly improves the performance of state-of-the-art algorithms. Furthermore, our results indicate that CNV-P is robust to different sizes of CNVs and the platforms of sequencing.

Conclusions: Our framework for classifying high-confident CNVs could improve both basic research and clinical diagnosis of genetic diseases.

CNV-P: A machine-learning framework for predicting high confident copy number variations

Taifu Wang[1*], Jinghua Sun[1,2*], Xiuqing Zhang[1,2,3], Wen-Jing Wang[1#], Qing Zhou[1#]

[1] BGI-Shenzhen, Shenzhen 518083, China
[2] College of Life Sciences, University of Chinese Academy of Sciences, Beijing 100049，China
[3] Guangdong Enterprise Key Laboratory of Human Disease Genomics, Beishan Industrial Zone, Shenzhen, 518083, China

Corresponding Author:
Qing Zhou [1]
China National GenBank, Jinsha Road, Shenzhen, Guangdong, 518120, China
Email address: zhouqing1@genomics.cn

Abstract

Background: Copy-number variants (CNVs) have been recognized as one of the major causes of genetic disorders. Reliable detection of CNVs from genome sequencing data has been a strong demand for disease research. However, current software for detecting CNVs has high false-positive rates, which needs further improvement.

Methods: Here, we proposed a novel and post–processing approach for CNVs prediction (CNV-P), a machine-learning framework that could efficiently remove false-positive fragments from results of CNVs detecting tools. A series of CNVs signals such as read depth (RD), split reads (SR) and read pair (RP) around the putative CNV fragments were defined as features to train a classifier.

Results: The prediction results on several real biological datasets showed that our models could accurately classify the CNVs at over 90% precision rate and 85% recall rate, which greatly improves the performance of state-of-the-art algorithms. Furthermore, our results indicate that CNV-P is robust to different sizes of CNVs, as well as the platforms of sequencing.

Conclusions: Our framework for classifying high-confident CNVs could improve both basic research and clinical diagnosis of genetic diseases.

Introduction

Copy number variations (CNVs) are one of the genetic variations and sequence polymorphisms that widely exist in the human genome. Research shows that CNVs are closely related to the pathogenesis and development of many human diseases such as autism, Parkinson and other neurological diseases (Hollox et al. 2008; Pankratz et al. 2011; Rosenfeld et al. 2010; Sebat et al. 2007). Therefore, accurate detection of CNVs is essential for the diagnosis and research of such diseases.

With the rapid development of high-throughput sequencing technology, genomic sequencing-based technology for CNVs detection has gradually become a leading method owing to its high speed, high resolution, and high repeatability. Many sequencing-based CNVs detection methods have been proposed (Kosugi et al. 2019; Pirooznia et al. 2015; Zhao et al. 2013). Typical CNVs detection approaches mainly utilize three signatures to detect CNVs: read depth (RD), read pairs (RP), split read (SR) (Pirooznia et al. 2015). RD means the number of reads that encompass or overlap CNVs. For example, a deletion indicates a decrease in the average depth of this area. RP refers to the distribution of the insert size of the sequenced library. If the mapping distance of read pairs significantly deviates from the average value of the sequencing library, such discordant alignment features herald the occurrence of CNVs. SR

indicates the split (soft-clipped) alignment features of reads that span CNVs. The initial strategies for detecting CNVs mainly focused on one of these features (Abyzov et al. 2011; Chen et al. 2009; Medvedev et al. 2010). Most of the approaches have high false-positive rates because of the noises of sequencing data, such as sequencing error and artificial chimeric reads. Ambiguous mapping of reads from repeat- or duplication-rich regions also decreases the accuracy of CNVs (Kosugi et al. 2019; Teo et al. 2012). Consequently, tools integrating multiple features to detect CNVs have been gradually developed (Bartenhagen & Dugas 2016; Layer et al. 2014; Rausch et al. 2012), while their performance still needs be further modified (Kosugi et al. 2019).

To identify high confident CNVs, a commonly used strategy is setting a cutoff value or applying various statistical distributions to filter fragments. This strategy greatly depends on the expertise of researchers and their subjective assumptions about the analyzed data. Another strategy uses the intersection of CNVs generated by two or more algorithms. However, due to various CNV-property-dependent and library-property-dependent features used by different detection methods, they usually provide inconsonant results. Thus, a large number of potentially true CNVs could be discarded. Additionally, some tools, such as MetaSV(Mohiyuddin et al. 2015), Parliament2 (Zarate et al. 2020) and FusorSV (Becker et al. 2018), use the method of integrating and merging CNVs from multiple software. These approaches require output results of several certain tools, usually more than four software, while reanalyzing CNVs using their default methods is impractical and time-consuming.

Here, we developed a machine-learning framework for CNVs prediction (CNV-P), aiming to accurately predict CNVs from the results of present software. CNV-P collected three aforementioned signatures (RD, RP and SR) and other information of the putative CNVs. The results of our model on real data demonstrate that CNV-P greatly improves the performance of state-of-the-art algorithms

Materials & Methods

Data download and preprocessing

The gold-standard sets of CNVs from 9 individuals (NA19238, NA19239, NA19240, HG00512, HG00513, HG00514, HG00731, HG00732, HG00733) were downloaded from *Chaisson et al* (Chaisson et al. 2019). The whole genome sequencing (WGS) data (~30x) of these 9 individuals were downloaded from the National Center for Biotechnology Information (NCBI) with an accession number SRP159517 (***Table S1, S2***). For external validation samples, the sequencing data of NA12878 and HG002 were also downloaded from NCBI with accession

numbers SRP159517 and SRP047086 respectively. The gold-standard CNVs of NA12878 were generated by three data sets: the Database of Genomic Variants (http://dgv.tcag.ca/dgv/app/home?ref=GRCh37/hg19) (R. et al. 2013), the 1000 Genomes Project phaseIII (https://ftp.ncbi.nih.gov/1000genomes/ftp/phase3/integrated_sv_map) (Sudmant et al. 2015), and the CNVs of PacBio data from *Pendleton, M. et al* (Pendleton et al. 2015). The gold-standard CNVs of HG002 were downloaded from *Zook, J.M., et al* (Zook et al. 2019).

