ClearCNV: CNV calling from NGS panel data in the presence of ambiguity and noise

May V., Koch L., Fischer-Zirnsak B., Horn D., Gehle P., Kornak U., Beule D., Holtgrewe M.

Bioinformatics
2022 AUG 15 ; 38(16): 3871-3876
2022 JUN 25 (first published online: final publication)
DOI: 10.1093/bioinformatics/btac418

Publisher: Oxford University Press

Bioinformatics, YYYY, 0–0
doi: 10.1093/bioinformatics/xxxxx
Advance Access Publication Date: DD Month YYYY
Manuscript Category

Sequence Analysis

ClearCNV: CNV calling from NGS panel data in the presence of ambiguity and noise

Vinzenz May[1], Leonard Koch[2], Björn Fischer-Zirnsak[2,3], Denise Horn[2], Petra Gehle[4,5], Uwe Kornak[3,6], Dieter Beule[1] and Manuel Holtgrewe[1,*]

[1] Berlin Institute of Health at Charité – Universitätsmedizin Berlin, Core Unit Bioinformatics (CUBI), Charitéplatz 1, 10117 Berlin, Germany
[2] Charité – Universitätsmedizin Berlin, corporate member of Freie Universität Berlin and Humboldt-Universität zu Berlin, Institute of Medical Genetics and Human Genetics, Augustenburger Platz 1, 13353 Berlin, Germany
[3] Max-Planck-Institut für Molekulare Genetik, FG Development & Disease, 14195 Berlin, Germany,
[4] Charité-Universitätsmedizin Berlin, corporate member of Freie Universität Berlin and Humboldt-Universität zu Berlin, Department of Internal Medicine - Cardiology, Berlin, Germany
[5] DZHK (German Center for Cardiovascular Research), partner site Berlin, Germany
[6] Institute of Human Genetics, University Medical Center Göttingen, Göttingen, Germany.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: While the identification of small variants in panel sequencing data can be considered a solved problem, the identification of larger, multi-exon copy number variants (CNVs) still poses a considerable challenge. Thus, CNV calling has not been established in all laboratories performing panel sequencing. At the same time such laboratories have accumulated large data sets and thus have the need to identify copy number variants on their data to close the diagnostic gap.
Results: In this manuscript we present our method clearCNV that addresses this need in two ways. First, it helps laboratories to properly assign data sets to enrichment kits. Based on homogeneous subsets of data, clearCNV identifies CNVs affecting the targeted regions. Using real-world data sets and validation, we show that our method is highly competitive with previous methods and preferable in terms of specificity.
Availability: The software is available for free under a permissible license at {{https://github.com/bihealth/clear-cnv}}
Contact: manuel.holtgrewe@bih-charite.de
Supplementary information: Supplementary data are available at Bioinformatics online.

1 Introduction

Hybrid capture methods (Ng et al., 2009) allow for targeted sequencing ranging from whole exome sequencing to panel sequencing of few known disease genes. They have thus made high throughput sequencing affordable for clinical applications by strongly reducing the required sequencing data. From the perspective of bioinformatics there are few differences in analyzing small panels, whole exome (WES), or whole genome (WGS) sequencing data for single nucleotide variants (SNVs), and small insertions and deletions.

However, the detection (commonly also referred to as calling) of copy number variants (CNVs) is considerably harder because of structured but very inhomogeneous variances in depth of coverage that are typical for hybrid capture methods. Reasons for such variance include GC content of the targeted regions, biochemical properties of the used enrichment kits, and batch effects in producing the enrichment reagents (Daniel et al. 2011, Benjamini 2012).

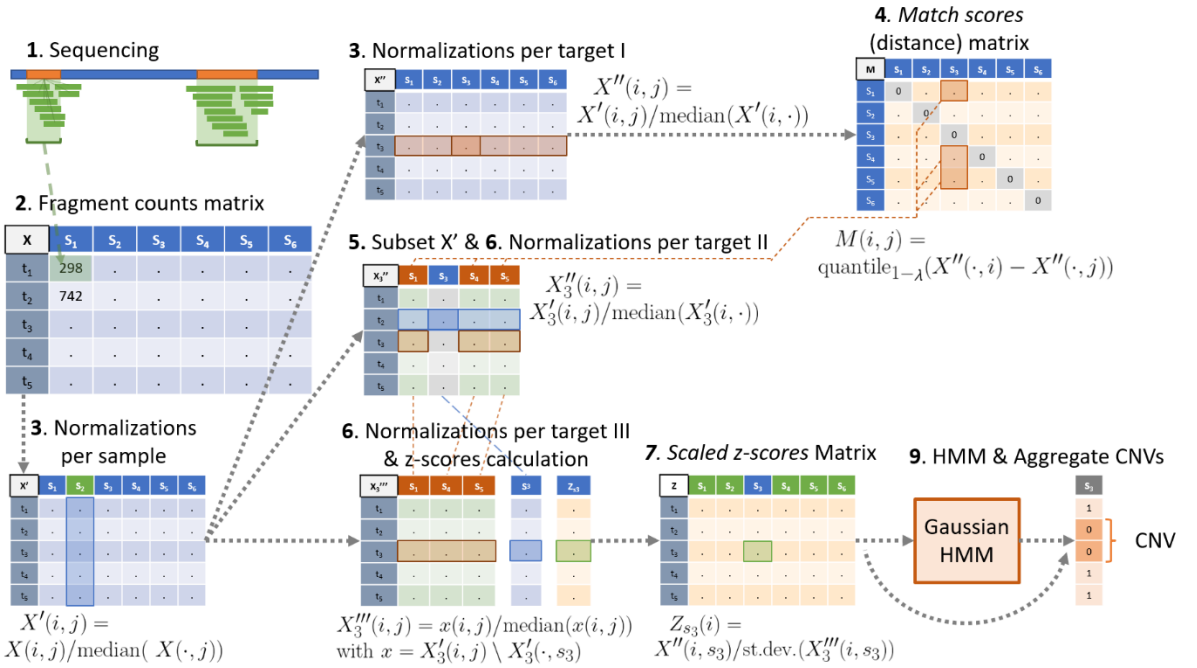

Fig. 1. CNV calling algorithm. This figure is an illustrates the steps described in the main text. In step 3, green indicates a sample that is normalized in that step. In steps 5 & 6 orange indicates the sample group background and blue the CNV calling sample. In steps 6 and 7, green indicates the calculated (and scaled) z-score.

Nevertheless, CNVs are of large interest as they account for 4.5 to 12 percent of genome variation in humans (Sudmant,P.H. et al. 2015, Collins, R.L. et al. 2020) and are implied in many diseases (Nowakowska 2017, Conrad et al. 2010, Zhang et al. 2009). Copy number variants are a subclass of structural variants; the latter is commonly defined as variation with a size of larger than 50bp. The size of CNVs can range from the lower limit to the loss or amplification of whole chromosome arms or chromosomes.

From the authors' experience, CNV calling for panel sequencing data has not been systematically established in many laboratories performing targeted panel sequencing yet despite the wide use of panel sequencing. While many centers are now introducing WES or even WGS into standard care, they have considerable numbers of panel sequencing data already available (Marshall et al. 2020). Obviously, being able to reanalyze this data for CNVs is highly desirable to solve more cases without additional sequencing.

Various tools for the CNV analysis of panel sequencing have recently been published in the literature including CoNVaDING (Johansson et al. 2016) and AtlasCNV (Chiang et al. 2019). Further, tools for the analysis of exome data have been enabled for the analysis of panel data, including panelcn.MOPS (Povysil et al. 2017) or ExomeDepth (Plagnol et al., 2012). Some methods have been developed and evaluated solely for a single panel such as AtlasCNV while others can be used more widely such as panelcn.MOPS or ExomeDepth. Approaches to combine CNV calling tools to achieve the highest possible accuracy can differ in their results by a lot given different datasets (Moreno-Cabrera et al. 2020, Sadedin et al. 2018)

However, centers wishing to analyze their panel data in hindsight often also face unexpected challenges. From the experience of the authors, these also include missing, incomplete or incorrect documentation of which gene panel or gene panel version was used for a particular sample.

In this manuscript we present our software package *clearCNV* that (1) contains a program that helps users to properly assign panel sequence data to the used panel and to separate sequencing batches, (2) provides a novel method that allows to analyze their data for copy number variations, and (3) provides an easy-to-use visualization of the coverage data and the called CNVs.

2 Methods

2.1 CNV Calling algorithm

The first part of our method is the implementation of a novel algorithm for the identification of CNVs from targeted sequencing data. Some steps are built on the ideas of already existing algorithms. The steps of the algorithm are described below and illustrated in Fig. 1.

1. Target file creation. Overlapping and nearby targets are merged to avoid ambiguities further downstream.

2. Fragment counting. We count the number of fragments (reads or read pairs) per target. Fragments overlapping with multiple targets are assigned to the one closest to the center. The results are tabulated in a matrix x with entries $X(i,j)$ in row i and column j; that is the number of fragments of sample j on target i. Samples with a median fragment count smaller than five are excluded to avoid downstream problems.

3. Data normalization. The matrix is first normalized per sample (per column) by dividing each column's values by the column's median $X'(i,j) := X(i,j) \,/\, \text{median}(X'(i,\cdot))$ and then per target (per row) $X''(i,j) := X'(i,j) \,/\, \text{median}(X'(\cdot,j))$, where "$\cdot$" indicates all elements in that dimension.

4. Match scores are a distance metric to identify samples with similar coverage patterns (similarly used in the context of CNV calling by Johannson et al. (2016). A match score m of two samples (vectors) s and k is defined as the mean difference of two vectors. clearCNV additionally removes the λ greatest differences before computing the mean: $m_{s,k} := \text{mean}(\text{quantile}_{1-\lambda}(\text{abs}(s-k)))$, where λ (default is 0.02) is a user-adjustable factor that attributes for expected uneven variance. Such a

variance includes signals for CNVs as well as forms of noise. The final visualizations (Fig. 2) help to adjust this factor.

5. *Sample group.* Each sample *S* gets assigned a set of background samples. Any sample x in each sample group satisfies that its match score $m_{x,S}$ is below the median match score of all match scores of all samples multiplied by a user-specifiable constant θ (default 2):
$m(x,S) \leq \text{median}\big(m(\cdot,\cdot)\big) \cdot \theta$. This way each sample gets assigned an individually sized sample group. We determined the default value for θ empirically by inspecting histograms of all match scores. No CNV calls are generated on a sample that has a sample group size below a user-specifiable threshold γ (default 20), however it may appear in another sample's sample group. θ and γ were determined by choosing a relative optimum between the number of samples and the variance in a sample group. In CNV calling, a subset of fragment counts normalized per sample is chosen according to the selected sample group of sample *S*. Let this table be X'_S.

6. *Data normalization II & III.* X'_S is normalized per target to get X''_S and the vector $X''_S(\cdot,S)$ in X''_S containing all values of *S* is extracted.
X''_S without *S* ($X''_S \setminus S$) is again normalized per target to remove any effect of *S* on the sample group's statistics which yields X'''_S. The 10% columns of X'''_S with greatest variance are then dropped from X'''_S to further reduce variance.

7. *Scaled z-scores.* z-scores are calculated for sample S, which is found in X''_S on each row i: $z(i) := X''_S(i,S) \,/\, \sigma(i)$, where σ is the vector of per row standard deviations of X'''_S. The resulting z-scores are then scaled to reduce the effects of noise in CNV calling: $z'(i) := z(i)^{2-\alpha}$, where z(*i*) is the z-score of target *i* of sample *S*. The value α is the user-provided factor which is 0.65 by default. We determined the default for α in comparison with plotted heatmaps and checked where most CNV calls aligned with our judgement. The resulting vector of z' is saved in a matrix *Z* which contains all scaled z-scores of all samples.

8. *r-scores* approximate a copy number of a target in a sample. r-scores are created in the previous step on the vector $X''_S(\cdot,S)$. Ideally, a r-score of 1.0 indicates a wild type, while 0.5 indicates a heterozygous deletion, 1.5 indicates a heterozygous duplication and 2.0 indicates a homozygous duplication and so on. These values are saved to a matrix *R* for each target and sample. This matrix holds all r-scores at the end.

9. *CNV calling*. Two types of CNVs are called: a) multi-exon CNVs and b) single-exon CNVs.