For the above gold-standard sets, we excluded other types of CNVs except for deletion and duplication, removed CNVs shorter than 100bp and merged fragments with over 80% reciprocal overlaps. On average, each sample had more than 10,000 CNVs after processing (***Table S1***). For WGS data, the clean reads after removing adapter and filtering low-quality reads were aligned to the human genome reference (hg19) with bwa (Li 2013) 'mem' command to generate the BAM file. All of these datasets were generated by standard WGS protocol, with libraries of approximate 400bp insert size and average ~30X coverage (***Table S2***).

Generate simulated dataset

We generate random CNVs (range from 100bp to 100kb) based on a copy of human genome (hg19) using mason2 (Holtgrewe 2010). To avoid the same or similar CNVs between training data and test data, we selected fragments on chromosome 1 and chromosome 2 as training samples and CNVs on chromosome 3 and chromosome 4 as testing samples (more details in ***Table S1***). Then, the paired-end sequencing reads (100bp) from the altered genome was simulated by wgsim (Li 2011), with an insert size of 500bp and 0.001 base error rate.

Training set and test set

We chose five common software to obtain the initial sets of CNVs for simulated data and the downloaded sequencing data (deletions and duplications): Lumpy (Layer et al. 2014), Manta (Chen et al. 2015), Pindel (Ye et al. 2009), Delly (Rausch et al. 2012) and breakdancer (Chen et al. 2009). The details of running parameters were shown in the supplemental methods section. The original CNVs were then performed as follows: 1. Removed other types of CNVs except for deletion and duplication. 2. Removed CNVs with > 10 bp overlapped with N region of human genome (download from http://genome.ucsc.edu/). 3. Merged CNVs with ≥80% reciprocal overlaps and kept the union part of fragments. 4. Removed CNVs that less than 100bp. Then, we labeled these treated CNVs as either 'True' or 'False' based on their overlapped part with gold-standard CNVs. CNVs having ≥80% reciprocal overlap with the gold-standard CNVs in simulated data were labeled as "True" and the cutoff was set to ≥50% for sequencing data. We

then selected data of 6 individuals as a training set and the other 3 samples as a test set, including two dependent validation datasets (more details in ***Table S1, S2***).

Feature extraction

We chose commonly used signals by detection tools as features in our training model, such as read depth, information of paired and spited read, mapping quality and GC content of CNVs, as well as all these features around CNV's boundaries (***Table S3***). Training features were obtained from the alignment results (***Fig. 1***) in BAM format, which was generated by a read aligner that supports partial read alignments, such as BWA-MEM (Li 2013). For read depth-based and GC content-based features, we computed the read depth and GC rate of three regions: 500 bp upstream and downstream of the left breakpoint $\boldsymbol{L}_{\mathbf{b1k}}$, 500 bp upstream and downstream of the right breakpoint $\boldsymbol{R}_{b1k}$, and the region from start to end $\boldsymbol{C}_{start\text{-}end}$. Read depth was calculated by total number of aligned bases divided by the length of the region. We then normalized the read depth by the average coverage of entire genome and processed log2 transformation to eliminate the impact of fluctuations in sequencing depth. GC content was also calculated in these three regions. Thus, using read depth and GC content of the three local regions ($\boldsymbol{L}_{\mathbf{b1k}}$, $\boldsymbol{R}_{b1k}$ and $\boldsymbol{C}_{start\text{-}end}$), six features were defined. Split-read, read pair and mapping quality were computed for two regions: $\boldsymbol{L}_{\mathbf{b1k}}$ and $\boldsymbol{R}_{b1k}$. Split read-based features were defined as the number of clipped reads within the area $\boldsymbol{L}_{\mathbf{b1k}}$ or $\boldsymbol{R}_{b1k}$. Read pair–based features were defined as the number of outlier reads pair within $\boldsymbol{L}_{\mathbf{b1k}}$ or $\boldsymbol{R}_{b1k}$. Normally, The insert size of a normal paired-end read should be within $\boldsymbol{m}_{is} \pm \boldsymbol{n}\boldsymbol{\sigma}_{is}$, where $\boldsymbol{m}_{is}$ and $\boldsymbol{\sigma}_{is}$ are the median and standard deviation of insert size, respectively, and $\boldsymbol{n}$ is the number of standard deviation from the median(we set is to 3). In addition to aberrant insert size, we also calculated the number of reads without pair within the area $\boldsymbol{L}_{\mathbf{b1k}}$ or $\boldsymbol{R}_{b1k}$. The features of mapping quality were defined as the number of reads with mapping quality <10 within $\boldsymbol{L}_{\mathbf{b1k}}$ or $\boldsymbol{R}_{b1k}$. Finally, we also normalized the value of split reads, read pair and mapping quality according to the mean value of genome coverage. Besides, we included the size and type of CNVs as training features, since the efficacy of CNVs could vary for different size ranges and types (duplication/deletion).

Comparison with CNV-JACG, MetaSV and hard cutoff method

We compared the performance of CNV-P with that of CNV- JACG (Zhuang et al. 2020), MetaSV (Mohiyuddin et al. 2015) and hard cutoff method in the same datasets. Since MetaSV currently does not support Delly's output, only four CNV detection tools (Lumpy, Manta, Pindel,

and breakdancer) were taken into consideration. CNV-JACG was conducted running with default parameters (details in supplementary methods). MetaSV was carried out with complete mode. For hard cutoff method, we used SR and RP as the evidence to support the existence of CNVs, therefore, the number of SR and RP greater than 2, 5, and 10 were set as hard cutoff to evaluate. SURVIVOR(Jeffares et al. 2017) was used to merge fragments with 80% overlap after filtering by CNV-P, CNV- JACG, MetaSV and hard cutoff method.

Methodology evaluation

we calculated the classifier performance on the test dataset in terms of precision and recall (TP: true positive, TN: true negative, FP: false positive, FN: False negative)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1 score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Also, we plotted the ROC with the AUC for model evaluation. ROC curves were drawn based on a series of false positive rates (FPR) and true positive rates (TPR).

Results

Study overview

In this study, we built a random forest (RF) framework for the CNVs prediction base on both simulated and real datasets (***Fig. 1A***). Firstly, we identified CNVs using five common tools (Lumpy, Manta, Pindel , Delly and breakdancer). For each set, we removed CNVs with low quality or locating on the N region of the human genome (details in **Methods**). Secondly, we labeled CNVs as either “True” or “False” based on a 50% reciprocal overlap with the gold-standard CNVs in real data and 80% reciprocal overlap in simulated data respectively (details in **Methods**). Next, we extracted the signatures around these CNVs such as RD, SR and RP as training features from alignment results (***Fig. 1B***).

We then split the data set into a training set and a test set. Based on the training set, we trained a RF classifier to identify CNVs as “true” or “false”. We performed 10-repeated 10-fold cross-validation for optimal parameter selection and used the receiver operating characteristic (ROC) curve to quantify the prediction performance. Next, we evaluated the robustness of our models on test data from multiple aspects, such as sizes of CNVs and platforms of raw

sequencing data. We also compared the performance of Support Vector Machine (SVM) and Gradient Boosting classifier (GBC) with our random forest model. Finally, we validated our model on two extra data sets.

Performance of CNV-P on a simulated dataset

We train RF, GBC, and SVM classifiers for CNV prediction based on a simulated dataset (details in **Methods**). The results show that there was a significant improvement after CNV-P prediction compared with the original CNV results. The precision of CNVs produce by each CNV detection tools improved from 49.58% to 99% with almost zero loss recall rate (***Fig. 2A, B***). Compared with GBC and SVM classifiers, RF was slightly superior. The RF classifiers for 5 tools achieved comparable performance since an average increasing of F1-score was about 14.55% for Lumpy, 14.34% for Manta, 13.84% for Pindel, 11.10% for Delly and 16.16% for breakdancer (***Fig. 2C***).

Performance of CNV-P on a real dataset

In this part, we trained RF classifier for the five selected tools respectively based on real samples. The 10-repeated 10-fold cross-validation was performed for optimal parameter selection (***Fig. S1***). The overall diagnostic ability of each classifier was measured as the area under the receiver operating characteristic curve (AUC) for the test dataset. The highest value of AUC was 97.10% for the model of Lumpy while the model for Pindel had the smallest value of 93.62% (***Fig. 3A***). Each classifier accurately classified the CNVs as either true or false at 91.76-95.17% precision and 87.75-96.54% recall rate (***Fig. 3B***). After processing by CNV-P, a large number of false-positive CNVs were removed, and the majority of true CNVs were remained (***Fig. 3C***).

To dissect the principle of the CNV-P classifier, we assessed the relative importance of each feature for corresponding classifiers. As expected, for all classifiers, read-depth provided the most discriminatory power to make accurate predictions (***Fig. S2***). However, the second important feature was inconsistent between different classifiers. It was probably due to various detection algorism these tools used.

To evaluate the robustness of CNV-P, we trained each model on various proportions of training data (from 10% to 90% in increments of 20%). The results showed a steady improvement in accuracy (precision and recall rate) with an increase in the number of training data (***Fig. S3***).

We further assessed the performance of CNV-P for CNVs of different sizes. We divided

CNVs into three sets based on their length: CNV_S (100 bp to 1 kb; bp: base pair, kb: kilobase), CNV_M (1 kb to 100 kb) and CNV_L (>100kb). The overall precisions were greatly improved, comparing with the raw CNVs achieved by the corresponding software (***Fig. 3D***). We noticed that almost all precision and recall rates of CNV_S and CNV_M were over 90%, while theses values of CNV_L were slightly lower. These results are probably caused by the insufficient number of CNV_L in our training data.

We also profiled the distribution of predicted probability scores for all CNVs within a different size range. Since CNVs with a probability score >0.5 were classified as true in our CNV-P prediction results, we found that the threshold of 0.5 distinguished true and false CNVs very well (***Fig. S4***). Besides, the probability scores could be used as a measurement of confidence for a certain fragment of CNVs, which would provide support evidence in further analysis.

Furthermore, we implanted two additional models, GBC and SVM, to train CNV-P classifiers. Comparing the precision and recall values, as well as the result of ROC curve, we found they had comparable performance (***Fig. S5***). Still, the RF classifier was recommended as the first choice with a slight superiority.

Prediction on external data sets

To further evaluate the performance of CNV-P, we implemented our models on two independent WGS datasets of NA12878 and HG002 (***Table S1***). Since we had proved that increasing the size of training data could improve the accuracy of our model (***Fig. S3***), the final classifiers were trained on both the training set and test set mentioned above. Consistent with the above results, CNV-P produced the optimal performance with AUCs of 0.89-0.95 in NA12878 (***Fig. 4A***). Most of the false-positive CNVs were removed with a loss of a small number of true positive fragments **(*Fig 4B, C*)**. Likewise, our approach had a similar performance on sample HG002 (***Fig. 4D-F***).

We next compared CNV-P with other post-process tools for CNV filtering, including CNV-JACG and MetaSV. We also included commonly used hard filtering method, setting cut-off of SR and RP number for each CNV. We applied various filtering approaches on NA12878 and HG002, and then evaluated fragments using gold-standard CNVs of these two samples. Our results showed that CNV-P had the highest F1-score among all the post-process methods (***Table 1***).

Besides, we evaluated the performance of our approach in data generated from multiple sequencing platforms. With precision of 91.6-96.8% and recall rates of 84.1-94% (***Fig. 5***), CNV-P showed similar performance on sequencing data generated by BGI-500. Moreover, in addition

to the trained classifiers for the above five software, we provided extra modules in our approach for training and predicting if CNVs were detected by other tools. These results suggest that our approach is suitable for CNVs generated from multiple sequencing platforms and detecting software.

Discussion

Detecting CNVs from WGS is error-prone because of short-length reads and library-property-dependent bias [5]. Inflated false positive makes it a big challenge for researchers to identify clinically relevant CNVs, as it is time- and money-consuming to validate a large amount of false positive CNVs. To solve this problem, we develop CNV-P, an effective machine-learning-based framework to acquire high-confident CNVs. Instead of handling the shortcomings of existing methods by developing another detecting algorithm, CNV-P focuses on providing a reliable set of CNVs from existing detection software. We demonstrate that CNV-P can identify a set of high-confidence CNVs with high precision and recall rates. Moreover, CNV-P is robust to the proportion of variants in training sets, size of CNVs and sequencing platforms, indicating the utility of CNV-P in a variety of clinical or research contexts.