9a) At first, the Viterbi algorithm is used on a Gaussian HMM (Hidden Markov Model). The means of the three states (*deletion, wild type, duplication*) are semi-automatically adjusted. For *deletion*, the mean is calculated as $m_{del} = -3\,\sigma$, for *wild type* it is $m_{wt} = \bar{m}$, and for *duplication* it is $m_{dup} = 4\,\sigma$, with σ the st.dev., and $\bar{m}$ the median of all scaled z-scores. The Viterbi algorithm is then run on the z-scores in *Z*. The covariances are set to 1.0. The transition probability matrix is created from the user-adjustable transition probability τ (default τ = 0.001):

$$\begin{pmatrix} 1-(\tau*2) & \tau & \tau \\ \tau & 1-(\tau*2) & \tau \\ \tau & \tau & 1-(\tau*2) \end{pmatrix}.$$

The resulting hidden states of each sample are saved in a matrix *H* that holds all hidden states of all targets and all samples. For each sample *S*, a consecutive interval *T* of targets is aggregated to a single CNV if the average ratio score $r = \text{mean}(R(T,S))$ satisfies $r < \mu$ (default is 0.75) or $r > \omega$ (default is 1.35) and all hidden states of *H*(*T*,*S*) are the same and not the wild type. μ and ω were chosen under the assumption that they separate the ratio scores well. The score function *c* of a CNV is the absolute value of the mean of scaled z-scores of all contributing targets *c*(*T*,*S*) = mean(abs(*Z*(*T*,*S*))).

9b) To call single-exon CNVs, two thresholds are applied to *Z*. A single target *t* of sample S is called a deletion if $Z(t,S) < -3.5$ and $R(t,S) < 0.75$ or a duplication if $Z(t,S) > 4.5$ and $R(t,S) > 1.35$ and only if it is not contained in an already called multi-exon CNV by the HMM-guided method. All default values for parameters were determined by empirical methods, including comparisons with data visualizations.

10. *Output.* The CNV calls are saved in a tabular file containing the gene names, aberration, size, score, and sample score. Furthermore, the scaled Z-scores matrix *Z* and the ratio-scores matrix *R* and a list of samples that failed to have a sufficient sample group size are written to output files.

2.2 Result Visualization

The second part of our method is a web browser—based visualization for the relative copy numbers per target and per sample represented by the ratio scores, as well as the scaled z-scores. This allows the user to visually screen the results of their experiments as well as the results of the CNV calling algorithm.

Scaled z-scores and ratio-scores are both visualized in responsive heatmaps. Each heatmap additionally shows a track of mappability, target size and GC-content at each target. These are calculated from the target file, the reference and a uniqueness-of-reference file. The scaled z-scores are clipped to the interval [-6,6]. The ratio-scores are clipped to the interval [0,2]. An example of such a heat map is shown in Fig. 2.

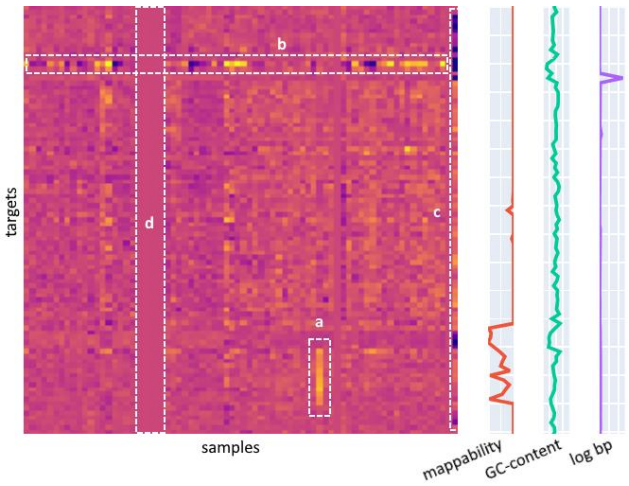

Fig. 2. Example heatmap of r-scores. This is a small example of a heatmap showing the ratio scores of each target (row) for each sample (column). A darker spot indicates a lower r-sore and vice versa. Aligned to the targets are three tracks: 1. Mappability, GC-content and log bp, which is the size of a target in bp then log transformed. Each of these three tracks show a value of 0 if the colored curve is on the left side. The Mappability and GC-content tracks have a value of 1 if the curve is on the right side. Log bp can be any size if on the right side but it scales with the maximum value in the track. Additionally, we marked several phenomena in the heatmap to illustrate its potential. a) shows a possible copy number gain, b) shows a target with high variance (or copy number variability), c) shows a low-quality sample with high variance, d) shows several samples that were too noisy and whose r-scores were set to 1.0 (imputed).

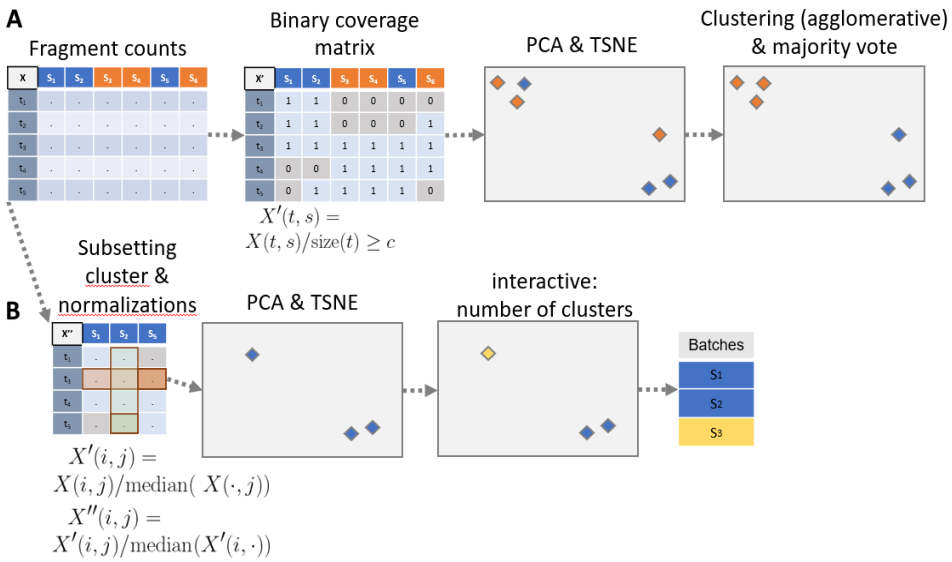

Fig. 3. Sample to Panel re-assignment and batch separation. The steps are numbered according to the main text's steps. Sub figure A illustrates the sample re-assignment and sub figure B illustrates the batch separation. *X, X′, and X″* illustrate matrices, where " . " means any numerical entry. Colors blue and orange indicate different clusters (or with yellow different batches). Frames overlaying a matrix indicate the vector that is subject to an operation.

2.3 Sample to Panel re-assignment

The third part of our method supports users in the assignment of aligned sequencing results in BAM format to panel target information in BED (Browser Extensible Format) files. This is important for retrospective analyses in the presence of artifacts such as sample swaps or erroneous documentation of the used sequencing kit. A BED file is a text file containing genomic regions, *e.g.*, exons. The following steps are illustrated in Fig. 3. The projection by PCA and TSNE, as well as the clustering can be interactively worked with in a *plotly Dash* app (https://plotly.com/dash/). When applied with clearCNV, batch separation (see 2.4) should be done before the CNV calling step.

1. *BED file merging*. The panel re-assignment algorithm starts with merging all input target files to one union of target files. This is necessary to make the given samples comparable.

2. *Fragment counting* is done the exact same way as in the CNV calling algorithm. An entry in the resulting matrix is addressed as $X(i,j)$ for the i-th row (target) and j-th column (sample).

3. *Binary matrix*. We are interested only whether a target is covered and not in the depth of coverage. Since we expect off-target effects in the enrichment, we call a target covered only if the per target fragment counts divided by target size is above 1/50. In the case of reads of 100bp size, this, for example, would correspond to a read depth coverage of two. The final matrix *X* is binary (1 = covered, 0 = uncovered).

4. *PCA and transformation*. A principal component analysis (PCA) transformation is applied to X with δ dimensions (default is 20, adjustable by the user) yielding *X'*. The default of δ is chosen according to the order of magnitude of the number of targets per panel.

5. *TSNE*. A t-distributed stochastic neighbor embedding (TSNE) projects *X'* to a latent space (here with two components) which allows fast and simple clustering on the resulting matrix *X''*. The random process within the TSNE makes it necessary for the user to occasionally re-run the process to arrive at a desired projection and clustering result.

6. *Clustering*. An agglomerative clustering on *X''* finds the clusters.

7. *Cluster assignment*. The resulting clustering is mapped to the provided target files. A majority vote is used to assign each cluster to a target file. This implies the constraint that each cluster must have a majority of correctly assigned samples to a target file.

8. *Output*. At this point the data sets are untangled and new lists of bam-files are written to the according output files.

2.4 Batch separation

We observed separable subsets in the data, which were not explained by erroneous sample–panel assignments. We suspect that limited numbers of well plate units may have introduced such batch effects. We suspected even more possible reasons such as the design of custom enrichment kits or flow cell biases.

Batch separation is done for each cluster with its previously assigned panel resulting from the sample re-assignment. The batch-separation algorithm is like the sample re-assignment algorithm. Again, the interactive parts are implemented in a *plotly Dash* app.

1. *Fragment counts sub setting*. The matrix containing the fragment counts per target is subset to the samples found in the given cluster. The targets are subset from the union BED-file to contain only targets that are present in the assigned panel. Differently to the panel re-assignment procedure, the resulting matrix is not reduced to a binary matrix but holds the per sample and per target normalized fragment counts.

2. *PCA and transformation*. Analog to panel re-assignment step 4.

3. *TSNE*. Analog to panel re-assignment step 5.

4. *Clustering*. The interactive interface lets the user control the number of batches on each set of samples for each panel.

5. *Output*. A list of alignment file paths (BAM format) per identified batch is written to an output file for each found cluster. This file can be used in the CNV calling step.

2.5 Evaluation

To evaluate the performance of our algorithm, we compared it to existing approaches to call CNVs on targeted sequencing data. The tool selection was limited to those having a scientific publication and being freely available for research applications. We chose ExomeDepth (Plagnol et al., 2012), CoNVaDING (Johansson et al. 2016), panelcn.MOPS (Povysil et al. 2017) to evaluate comparatively with clearCNV. We chose not to use

the recent tool Atlas-CNV by Chiang et al. (2019) in the evaluation because it was designed to find single Exon CNVs in the *eMERGESeq* panel, which we did not use (an earlier evaluation showed no competitive results on our data set, data not shown). A brief overview of the tools' features can be found in the supplement section S8.

We wrote CNV calls and internal or explicit scorings of clearCNV, CoNVaDING, ExomeDepth and panelcn.MOPS, each to a uniformly formatted file for a comparative evaluation.

We attempted to evaluate all four tools on simulated targeted sequencing data. The data was simulated with CapSim (Cao 2018). A detailed analysis of the generated simulated data showed that important properties and biases that we observe in real-world data are not captured appropriately in the simulation, in particular missing correlation between adjacent exons. Nevertheless, simulated data allows to evaluate tool performance with a known ground truth.

The results on the simulated data can be summarized as follows. ClearCNV is competitive with the other tools on simulated data and showed the highest positive predicted value (PPV) at the cost of some sensitivity. ExomeDepth shows good performance overall, CoNVaDING and panelCN.MPOS showed a very high false positive rate for certain samples (we were unable to determine the root cause for this).

All details regarding the simulation, analysis, and performance results can be found in the Supplementary Sections S6 and S7.

To compare the results of the CNV calling of each tool on real-world data, we used two different approaches. First, we compared the scores given to each single target in each sample for each tool. We did this before we chose a subset of CNV calls to be validated via quantitative PCR (qPCR). The details can be found in section S2 of the supplement. Second, we compared the scores of the aggregated called CNVs after we had the qPCR results. We ranked the called CNVs for each tool to achieve comparability. The ranking is described in detail in the results section. Finally, we visualized these rankings for each tool's results, which can be seen in the results section and in Fig. 5.

Regarding the results, two different parameters were scored. First, each tool scores single targets for each sample. These are the target scores. Second, CNV calls are scored and the *target scores* are the underlying scores. They are aggregated in some way, e.g., by taking the mean of the consecutive targets that form the called CNV. These are the *CNV scores*.