Comparing with the conventional method of using hard cutoff, such as a minimum number of supporting reads, to filtering CNV results, CNV-P greatly reduces errors caused by lack of expertise and subjective assumptions. Instead of running default multiple software in advance, CNV-P can make accurate predictions for each tool dependently. In addition to the five commonly used software that we have trained prediction models, we provide an extra module in CNV-P including the function of model training and predicting if CNVs are detected by other tools.

However, our models may have weaker power for large-size CNVs, because there are only a small number of large fragments in our training data. Besides of data from healthy individuals, we believe that great improvement could be made to identify large-size true CNVs in the future when more datasets are accumulated.

Conclusions

CNV-P is a well-performed machine-learning framework for accurately filtering CNVs. CNV-P framework can be applied on CNVs from various detection methods and sequencing platforms, making our framework easy to adopt and customize. CNV-P greatly helps to generate a set of high-confident CNVs, benefiting both basic research and clinical diagnosis of genetic diseases.

Availability of data and materials

All data generated or analyzed during this study are included in this published article and its supplementary information files.
CNV-P is available at https://github.com/wonderful1/CNV-P.

Acknowledgements

The authors thank Dr. Jian Guo for constructive comments on this project and Chen Ye for data download and management.

References

Abyzov A, Urban AE, Snyder M, and 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. 10.1101/gr.114876.110

Bartenhagen C, and Dugas M. 2016. Robust and exact structural variation detection with paired-end and soft-clipped alignments: SoftSV compared with eight algorithms. *Brief Bioinform* 17:51-62. 10.1093/bib/bbv028

Becker T, Lee WP, Leone J, Zhu Q, Zhang C, Liu S, Sargent J, Shanker K, Mil-Homens A, Cerveira E, Ryan M, Cha J, Navarro FCP, Galeev T, Gerstein M, Mills RE, Shin DG, Lee C, and Malhotra A. 2018. FusorSV: an algorithm for optimally combining data from multiple structural variation detection methods. *Genome Biol* 19:38. 10.1186/s13059-018-1404-6

Chaisson MJP, Sanders AD, Zhao X, Malhotra A, Porubsky D, Rausch T, Gardner EJ, Rodriguez OL, Guo L, Collins RL, Fan X, Wen J, Handsaker RE, Fairley S, Kronenberg ZN, Kong X, Hormozdiari F, Lee D, Wenger AM, Hastie AR, Antaki D, Anantharaman T, Audano PA, Brand H, Cantsilieris S, Cao H, Cerveira E, Chen C, Chen X, Chin CS, Chong Z, Chuang NT, Lambert CC, Church DM, Clarke L, Farrell A, Flores J, Galeev T, Gorkin DU, Gujral M, Guryev V, Heaton WH, Korlach J, Kumar S, Kwon JY, Lam ET, Lee JE, Lee J, Lee WP, Lee SP, Li S, Marks P, Viaud-Martinez K, Meiers S, Munson KM, Navarro FCP, Nelson BJ, Nodzak C, Noor A, Kyriazopoulou-Panagiotopoulou S, Pang AWC, Qiu Y, Rosanio G, Ryan M, Stutz A, Spierings DCJ, Ward A, Welch AE, Xiao M, Xu W, Zhang C, Zhu Q, Zheng-Bradley X, Lowy

E, Yakneen S, McCarroll S, Jun G, Ding L, Koh CL, Ren B, Flicek P, Chen K, Gerstein MB, Kwok PY, Lansdorp PM, Marth GT, Sebat J, Shi X, Bashir A, Ye K, Devine SE, Talkowski ME, Mills RE, Marschall T, Korbel JO, Eichler EE, and Lee C. 2019. Multi-platform discovery of haplotype-resolved structural variation in human genomes. *Nat Commun* 10:1784. 10.1038/s41467-018-08148-z

Chen K, Wallis JW, McLellan MD, Larson DE, Kalicki JM, Pohl CS, McGrath SD, Wendl MC, Zhang Q, Locke DP, Shi X, Fulton RS, Ley TJ, Wilson RK, Ding L, and Mardis ER. 2009. BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. *Nat Methods* 6:677-681. 10.1038/nmeth.1363

Chen X, Schulz-Trieglaff O, Shaw R, Barnes B, and Saunders CT. 2015. Manta: Rapid detection of structural variants and indels for germline and cancer sequencing applications. *Bioinformatics* 32:1220-1222.

Hollox EJ, Huffmeier U, Zeeuwen PL, Palla R, Lascorz J, Rodijk-Olthuis D, van de Kerkhof PC, Traupe H, de Jongh G, den Heijer M, Reis A, Armour JA, and Schalkwijk J. 2008. Psoriasis is associated with increased beta-defensin genomic copy number. *Nat Genet* 40:23-25. 10.1038/ng.2007.48

Holtgrewe M. 2010. Mason@ A Read Simulator for Second Generation Sequencing Data.

Jeffares DC, Jolly C, Hoti M, Speed D, Shaw L, Rallis C, Balloux F, Dessimoz C, Bahler J, and Sedlazeck FJ. 2017. Transient structural variations have strong effects on quantitative traits and reproductive isolation in fission yeast. *Nat Commun* 8:14061. 10.1038/ncomms14061

Kosugi S, Momozawa Y, Liu X, Terao C, Kubo M, and Kamatani Y. 2019. Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing. *Genome Biol* 20:117. 10.1186/s13059-019-1720-5

Layer RM, Chiang C, Quinlan AR, and Hall IM. 2014. LUMPY: a probabilistic framework for structural variant discovery. *Genome Biol* 15:R84. 10.1186/gb-2014-15-6-r84

Li H. 2011 wgsim - Read simulator for next generation sequencing. *Github Repository [online]* http://githubcom/lh3/wgsim

Li H. 2013. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv preprint arXiv:13033997*.

Medvedev P, Fiume M, Dzamba M, Smith T, and Brudno M. 2010. Detecting copy number variation with mated short reads. *Genome Res* 20:1613-1622. 10.1101/gr.106344.110

345 Mohiyuddin M, Mu JC, Li J, Bani Asadi N, Gerstein MB, Abyzov A, Wong WH, and Lam HY.
346 2015. MetaSV: an accurate and integrative structural-variant caller for next generation
347 sequencing. *Bioinformatics* 31:2741-2744. 10.1093/bioinformatics/btv204
348 Pankratz N, Dumitriu A, Hetrick KN, Sun M, Latourelle JC, Wilk JB, Halter C, Doheny KF,
349 Gusella JF, Nichols WC, Myers RH, Foroud T, DeStefano AL, Psg P, GenePd Investigators C,
350 and Molecular Genetic L. 2011. Copy number variation in familial Parkinson disease. *Plos One*
351 6:e20988. 10.1371/journal.pone.0020988
352 Pendleton M, Sebra R, Pang AW, Ummat A, Franzen O, Rausch T, Stutz AM, Stedman W,
353 Anantharaman T, Hastie A, Dai H, Fritz MH, Cao H, Cohain A, Deikus G, Durrett RE,
354 Blanchard SC, Altman R, Chin CS, Guo Y, Paxinos EE, Korbel JO, Darnell RB, McCombie
355 WR, Kwok PY, Mason CE, Schadt EE, and Bashir A. 2015. Assembly and diploid architecture
356 of an individual human genome via single-molecule technologies. *Nat Methods* 12:780-786.
357 10.1038/nmeth.3454
358 Pirooznia M, Goes FS, and Zandi PP. 2015. Whole-genome CNV analysis: advances in
359 computational approaches. *Front Genet* 6:138. 10.3389/fgene.2015.00138
360 R. MJ, Robert Z, Yuen RKC, Lars F, and Scherer SW. 2013. The Database of Genomic Variants:
361 a curated collection of structural variation in the human genome. *Nucleic Acids Research*:D1.
362 Rausch T, Zichner T, Schlattl A, Stütz AM, Benes V, and Korbel JO. 2012. DELLY: structural
363 variant discovery by integrated paired-end and split-read analysis. *Bioinformatics* 28:i333.
364 Rosenfeld JA, Ballif BC, Torchia BS, Sahoo T, Ravnan JB, Schultz R, Lamb A, Bejjani BA, and
365 Shaffer LG. 2010. Copy number variations associated with autism spectrum disorders contribute
366 to a spectrum of neurodevelopmental disorders. *Genet Med* 12:694-702.
367 10.1097/GIM.0b013e3181f0c5f3
368 Sebat J, Lakshmi B, Malhotra D, Troge J, Lese-Martin C, Walsh T, Yamrom B, Yoon S,
369 Krasnitz A, Kendall J, Leotta A, Pai D, Zhang R, Lee YH, Hicks J, Spence SJ, Lee AT, Puura K,
370 Lehtimaki T, Ledbetter D, Gregersen PK, Bregman J, Sutcliffe JS, Jobanputra V, Chung W,
371 Warburton D, King MC, Skuse D, Geschwind DH, Gilliam TC, Ye K, and Wigler M. 2007.
372 Strong association of de novo copy number mutations with autism. *Science* 316:445-449.
373 10.1126/science.1138659
374 Sudmant PH, Rausch T, Gardner EJ, Handsaker RE, Abyzov A, Huddleston J, Zhang Y, Ye K,
375 Jun G, Fritz MH, Konkel MK, Malhotra A, Stutz AM, Shi X, Casale FP, Chen J, Hormozdiari F,

Dayama G, Chen K, Malig M, Chaisson MJP, Walter K, Meiers S, Kashin S, Garrison E, Auton A, Lam HYK, Mu XJ, Alkan C, Antaki D, Bae T, Cerveira E, Chines P, Chong Z, Clarke L, Dal E, Ding L, Emery S, Fan X, Gujral M, Kahveci F, Kidd JM, Kong Y, Lameijer EW, McCarthy S, Flicek P, Gibbs RA, Marth G, Mason CE, Menelaou A, Muzny DM, Nelson BJ, Noor A, Parrish NF, Pendleton M, Quitadamo A, Raeder B, Schadt EE, Romanovitch M, Schlattl A, Sebra R, Shabalin AA, Untergasser A, Walker JA, Wang M, Yu F, Zhang C, Zhang J, Zheng-Bradley X, Zhou W, Zichner T, Sebat J, Batzer MA, McCarroll SA, Genomes Project C, Mills RE, Gerstein MB, Bashir A, Stegle O, Devine SE, Lee C, Eichler EE, and Korbel JO. 2015. An integrated map of structural variation in 2,504 human genomes. *Nature* 526:75-81. 10.1038/nature15394

Teo SM, Pawitan Y, Ku CS, Chia KS, and Salim A. 2012. Statistical challenges associated with detecting copy number variations with next-generation sequencing. *Bioinformatics* 28:2711-2718. 10.1093/bioinformatics/bts535

Ye K, Schulz MH, Long Q, Apweiler R, and Ning Z. 2009. Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics* 25:2865-2871. 10.1093/bioinformatics/btp394

Zarate S, Carroll A, Mahmoud M, Krasheninina O, Jun G, Salerno WJ, Schatz MC, Boerwinkle E, Gibbs RA, and Sedlazeck FJ. 2020. Parliament2: Accurate structural variant calling at scale. *Gigascience* 9. 10.1093/gigascience/giaa145

Zhao M, Wang Q, Wang Q, Jia P, and Zhao Z. 2013. Computational tools for copy number variation (CNV) detection using next-generation sequencing data: features and perspectives. *Bmc Bioinformatics* 14 Suppl 11:S1. 10.1186/1471-2105-14-S11-S1

Zhuang X, Ye R, So MT, Lam WY, Karim A, Yu M, Ngo ND, Cherny SS, Tam PK, Garcia-Barcelo MM, Tang CS, and Sham PC. 2020. A random forest-based framework for genotyping and accuracy assessment of copy number variations. *NAR Genom Bioinform* 2:lqaa071. 10.1093/nargab/lqaa071

Zook JM, Hansen NF, Olson ND, Chapman LM, Mullikin JC, Xiao C, Sherry S, Koren S, Phillippy AM, Boutros PC, Sahraeian SME, Huang V, Rouette A, Alexander N, Mason CE, Hajirasouliha I, Ricketts C, Lee J, Tearle R, Fiddes IT, Barrio AM, Wala J, Carroll A, Ghaffari N, Rodriguez OL, Bashir A, Jackman S, Farrell JJ, Wenger AM, Alkan C, Soylev A, Schatz MC, Garg S, Church G, Marschall T, Chen K, Fan X, English AC, Rosenfeld JA, Zhou W, Mills RE, Sage JM, Davis JR, Kaiser MD, Oliver JS, Catalano AP, Chaisson MJ, Spies N, Sedlazeck FJ,

and Salit M. 2019. A robust benchmark for germline structural variant detection. *bioRxiv*:664623. 10.1101/664623