Each tool has a different scoring metric per target and only ExomeDepth and clearCNV aggregate called CNVs. In the case of CoNVaDING we chose to score the targets according to the median "AUTO_ZSCORE" scores which are found in the matching *.totallist files. panelcn.MOPS does also not provide aggregated CNV calls and no target scores. After very helpful correspondence with the authors, we followed their advice to calculate scores per target and used the RC ratio (RC.norm/medRC.norm) to score single targets per sample. To aggregate called CNVs, we merged consecutive targets if they were called the same copy number unequal two. clearCNV generates both the target scores and the CNV scores.

We performed our evaluation on seven custom panels manufactured by Agilent from different genetic rare disease fields. Data was generated in a diagnostic setting and all patients gave informed consent for further research. Four panels have about six thousand targets, the others have about one or two thousand. In total, we had data from 1407 different individual samples. More detailed information about the data can be found in the supplement section S1, Table S1, and Fig. S1.

We chose not to evaluate with simulated data as targeted sequencing data is known to contain a large amount of noise and biases that have not been comprehensively characterized and modeled yet.

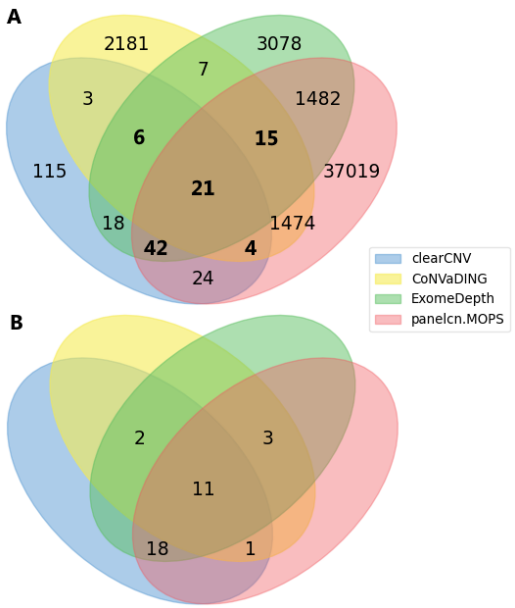

Fig. 4. Venn diagrams of called and confirmed CNVs. Sub figure A shows all CNV calls on all available data. CNVs selected for validation are marked in bold letters. Sub figure B shows only the confirmed (by qPCR) CNVs. Unlabeled subsets have a cardinality of zero.

3 Results

3.1 Sample to panel re-assignment and batch separation

38 out of 1407 total samples were re-assigned to different panels or panel versions, for which the documentation was not complete anymore. 16 Samples had a fragment-per-target count so low that they were excluded from any further processing by clearCNV. A detailed log of the sample re-assignment and batch separation process can be found in the supplement section S3.

3.2 CNV calls

The Venn diagram in Fig. 4.A shows all called CNVs on all data sets by CoNVaDING, ExomeDepth, panelcn.MOPS and clearCNV. As it can be seen, the results show a great discordance in terms of called CNVs. To select a feasible number of variants for validation by qPCR, we limited the results to those CNVs called by three tools or more. We finally had to exclude samples for which no DNA for validation was available. This resulted in a set of 88 CNV calls to be validated. We could confirm 35 CNV calls, of which 15 were duplications and 20 were deletions. One deletion could be confirmed by inspecting the corresponding WGS track in IGV highlighting extended fragment spans (see section 2 and Fig. S2 in the supplement). The other 34 CNVs were confirmed via qPCR, following the protocol detailed in Ott *et al.* (2010).

Tab. 1. CNV calls by all tools. This table shows the total number of CNV calls made by each tool and the according number of validated and confirmed CNV calls. The subsets can be inspected in Fig. 4.

Tool	Total CNV calls	validated calls	confirmed calls
clearCNV	233	73	32

CoNVaDING	3 711	46	17
ExomeDepth	4 669	84	34
panelcn.MOPS	40 081	82	33

Even after several adjustments of the DNA melting temperatures, we were not able to identify the true copy number via qPCR of nine out of the 88 CNV calls. We treated ambiguous results as unconfirmed calls (same as wild type). We analyzed mappability and GC-content of the CNV calls and our whole data. We found that for about 15% of the targets (data points) a too high GC content rendered qPCR validation infeasible. The details can be found in supplement section S4.

Fig. 4B shows a Venn diagram of the 35 confirmed CNVs. The subsets overlapping only two or one tools are left blank, as these CNV calls were excluded from validation. Eleven CNV calls were made by all four tools and were successfully validated. Three CNV calls that were confirmed by qPCR were not called by clearCNV. CoNVaDING missed 18, panelCN.MOPS missed two and ExomeDepth missed one. As can be seen in Figure 4A, the overall number of CNV calls varies greatly between the tools. Besides the total number of true positives, one must consider the rank of the true within the false positives within each tool.

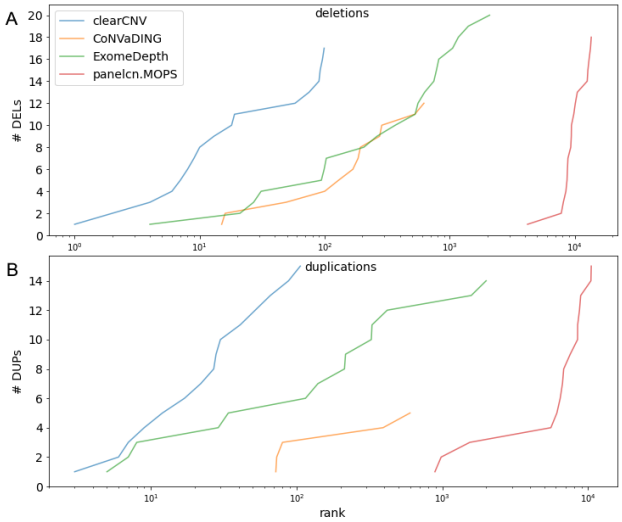

Fig. 5. CNV calling score ranks vs. cumulated number of confirmed CNVs. All CNV calls are ranked for each tool and separately for deletions (sub fig. A) and duplications (sub fig. B). Each tool's results with standard parameters are represented by a solid line.

Fig. 5 shows receiver/operator characteristics-like plots for each tool. Each tool attaches scores to its resulting CNV calls. These scores can be understood as a transformed approximation of confidence. Ranking these scores (sort, then enumerate) allows to compare each tool's results directly. The horizontal axis shows the (log10-scaled) rank of the curve and the vertical axis shows the cumulative number of positively validated CNVs. The curve ends for each tool where the lowest ranking CNV call was confirmed.

It can be seen that ExomeDepth and clearCNV both created some true positive CNV calls among the highest ranks. Also, this approximation of specificity shows a difference in calling either deletions or duplications on our data set. ExomeDepth starts similarly specifically as clearCNV. The curve flattens and reaches into the 1000th rank to find all 20 deletions. The CNV calls by CoNVaDING and panelcn.MOPS start with lower ranks and end with very low ranks. But in the case of duplications, CoNVaDING's results show even a slightly higher specificity than ExomeDepth's. Overall, clearCNV misses three CNV calls but shows superior specificity when compared to other tools. A detailed analysis of the three missed CNVs can be found in Supplement section S5. All results can be downloaded from the supplementary repository (https://github.com/bihealth/clear-cnv-supplementary).

4 Discussion

clearCNV showed competitive sensitivity and excellent specificity on different real-world data sets, which were partially very heterogeneous in the underlying batches and unknown variances (more details in Supp. section S3). High specificity is important in clinical applications as it reduces the number of false positive and effort for validation.

Differences in specificity of the different tools can be attributed to different design decisions by their authors. panelcn.MOPS was not designed to operate at a high specificity, which is an intentional choice by the authors who worked with very high quality data (see discussion on GitHub: https://github.com/bioinf-jku/panelcn.mops/issues/19). CoNVaDING and ExomeDepth were designed to handle noise and more difficult data, but both rely on the user to discard low-quality samples or to isolate reference samples which have low noise to be used as models to fit their models on. clearCNV works without such preparatory steps by clustering the data beforehand and then filtering out low quality samples.

clearCNV also allows the user to visualize the results of the clustering and filtering steps to validate the parameter choices and allow to adjust parameters for fine-tuning when necessary.

Adding preprocessing steps, such as clustering, and batch separation allowed clearCNV to compensate for greater structural difficulties observable in the data. Other tools take similar but not as far-reaching approaches by finding subsets of samples that form a common statistical background for any single sample. clearCNV does that in addition to the two previous steps of panel-re-assignment and batch separation, which are also embedded in a user-friendly interactive *Dash* interface.

Acknowledgements

The Authors thank Dr. January Weiner for the helpful discussion.

Funding

The authors have no funding to declare beyond their organisation.

Conflict of Interest: none declared.

References

Benjamini,Y. and Speed,T.P. (2012) Summarizing and correcting the GC content bias in high-throughput sequencing. Nucleic Acids Res., 40, 1–14.

Cao,M.D. et al. (2018) Simulating the dynamics of targeted capture sequencing with CapSim. Bioinformatics, 34, 873–874.

Chiang,T. et al. (2019) Atlas-CNV: a validated approach to call single-exon CNVs in the eMERGESeq gene panel. Genet. Med., **0**, 1–10.

Collins,R.L. et al. (2020) A structural variation reference for medical and population genetics. Nature, 581, 444–451.

Conrad,D.F. et al. (2010) Origins and functional impact of copy number variation in the human genome. Nature, **464**, 704–712.

Daniel et al. (2011) Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. Genome Biol., 12.

Johansson,L.F. et al. (2016) CoNVaDING: Single Exon Variation Detection in Targeted NGS Data. Hum. Mutat., **37**, 457–464.

Marshall,C.R. et al. (2020) The Medical Genome Initiative: moving whole-genome sequencing for rare disease diagnosis to the clinic. Genome Med., 12, 48.

Moreno-Cabrera,J.M. et al. (2020) Evaluation of CNV detection tools for NGS panel data in genetic diagnostics. Eur. J. Hum. Genet., 1645–1655.

Ng,S.B. et al. (2009) Targeted capture and massively parallel sequencing of 12 human exomes. Nature, **461**, 272–276.

Nowakowska,B. (2017) Clinical interpretation of copy number variants in the human genome. J. Appl. Genet., **58**, 449–457.

Ott,C.E. et al. (2010) Deletions of the RUNX2 gene are present in about 10% of individuals with cleidocranial dysplasia. Hum. Mutat., **31**, E1587–E1593.

Povysil,G. et al. (2017) panelcn.MOPS: Copy-number detection in targeted NGS panel data for clinical diagnostics. Hum. Mutat., **38**, 889–897.

Sadedin,S.P. et al. (2018) Ximmer: A system for improving accuracy and consistency of CNV calling from exome data. Gigascience, 7, 1–11.

Sismani,C. et al. (2015) Copy number variation in human health, disease and evolution. Genomic Elem. Heal. Dis. Evol. Junk DNA, 129–154.

Sudmant,P.H. et al. (2015) An integrated map of structural variation in 2,504 human genomes. Nature, 526, 75–81.

Zarrei,M. et al. (2015) A copy number variation map of the human genome. Nat. Rev. Genet., 16, 172–183.

Zhang,F. et al. (2009) Copy number variation in human health, disease, and evolution. Annu. Rev. Genomics Hum. Genet., **10**, 451–481.

Supplemental Material for

ClearCNV: CNV calling from NGS panel data in the presence of ambiguity and noise

Vinzenz May[1], Leonard Koch[2], Björn Fischer-Zirnsak[2,3], Denise Horn[2], Petra Gehle[4,5], Uwe Kornak[3,6], Dieter Beule[1] and Manuel Holtgrewe[1,*]

[1] Berlin Institute of Health at Charité – Universitätsmedizin Berlin, Core Unit Bioinformatics (CUBI), Charitéplatz 1, 10117 Berlin, Germany

[2] Charité – Universitätsmedizin Berlin, corporate member of Freie Universität Berlin and Humboldt-Universität zu Berlin, Institute of Medical Genetics and Human Genetics, Augustenburger Platz 1, 13353 Berlin, Germany

[3] Max-Planck-Institut für Molekulare Genetik, FG Development & Disease, 14195 Berlin, Germany

[4] Charité-Universitätsmedizin Berlin, corporate member of Freie Universität Berlin and Humboldt-Universität zu Berlin, Department of Internal Medicine - Cardiology, Berlin, Germany

[5] DZHK (German Center for Cardiovascular Research), partner site Berlin, Germany

[6] Institute of Human Genetics, University Medical Center Göttingen, Göttingen, Germany.