Figure 1

The study overview of CNV-P

A) The workflow of CNV-P framework classifyng candidate CNVs as True or False. B) The features we used to train supervised machine learning models

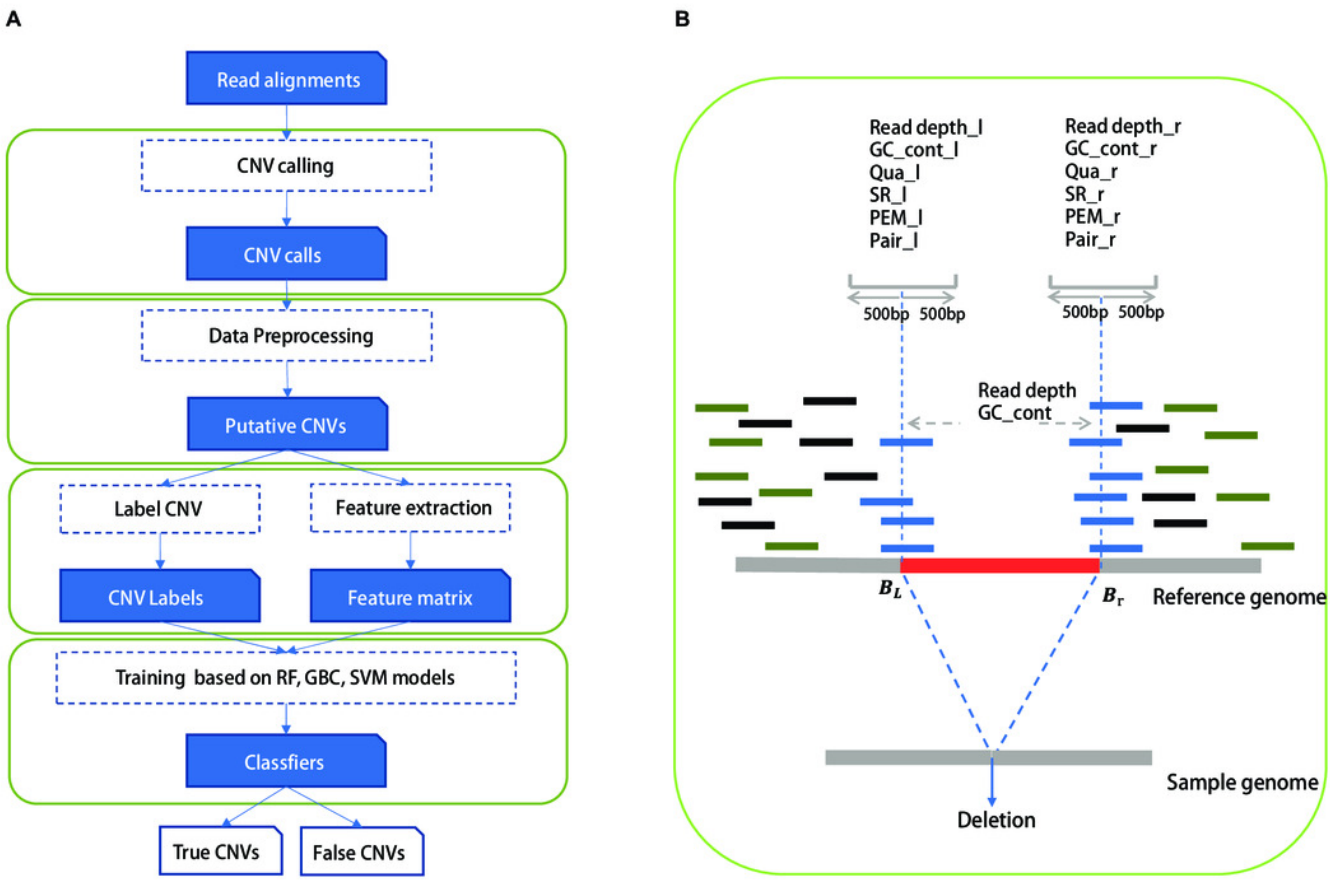

Figure 2

Performance of CNV-P on simulated dataset.

A)The F-score , B) sensitivity, and C) precision over testing simulated dataset.