*To whom correspondence should be addressed.

S1 Data overview

Table S1 shows the core characteristics of the panels used in our evaluation.

Panel ID	Panel Version	Disease Field	# Genes	# Exons and targets	Mbp target	# Cases input	# Cases resolved
A (BM)	1	Bone Mass Disorders	76	833	0.21	172	164
B (CBM)	2	Bone Mass Disorders	383	5866	1.60	95	94
C (CBM2)	3	Bone Mass Disorders	384	6194	1.15	270	281
D (SDAG1)	1	Skeletal Disorders	407	6538	1.22	117	104
E (SDAG2)	2	Skeletal Disorders	408	6233	1.71	345	340
F (TAAD)	1	Connective Tissues Disorder	37	972	0.13	252	252
G (TAAD2)	2	Connective Tissues Disorder	89	1953	0.52	156	156

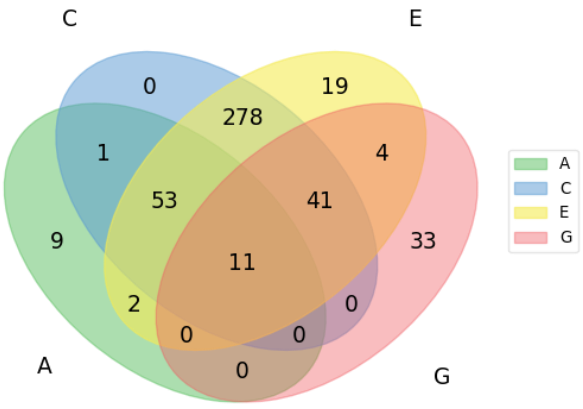

Fig. S1. Venn diagram of panels A,C,E, and G. Panels A,C,E, (and B,D) all share a relatively great set of common genes, whereas G (and F) have relatively many exclusive genes. Eleven genes are part of each panel.

S2 WGS read pair gap size based confirmation.

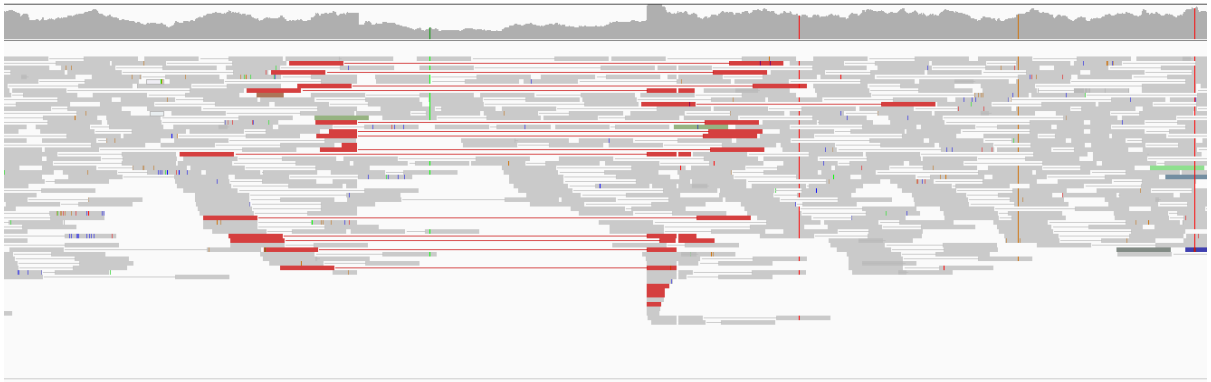

Fig. S2. IGV screenshot of heterozygous deletion found in WGS data. The top track shows the coverage per base. The aligned paired end reads are indicated by grey bars with thin strokes connecting them. Read pairs spanning a much larger region than a usual fragment does are marked in red color. The deletion can be seen in the coverage track in the form of a sudden interruption and a short area with a lower coverage, as well as it can be seen by the placement of the split reads.

We had a small subset of patients sequenced both as a WGS (whole genome sequencing) and a targeted sequencing panel sample. One CNV was found in both samples. This CNV could be validated via visual inspection in IGV. The corresponding screenshot is found in Fig. S2.

S3 Sample to panel re-assignment and batch separation

The following steps were done with clearCNV in its interactive *plotly Dash* environment. The graphics are screenshots taken from the responsive plots and all parameters are reported separately. The shown Figures represent the actual data (1407 samples in seven sequencing panels) and solutions we worked with. The final output was then used as the input of clearCNV's CNV calling pipeline.

I. The first step is the PCA transformation of the data to 20 dimensions. Depicted are the first two principal components. The underlying data matrix is a binary matrix (see paper section 2.4.). The colors indicate the underlying sequencing panel a sample is assigned to. The selected dimensions of the PCA ado not necessarily produce a nice cluster separation in the actual PCA plot, but rather in the following (tSNE) step.

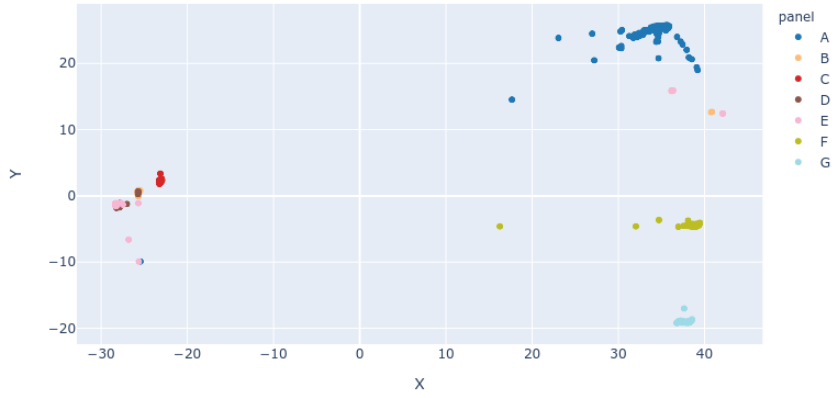

Fig. S3: Scatterplot of PCA transformed data. The x-axis shows the first, the y-axis the second principal component of the PCA transformed data (samples) and coloring according to the per sample assigned panel. Note that there is no good separation visible.

II. The data is transformed by a t-distributed stochastic neighbor embedding (tSNE). This brings the dimensions down from 20 to two and creates a much better separation. Again, the colors indicate the underlying sequencing panel a sample is assigned to.

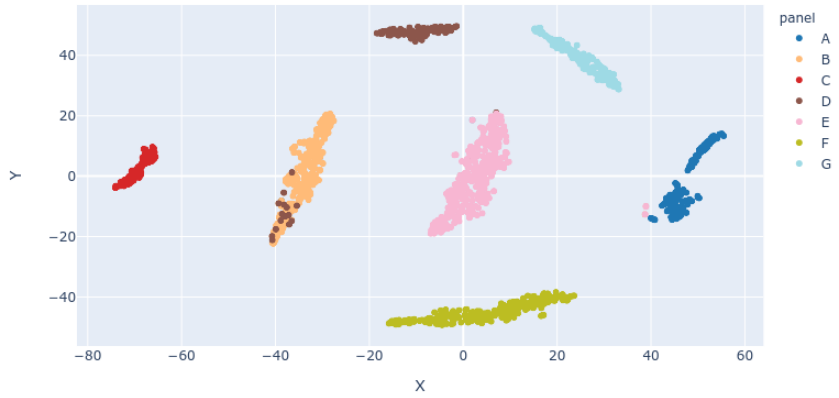

Fig. S4: Scatterplot of tSNE transformed data. The x-axis shows the first, the y-axis the second dimension of the transformed data. Again, the coloring is according to the per sample assigned panel. Note how some samples from panels D and E appear clearly displaced.

III. The two-dimensional data is clustered, and the samples are re-assigned.

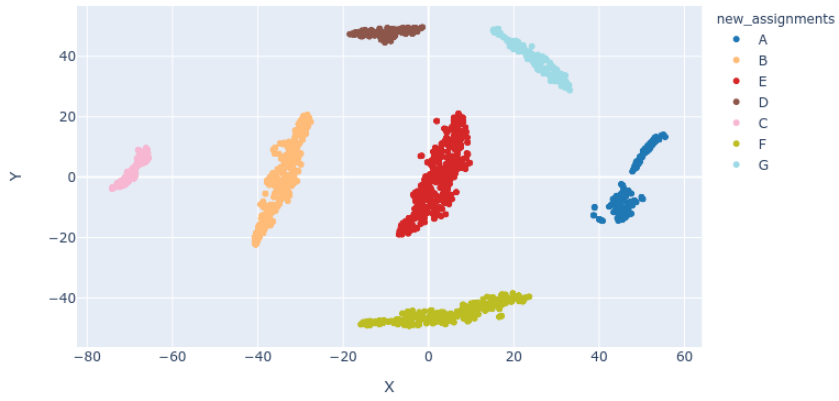

Fig. S5: Scatterplot of tSNE transformed data with re-assignment of samples to panels. Agglomerative clustering finds the clusters. The number is defined by input sequencing panels. To assign a panel to a cluster, a majority vote is held. This procedure assumes that only a minority of samples is wrongly assigned at the start of the analysis.

IV. A clustermap is plotted for the original assignment of the samples. Samples and targets (or exons) are both subject to clustering. The aim is to show the user the difference of the panels and at the same time to control the threshold, when a target is considered covered or uncovered or find the right discimination between low coverage an off-target read alignments. The threshold can be adjusted in the plotly Dash UI.

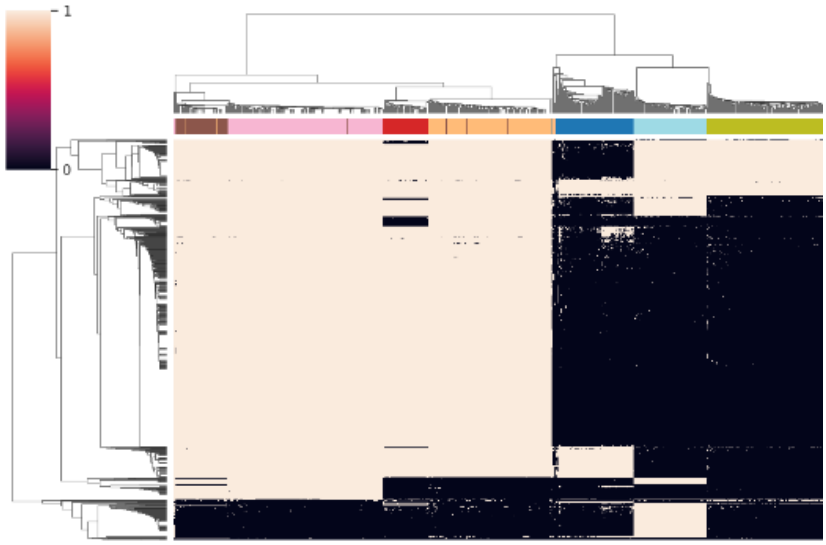

Fig. S6: Clustered heatmap of all samples on all targets from all sequencing panels. Each column represents a sample and each row a target (or exon). The horizontal colour-bar on top of the heatmap indicates the originally assigned sequencing panels each with one different colour - seven in this case. Displaced strokes indicate wrongly assigned samples. The heatmap shows if a target of a sample is considered covered (white or beige) or uncovered (black).

V. The new assignment is shown again based on the clustered heatmap. A user can compare the new assignment with the clustering of the targets and samples.

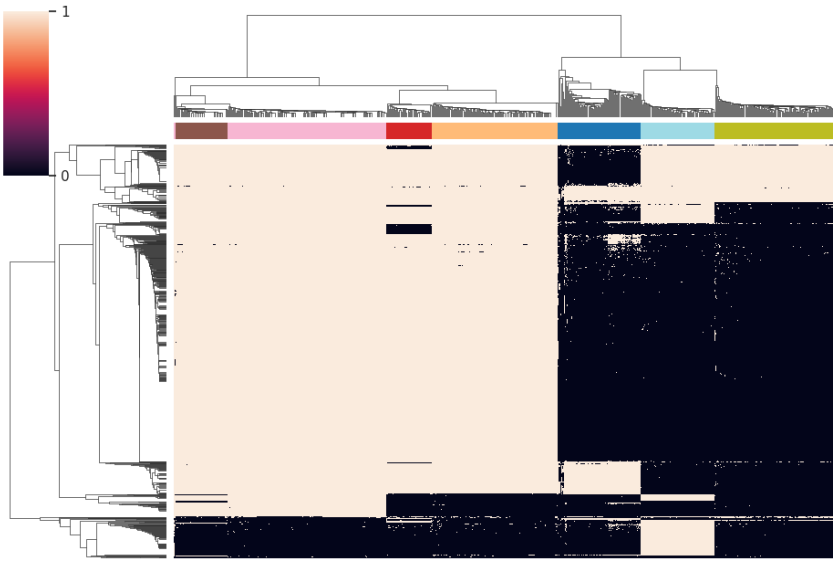

Fig. S7: Clustered heatmap with new sample-panel assignment. Same as Fig. S6. except for the color bar now showing the new sample to panel assignment.

VI. Batches are separated. The user defines the number of clusters per given panel. The clusters are found in a Gaussian mixture clustering. The final assignments are then printed to text files, allowing the user to use the new assignments directly in downstream CNV calling.

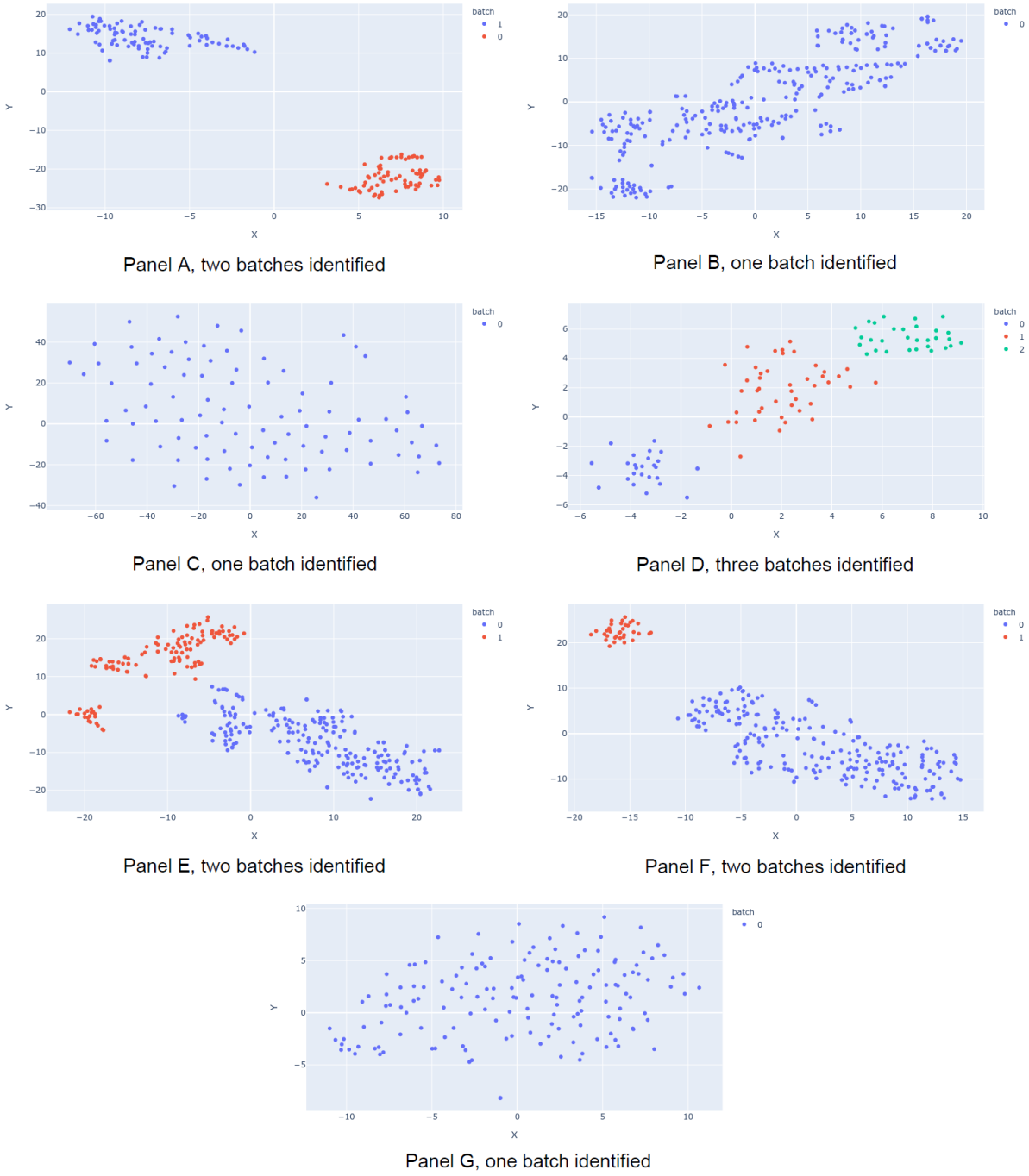

Fig. S8: Scatter plots of all batch separations. The seven Subfigures show how all batches within the seven panel-associated clusters of samples were resolved. The scatterplots are obtained by transforming the normalized coverage data via PCA and tSNE. For each panel, target fragment counts were obtained based on the provided BED-file. The underlying panel is found in each Subfigure's description. The identified batches are indicated by color and description.

S4 Ambiguous qPCR results and GC content analysis

Nine out of 88 CNV calls showed ambiguous results in the qPCR validation. After trying several different DNA melting temperatures, we analyzed GC-content, mappability and size of the CNV calls. Fig. S9 shows a boxplot of the GC content of all CNV calls for each qPCR outcome. For (1) GC-content, (2) mappability and (3) size we tested the differences as follows: (1): one-sided t-test at alpha level of 0.05, (2) one-sided t-test at alpha level oof 0.025 and (3) in a two-sided t-test at alpha level of 0.025.

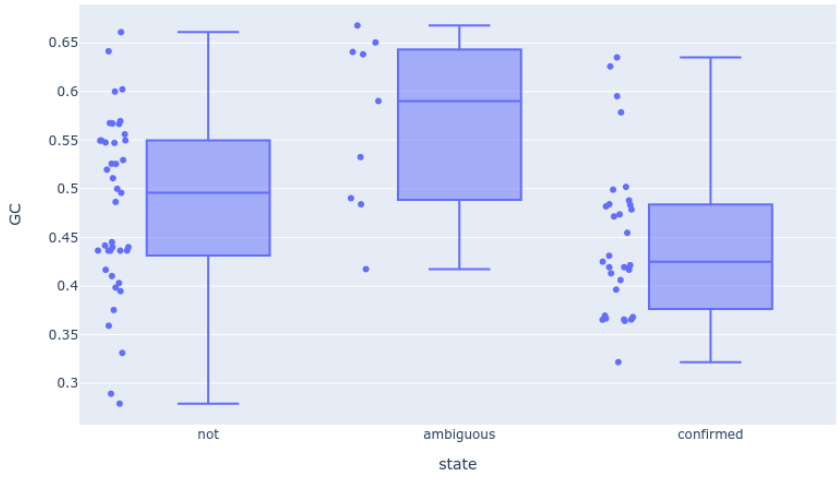

Fig. S9. Boxplot of the GC-content of all confirmed, unconfirmed, and ambiguous results of the qPCR validation. Each box shows the min and max, the 0.1 and 0.9 quantiles and the median. Each box has the corresponding GC values left of it as scatter plots. The unconfirmed CNV calls (*not*) joined with the confirmed CNV calls (*confirmed*) appear to have a lower GC-content than the *ambiguous* ones. Note that CNV calls from all three classes can have a GC-content higher than 0.6.

The GC-content of all ambiguous CNV calls was significantly higher (alpha=0.05) than the GC-content of unconfirmed and confirmed CNV calls joined together. We tested this in a one-sided t-test resulting in nosignificant difference at p=0.0199. The presence of ambiguous validations on targets with a GC-content of 0.55 to 0.4 shows that GC-content plays a role. However, they do not fully explain the reasons for success or failure of validation of CNV calls with qPCR. Therefore, we plotted another boxplot for the mappability (36-mers on the hg19/GRCh37 release) for each qPCR outcome in Fig.S10. We tested for difference using a one-sided t-test at alpha level of 0.025. We expected ambiguous qPCR outcomes to occur with lower mappability targets.

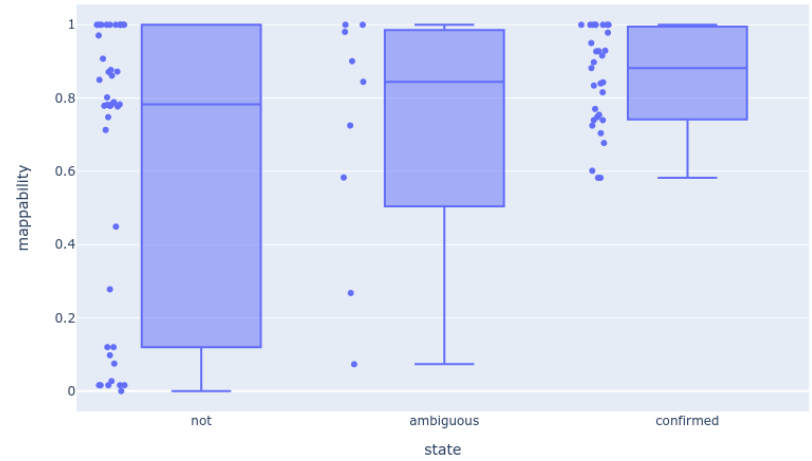

Fig. S10. Boxplot of the mappability of all confirmed, unconfirmed, and ambiguous results of the qPCR validation. The box and scatter plots show the same metrics as in Fig. S9. The unconfirmed CNV calls (‘*not’*) show regions with both low and high mappability. The confirmed CNV calls (*confirmed*) contain only targets with a mappability of 0.58 or greater. The *ambiguous* CNV calls have similar mappability values to the unconfirmed ones.

The one-sided t-test indicates that ambiguous qPCR results don’t show a significantly lower mappability than unconfirmed (‘not’) and confirmed (‘confirmed’) qPCR results at p = 0.64.

As it can be seen on Fig.S11, CNV calls can have both a moderate GC-content and a high mappability. Therefore, there must be another factor that determines if a CNV call can be confirmed via qPCR.

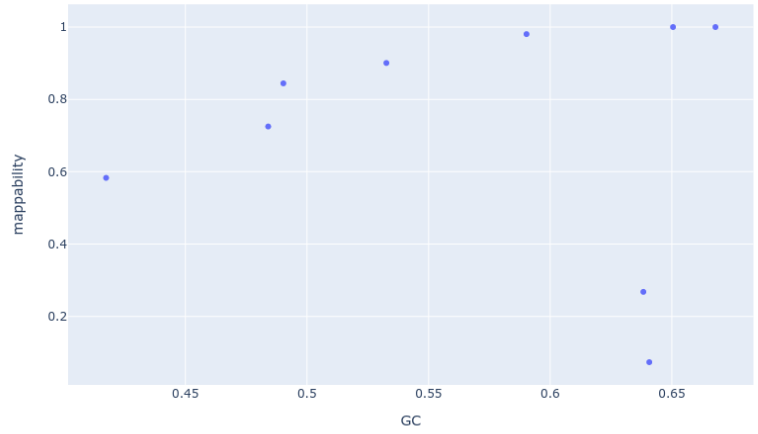

Fig. S11. Scatterplot of the ambiguous qPCR results with GC-content vs. mappability. Each dot indicates the GC-content and the mappability of the genomic region corresponding to a CNV call that showed an ambiguous result in the qPCR validation. Note that CNV calls can have both a moderate GC-content and a high mappability.