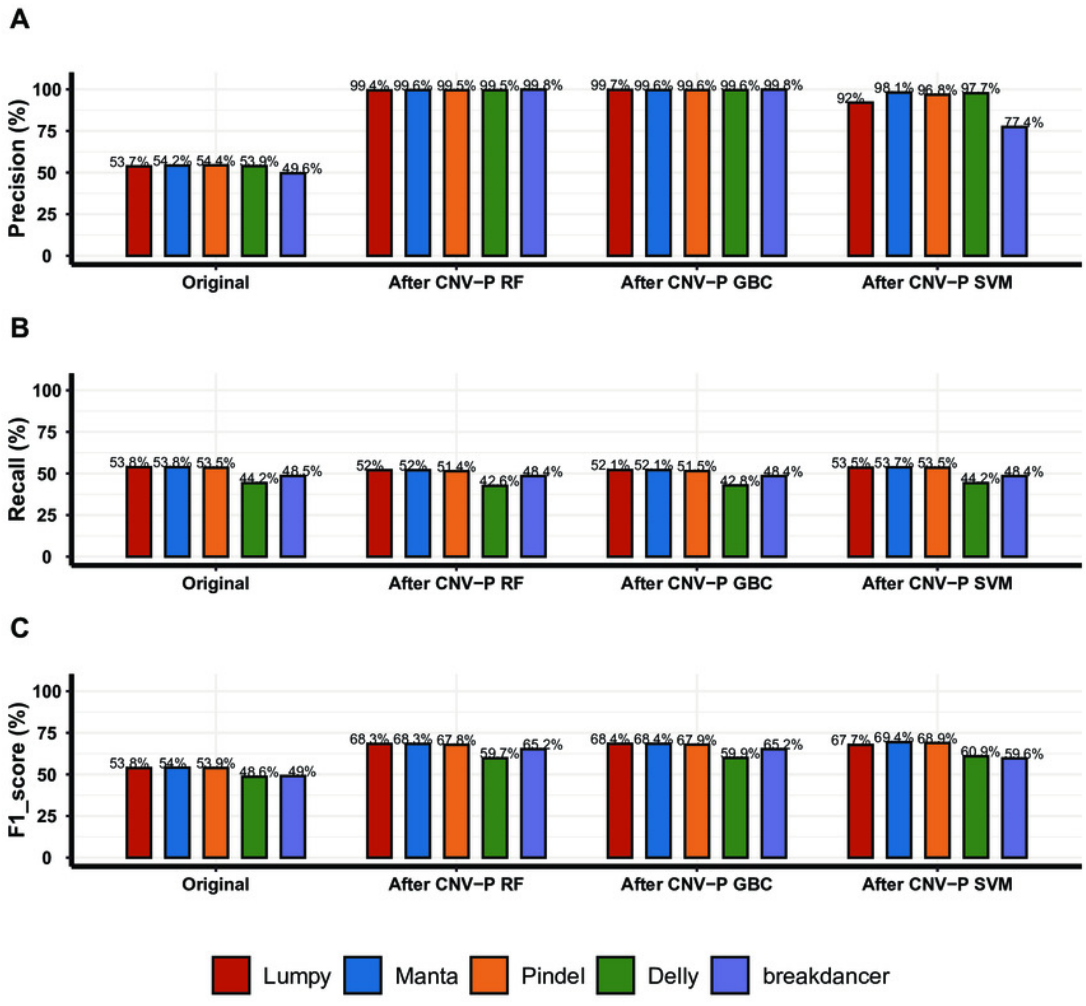

Figure 3

Performance of CNV-P on real dataset.

A) Area Under the receiver operating characteristic Curve (AUC) of CNV-P in 3 test datasets. B) The precise and recall rate of CNV-P. C) The number of CNVs before and after CNV-P predicting for five commonly used tools. D) The precise and recall rate of CNV-P at different size range of CNVs. CNV_S: 100 bp to 1 kb, CNV_M: 1 kb to 100 kb, CNV_L: >100kb.

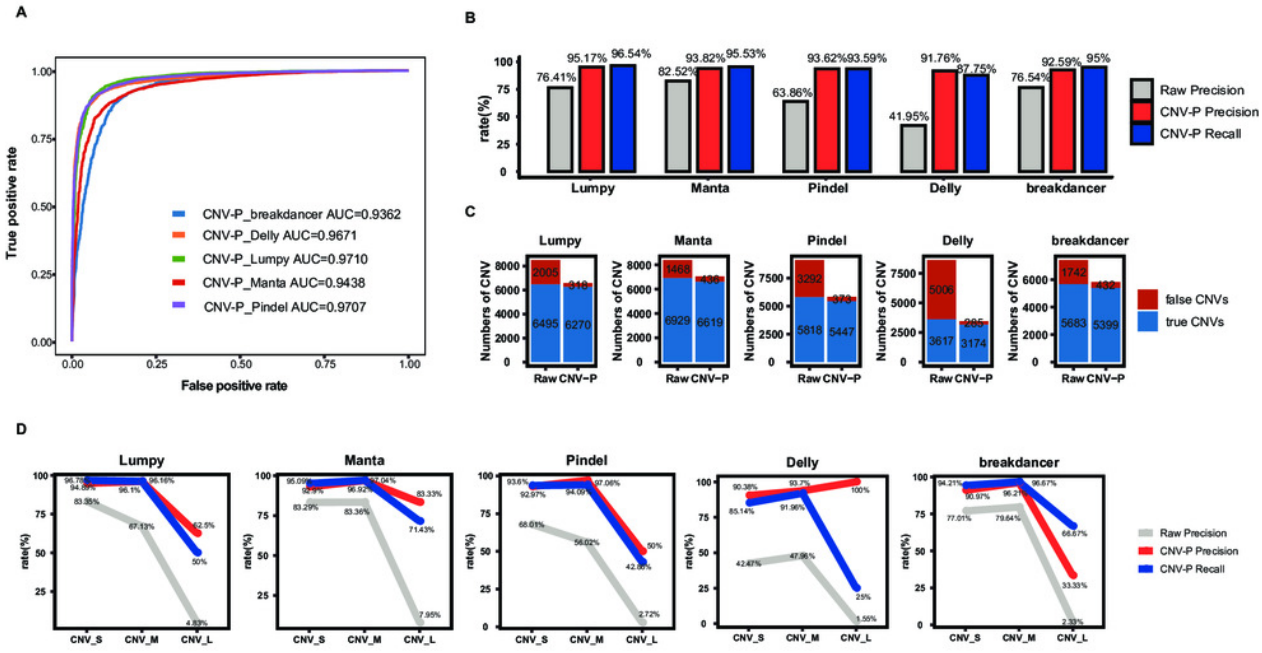

Figure 4

Performance of CNV-P on other validation dataset.

CNV-P detects high-confident CNVs with high precision and recall rates on two independent sequencing datasets from NA12878 (A, B, C) and HG002 (D, E, F). (A, D) Receiver operating characteristic (ROC) curves of CNV-P. (B, E) The precision and recall rate of CNV-P; (C, F) The number of classified CNVs by CNV-P from five commonly used tools.