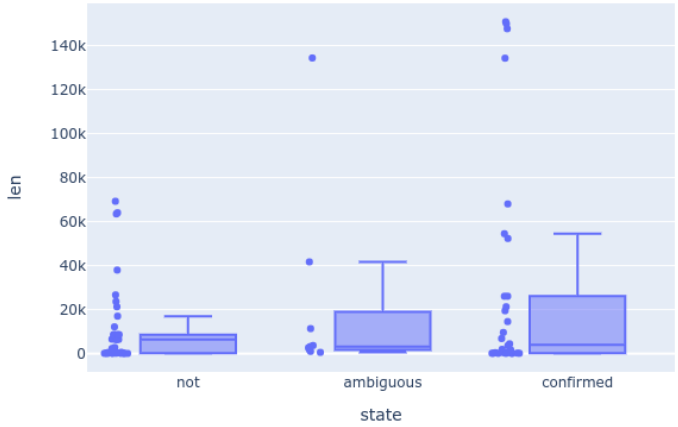

Fig. S12. Boxplot of the size in bp of all confirmed, unconfirmed, and ambiguous results of the qPCR validation. Each box shows the min and max, the 0.1 and 0.9 quantiles and the median. The unconfirmed CNV calls (*'not'*) show regions with both low and high mappability. The confirmed CNV calls (*confirmed*) contain only targets with a mappability of 0.58 or greater. The *ambiguous* CNV calls seem to be composed of mainly smaller sized regions but are not significantly smaller than the other (*'not'* and *'confirmed'*) CNV calls. We performed a student's t-test

We tested if ambiguous qPCR results are smaller in size (bp) than all others. The one-sided t-test showed no significant difference at p=0.6383.

To summarize, we were not able to find a single feature or a combination of features indicating if a qPCR result was ambiguous.

S5 False negatives in clearCNV's calls

In this section, we will investigate the false positive calls in the clearCNV results.

Three true CNVs were not called by clearCNV. All three are deletions and two of them belong to the same sample. This sample was found in the data set corresponding to panel A, which was composed of two major batches. The failed sample was found in one of the two clusters (see Fig. S8 subfig. For panel A) where it failed to be assigned to a sample group of sufficient size (its group held no other samples). clearCNV is designed to exclude such samples that would be expected to yield many CNV calls, which would achieve a high

score due to extreme noise. By dropping this sample, clearCNV excluded many CNV candidates that would have been high scoring false positives. These false positives would have overshadowed other high scoring CNV calls which are more likely to be true positives.

The third missed CNV call is a small deletion spanning two exons. Its region is noisy, resulting in a low scaled z-score on which the Gaussian HMM then did not make the call. Even though this might seem like clearCNV is biased against small CNVs, Fig. S13 shows that this is not the case. Small CNV calls are well present among all CNV calls of clear CNV as well as all confirmed CNV calls.

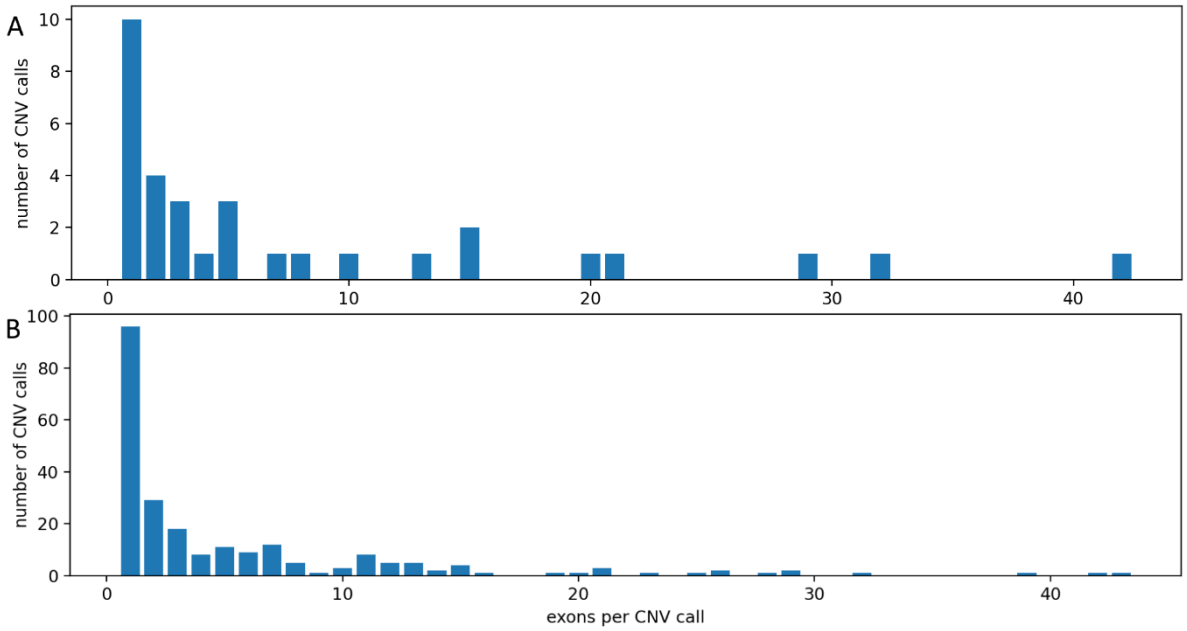

Fig. S13. Distribution of exons per CNV call in clearCNV's results. Sub Figure A shows the CNVs that were called by clearCNV and that could be confirmed by qPCR. Subfigure B shows all CNV calls made by clearCNV. Note how clearCNV called a lot of single-exon and other small CNVs.

S6 Performance evaluation on simulated targeted sequencing data

To complement our real-world data, we simulated targeted sequencing data. We could thus evaluate clearCNV, ExomeDepth, CoNVaDING and panelCN.MOPS with a known ground truth. This allows us to directly calculate a positive predictive value and a false discovery rate for each tool on any type of experiment.

As we also show, the downside of this approach is that simulated data is not able to reproduce all variances and biases seen in real data.

Simulation of reads

An experiment E is composed of a set of samples E = $\{R_1,R_2,\ldots,R_n\}$ containing simulated targeted NGS reads. For each sample R, simulated reads on defined regions (the panel) were generated on four identical haplotypes S = (S_1,S_2,S_3,S_4). Any combination of CNVs in a final sample R can be achieved by sampling reads from S, which means $R \subseteq S$. Given a sample R, for any given locus *l*, four different types of CNVs can be simulated as follows:

- Homozygous deletion: no reads in S_1,S_2,S_3,S_4 that align to *l* are found in R.
- Heterozygous deletion: only reads in S_1 that align to *l* are found in R.
- Homozygous duplication: only reads in S_1,S_2,S_3 that align to *l* are found in R.
- Heterozygous duplication: all reads in S that align to *l* are found in R.
- Wild type: only reads from S_1,S_2 are found in R

A wild type sample R_w is simply $R_w=\{S_{w1},S_{w2}\}$. The number of fragments N_R is sampled for each sample R from the empirical distribution of the **D** (SDAG1) panel, which holds 104 samples after panel reassignment by clearCNV. After adding CNVs, the number of fragments changes according to the copy numbers and sizes of the CNVs. This panel comprises 407 genes in 6538 exons (and targets) on 1.22mb of targeted sequence.

To rephrase, we first simulate reads of targeted sequencing for each individual for four alleles of a sample. By picking two alleles at a locus, we simulate a wild-type locus. By picking one allele, we simulate a heterozygous deletion, by picking three alleles, we simulate a heterozygous duplication, and so on. This approach offers much greater simplicity over spiking CNVs into the reference and then simulating from a modified reference.

We designed twelve experiments to investigate different features of each tool in different settings. Tab. S2 shows an overview of all experiments.

The Python3-script to generate the simulated reads can be found in the according repository (https://github.com/bihealth/clear-cnv-supplementary).

Tab. S2. Experiment designs.

Experiment name	Characteristics	Total samples	Sample composition	Sample structure
Wild types	Absence of true signal	60	60 wt	no CNVs
Variants het	Representative test	120	60 wt + 60	random DUP and random DEL
CNV rich	High prevalence of CNVs	81	60 wt + 3*7	3 samples with 6,10,16,24,34,46,60 CNVs each
Low coverage	Performance on different levels of low coverage	120	120	Three samples with $20{,}000 * i, i \in \{1, \dots, 20\}$ fragments (coverage). One deletion and one duplication of any size per sample.
Dels het small	Performance on single exon events	80	60 wt +20	single heterozygous deletion, single exon
Dels het medium	Performance on medium events	80	60 wt +20	single heterozygous deletion, 5 to 10 exons
Dels het big	Performance on large events	80	60 wt +20	single heterozygous deletion, 15+ exons
Dels hom	Performance on homozygous deletions	80	60 wt +20	single homozygous deletion, any size
Dups het small	Performance on single exon events	80	60 wt +20	single heterozygous duplication, single exon
Dups het medium	Performance on medium events	80	60 wt +20	single heterozygous duplication, 5 to 10 exons
Dups het big	Performance on large events	80	60 wt +20	single heterozygous duplication, 15+ exons
Dups hom	Performance on homozygous duplications	80	60 wt +20	single homozygous duplication any size

Tab. S1. Provides an overview of all 12 experiments. We wanted to investigate different setups, where the composition, sizes or numbers of CNVs or the coverage are distributed not as closely as possible to the data we have seen. This may the reader to decide which tool is best for given data with certain properties. The columns can be read as follows. **Experiment name**: The name of the experiment as it is shown in Fig. S14. **Characteristics**: key features of the experiment. **Total samples:** the total number of samples used in the experiment. **Sample composition**: for most experiments, we chose to use 60 background samples ("wt") with no CNVs and an additional set of samples (of size 20 in most cases) that carry CNV signals. The low coverage experiment is composed of only signal carrying samples. **Sample structure** summarizes the samples' features. The **CNV rich** experiment is made from seven different configurations, which are implemented in three samples each. The first three samples carry each three deletions and three duplications. The next three samples carry five deletions and five duplications and so on, until 30 deletions and 30 duplications are implemented in the last three samples. The **low coverage** experiment starts with two samples from which only 20,000 fragments are simulated. For the next two samples 40,000 fragments are simulated. This is repeated for another 18 steps, until the last two samples have a fragment count of 400,000. The **Variants het** experiment is composed of 60 samples without any CNVs (wild type) and 60 samples of which each carries one heterozygous deletion and one heterozygous duplication. The CNVs are picked randomly (i.i.d.) from the gnomAD SV 2.1 sites. All SVs picked are either duplications or deletions and fully cover at least one exon in the target regions of the **D** (SDAG1) panel.

Performance evaluation

In the following, different performance metrics are displayed with which we evaluated the results of the performance assessment experiments. The used metrics are defined as follows:

Abbreviation	Name	Formula
PPV	Positive predictive value	TP / (TP + FP)
FPR	False positive rate	FP / (FP + TN)
Sens	Sensitivity	TP / (TP + FN)
FP	False positives	
TP	True positives	
FN	False negatives	
TN	True negatives	

In Fig. S14. The different experiments highlight certain strengths and weaknesses of all four CNV calling tools. Please note that the true negative CNV calls on all targets are very high numbers and to make the table in Fig. S14 more concise, we decided to display them in a separate table Tab. S3.

clearCNV shows to have the highest PPV of all four tools in all experiments except *variants het*. In the *CNV rich, low coverage*, and the *del/dup het small* experiments, the PPV is highest at the cost of the sensitivity. *clearCNV* is very conservative on calling CNVs if the data seems to have a low quality or untrustworthy properties. On the other hand, the more balanced the data appears, the more *clearCNV* emphasizes a high sensitivity. The reason for such behavior is found in the use case we developed *clearCNV* for. We had only limited resources on the qPCR validation of CNV calls and aimed for a maximized positive yield.

ExomeDepth shows a high overall performance with no outstanding biases. It appears that *CoNVaDING* called many false positives, which are not evenly distributed but concentrate on only a single or a few samples as it can be seen in Figure S15. The case for panelCN.MOPS is similar. We were not able to find the root cause of this behavior.

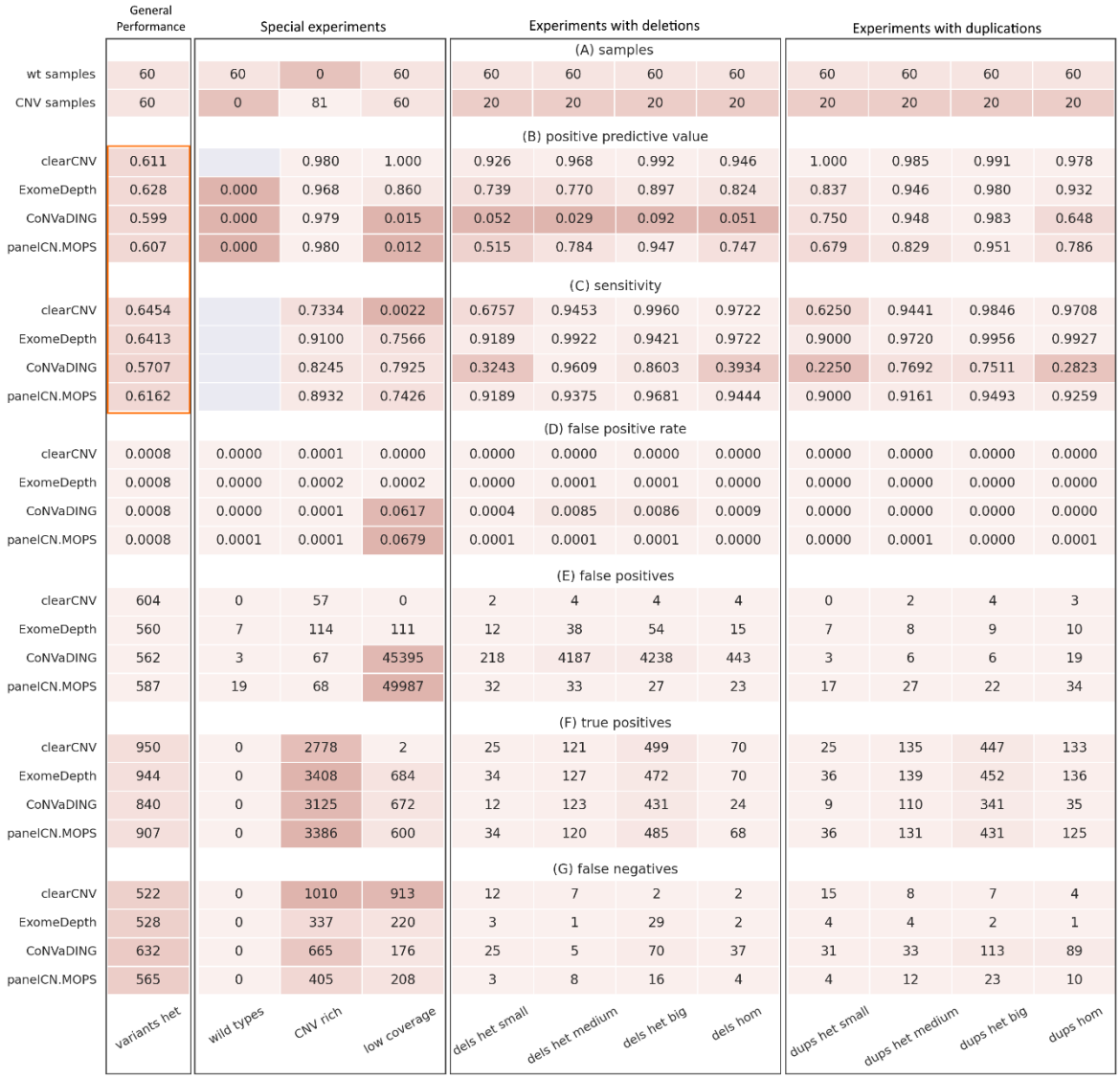

Fig. S14. Test results of all experiments using simulated reads. This is a summary of all evaluation results of the four selected CNV calling methods on twelve different experiments. The two top rows show the number of copy number neutral and copy number variable samples. Each four rows below that present the results of an indicated testing metric (rows B, C, D) or simply the counts (rows E, F, G). Each column corresponds to an experiment. The first column shows the results of the *variants het* experiment which carries several different CNVs. This experiment has no special focus and thus is best suited for a quick performance assessment. The most relevant results to quickly assess the overall performance are marked with an orange box. The next three columns show the results of experiments with extreme parameters. The *Wild types* experiment has no CNVs, *CNV rich* has a very high number of CNVs and no copy number neutral samples. *Low coverage* has samples with low coverage (40k to 400k fragments per sample). The next two blocks contain all experiments associated with only deletions and with only duplications. All TP, FP, TN (true negatives), and FN numbers are single exon counts. We chose to reduce the evaluation to single exon-level because not every tool merges adjacent CNV calls or called exons with a certain distance into one single CNV call. Fields showing no value and only a grey color could not be calculated due to division by zero. In the case of *clearCNV* in *PPV* this is the case because there were no false positive CNV calls. The *wild type* experiment has by design neither FNs nor TPs. Thus, the *sensitivity* could not be calculated.

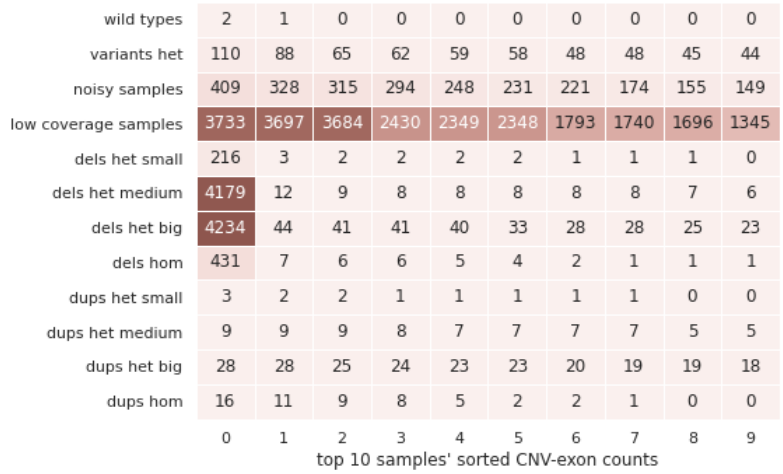

Fig. S15. Head of sorted called exons per sample for CoNVaDING. Each row shows the top ten sorted called exons per sample for the results of CoNVaDING. It allows to identify that in the experiments *dels het small*, *dels het medium*, *dels het big*, and *dels hom* there is each time one sample on which the great majority of CNV-exons was called.

Tab. S3. True Negative (TN) CNV called targets per tool and experiment.

	clearCNV	ExomeDepth	CoNVaDING	panelCN.MOPS
variants_het	734,844	734,888	734,886	734,861
wild_types	368,460	368,453	368,457	368,441
noisy_samples	493,576	493,562	493,564	493,562
low_coverage_samples	736,005	735,905	690,677	686,125
dels_het_small	491,241	491,231	491,025	491,211
dels_het_medium	491,148	491,114	486,965	491,119
dels_het_big	490,775	490,725	486,541	490,752
dels_hom	491,204	491,193	490,776	491,185
dups_het_small	491,240	491,233	491,237	491,223
dups_het_medium	491,135	491,129	491,131	491,110
dups_het_big	490,822	490,817	490,820	490,804
dups_hom	491,140	491,133	491,137	491,111

S7 Simulated data does not capture the variances of real-world data

A key issue to look out for when using simulated data is how well they match with real-world data. In this section, we make an in-depth analysis of the characteristics of the simulated data that we used and compare it to the characteristics that we observe in the real-world data. Such differences in data can explain different behavior on simulated vs. real-world data.

There are very few software packages available that allow for the simulation of targeted sequencing data. We chose CapSim as the most promising one.

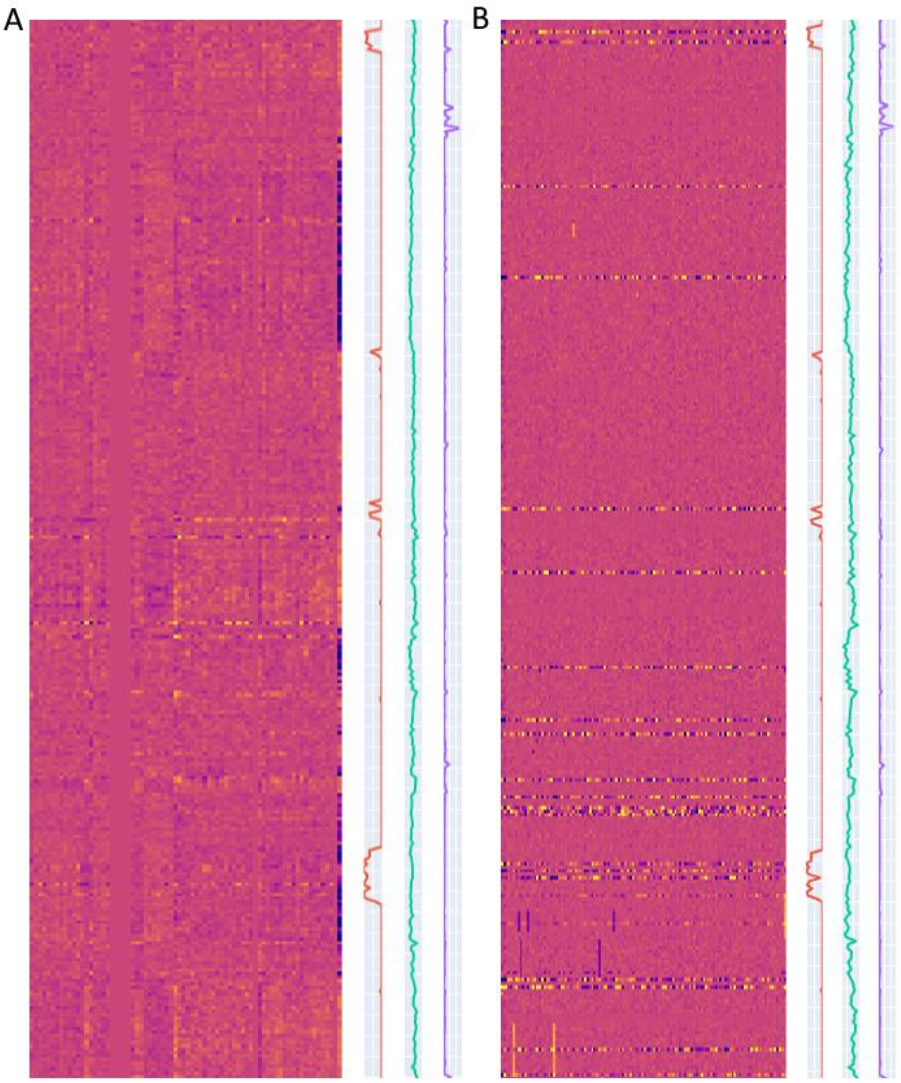

Fig. S16. Heatmap-like visualization of normalized coverage per exon. Both subfigures show a subset of the exons found in Panel D on all samples. Both plots are accompanied by three tracks in orange, green and purple, where the orange track indicates the 36-mer mappability, green indicates the GC-content and purple indicates the log size of the exon. Mappability and GC content range from 0 (left) to 1 (right). Purple (log size) ranges from 1 (left) to the local maximum. In the heatmap, a row corresponds to the ratio scores. Ratio scores are normalized fragment counts per target. A column corresponds to a sample. Vertical dark lines indicate deletions while vertical bright lines indicate duplication events. Horizontal lines of mixed colors indicate exons with a high variance. The two datasets show different characteristics. While the simulated data shows a homogeneous variance of normalized fragment counts in all targets except for the high variance targets and the CNV events. The base variance of the real-world data shows some patches of slightly elevated or lowered ratio scores.

The two subplots of Figure S16 show only a subset of targets on all available samples. The visualization of ratio scores allows to quickly spot CNV candidates and regions of high variance. It also allows to get an impression of the character of the base variance. The base variance of the simulated data (Subfigure B) seems to be rather homogeneous. The real-world data (Subfigure A) shows regions of slightly elevated or lowered coverage. These regions imply that the noise (or the variance) is not independently random.