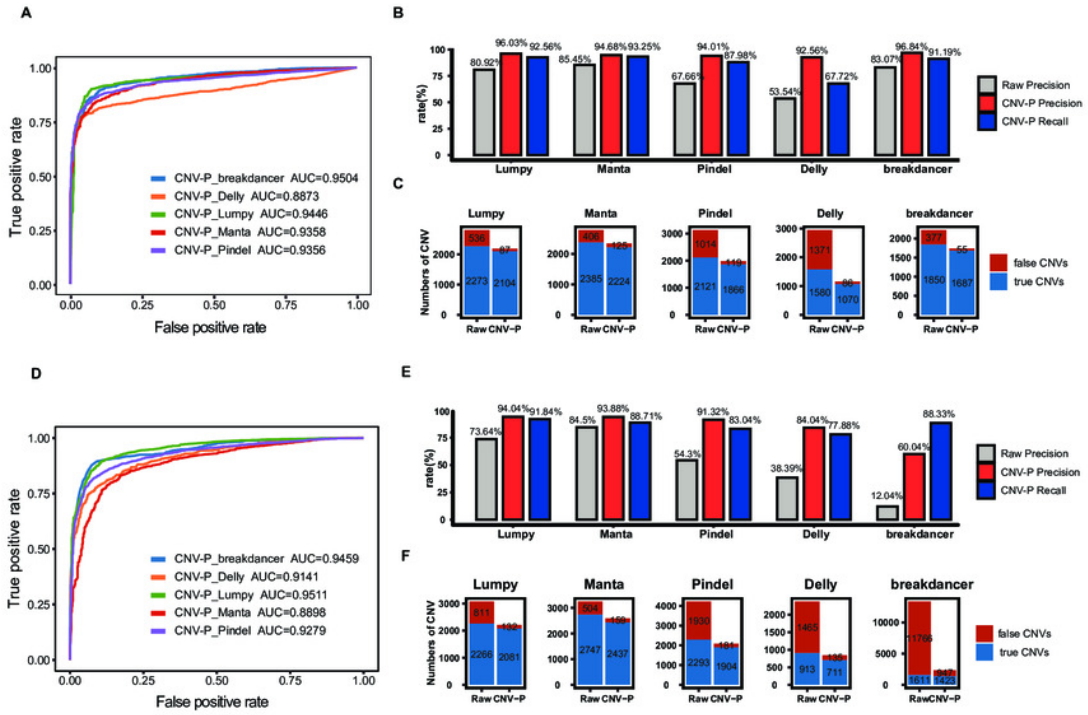

Figure 5

Performance of CNV-P on different sequencing platform.

The precise and recall rate of CNV-P for sample NA12878 using sequencing data generated from BGI-SEQ500 and Illumina. A) The raw precise results; B) Precise rate; C) Recall rate.

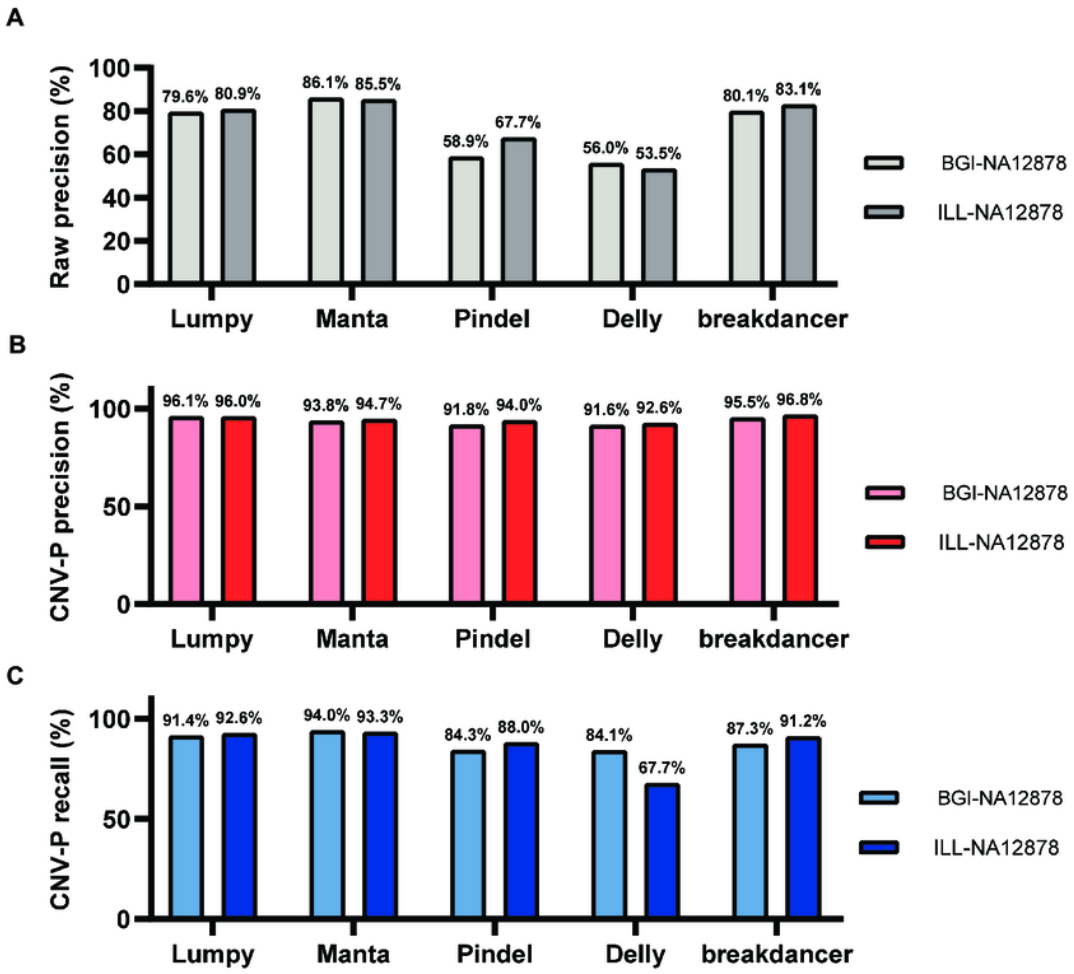

Table 1(on next page)

Comparison with CNV-JACG, MetaSV and hard cutoff method in NA12878 and HG002

Table 1: Comparison with CNV-JACG, MetaSV and hard cutoff method in NA12878 and HG002.

Sample	method	precise	recall	F1-score
NA12878	RAW	0.6032	1.0000	0.7525
	Hard_Cutoff_2	0.6197	0.9792	0.7590
	Hard_Cutoff_5	0.7145	0.8630	0.7818
	Hard_Cutoff_10	0.7780	0.6976	0.7356
	CNV-JACG	0.6828	0.7496	0.7146
	MetaSV	0.7094	0.8817	0.7862
	CNV-P	**0.9007**	**0.7977**	**0.8461**
HG002	RAW	0.2054	1.0000	0.3408
	Hard_Cutoff_2	0.4026	0.9729	0.5695
	Hard_Cutoff_5	0.5740	0.8653	0.6901
	Hard_Cutoff_10	0.6642	0.7482	0.7037
	CNV-JACG	0.5443	0.7076	0.6153
	MetaSV	0.5917	0.8274	0.6900
	CNV-P	**0.7078**	**0.7516**	**0.7290**