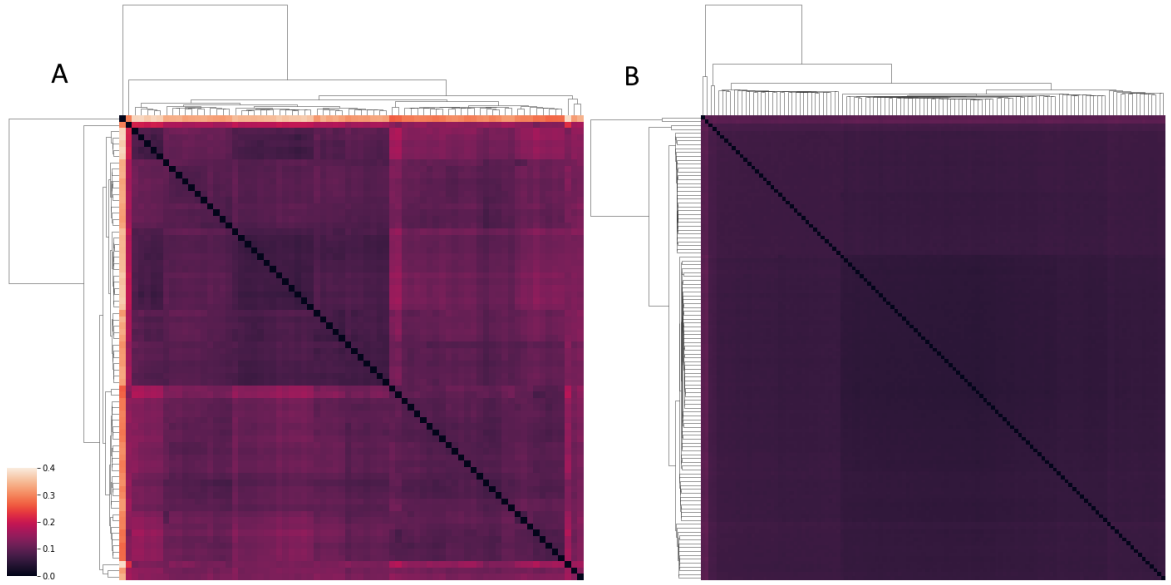

Fig S17. Clustered heatmaps of sample distances in real and simulated data. Both Subfigures show clustered heatmaps of the match scores (distance metric) calculated and visualized by clearCNV. A dark color indicates a smaller distance, a brighter color a greater distance. Both matrices are symmetric. Subfigure A shows the real-world data of panel D (SDAG1) of the larger batch. Subfigure B shows the simulated data of the experiment *variants het*. It can be seen in the real-world data that there are smaller groups of samples that share a similar coverage profile than other samples. Subfigure B shows a rather homogeneous situation with no smaller subclusters visible and a maximum match score of about 0.1, which is about half as much as in the real-world data.

clearCNV builds groups of samples that have similar profiles of this kind. Figure S17 shows a clustered heat map of distances of samples (match scores) of real-word data (Subfigure A) and simulated data (Subfigure B).

So far we have described qualitative and visual observations. We attempted to derive a quantitative measure for this. We calculated the linear correlation (Pearson correlation) between row **x** and row **x+i** in the heatmaps. The plots in Figure S18 show the difference of simulated and real-world data at different distances **i**.

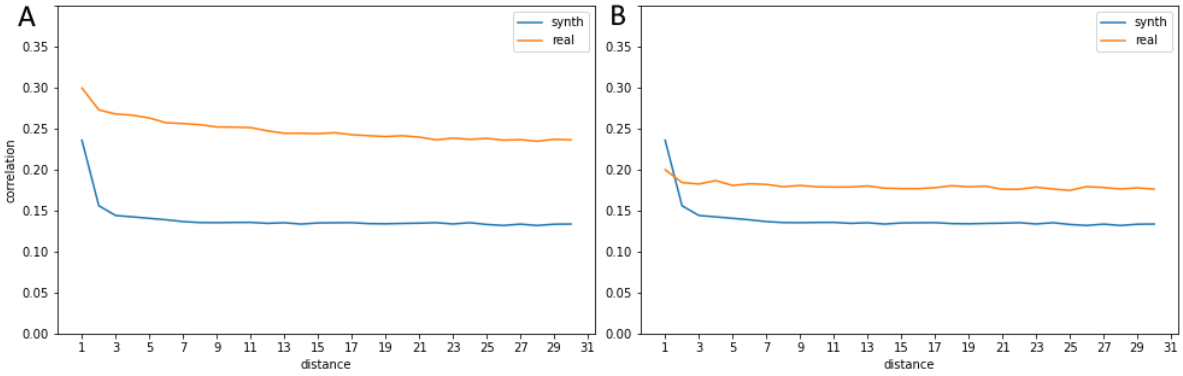

Fig. S18. Cross-correlation plots. Both subplots show the pairwise linear cross-correlation of normalized fragment counts (ratio scores) in all samples. All fragment counts per exon per sample can be represented in a matrix, where a row corresponds to an exon and a column corresponds to a sample. To approximate any correlation, we chose to calculate the absolute Pearson-correlation between a row_x and row_{x+i}. *i* corresponds to the *distance* axis of both plots. clearCNV separated the real-world data set into two batches to analyze them independently so we are presented two plots. The distance *i* ranges from 1 to 30. It is visible that the absolute linear correlation is much higher in the real-world data than it is in the simulated data. Subplot A represent the larger batch of 73 samples. Subplot B represents the smaller batch of 30 samples. The real-world data is shown in orange (*real*), the simulated data in blue (*synth*)

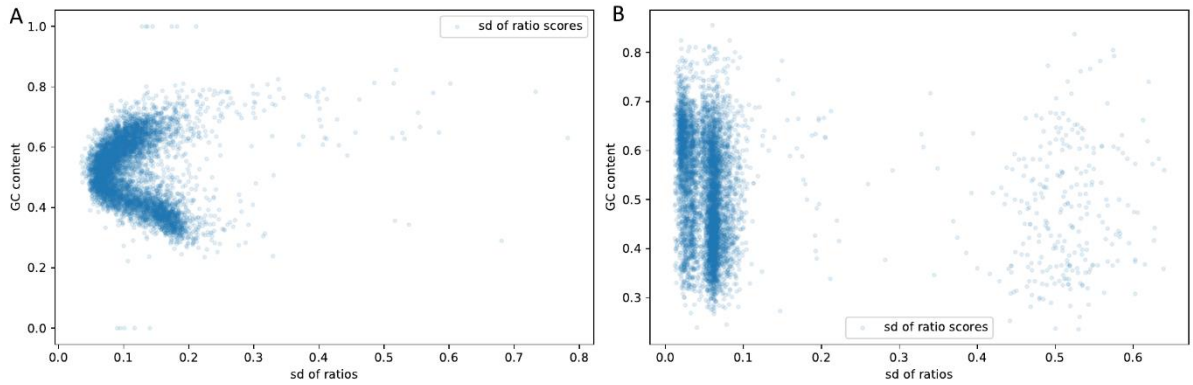

Fig. S19. Standard deviation of normalized coverage vs. GC content. Both subfigures show a scatterplot, where each dot is defined by an exon's standard deviation of ratio scores (taken from all samples) and an exon's GC content. Subplot A shows the first batch of the real-world data of Panel D (SDAG1). Subplot B shows the simulated data of the *variants het* experiment. The normalized coverage (ratio scores) of the real-world data has the lowest standard deviation on targets with a GC content around 0.5. The more deviant the GC content, the higher the sd seems to be. On Subplot B, the simulated data generated with CapSim shows a rather uniform distribution which seems independent of the GC-content. Furthermore, a group of outliers is present which is visually different than in Subplot A.

Another strong difference can be seen in Figure S19 which displays exon coverage standard deviation vs GC content of the sequence. It should be expected that the standard deviation (or variance) of the normalized coverage on an exon over all samples would increase with rather low or high GC-contents. The simulated data show no such characteristic.

S8 Overview of the selected tools' features

Tab. S2 summarizes the considered tools' features. The tools use quite different statistical models and algorithmic features to detect CNVs. Our approach uses on one hand a normal distribution assumption of the exponentially transformed z-scores, on the other hand incorporates many different steps to adjust to the data and to limit the influx of any distributional assumptions. This way, we try to properly capture the empirically observed biases and batch effects seen in the real-world data sets. clearCNV can cope relatively well with the wide variety of panel types, panel versions and vendor technologies present in typical heterogenous panel data collections found in rare disease research.

Tab. S2. Name, model, and features of the selected CNV detection tools.

Tool	model	features
ExomeDepth	Logistic regression; beta-binomial fit; hidden Markov model	Construction of reference exome set; fitting of beta-binomial distribution of fragment counts per exon
CoNVaDING	normal distribution, z-scores joint distribution cut-off	CNV calling on groups of selected samples; sample groups share a similar coverage pattern
panelCN.MOPS	Mixture of Poisson; expectation maximization algorithm	Model adjustment to fit to each sample's coverage pattern and noise in each region of interest
clearCNV	Normal distribution of exponentially transformed z-scores; hidden Markov model	Prior batch separation; sample groups share a similar coverage pattern

S9 Descriptive statistics on the CNV calls that were not taken to qPCR validation

Only a small fraction of all CNV calls generated on the real-world data was validated via qPCR. We limited this selection to the CNV calls that at were made by at least three of the selected CNV calling tools.

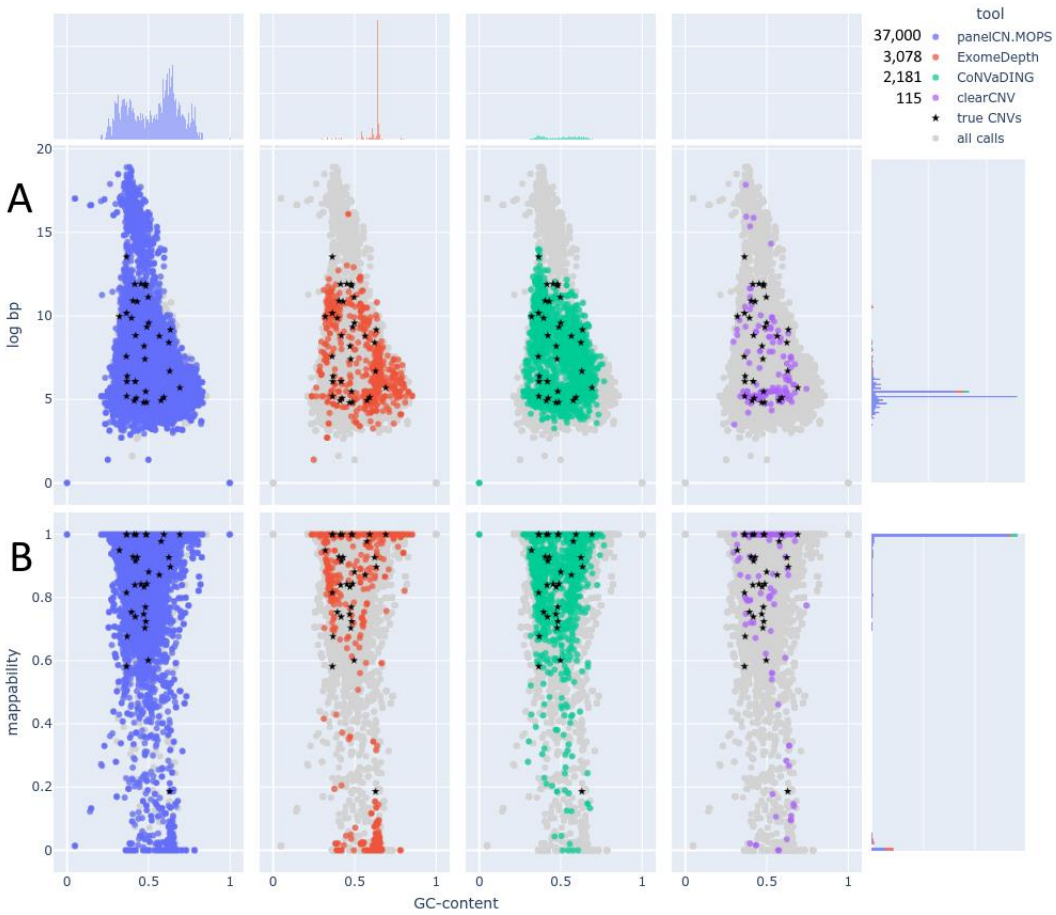

Fig. S20. Scatter plots with marginal distributions of all exclusively called CNVs. Only CNV calls that were called by only one single tool are represented here. Each color represents a tool's results. All tools are plotted as a background in light grey to compare with. CNV calls that were confirmed in the qPCR validation are marked as black stars. The scatter plots in subfigure A show the GC-content vs. the log size in bp. Subfigure A shows the GC-content vs. the mappability for each tool. The marginal distributions are displayed as histograms. The number of CNV calls present here are shown in the legend. 42,374 CNV calls are presented in total. By far the greatest set of CNV calls was generated by panelCN.MOPS (blue) with 37,000 calls. In subfigure A, panelCN.MOPS shows to have generated many very large CNV calls with a GC-content of around 0.4. The other tools' results don't exceed sizes of ~250kb (~12.5 log bp). The marginal distribution of the GC-content for ExomeDepth show a strong peak for CNV calls at around 0.65. The marginal distribution of the size in log bp shows two peaks with sizes of 180 bp or 240 bp (180 bp: 12,048; 240 bp: 8,021). These CNV calls are on single exons. Larger calls are likely multi-exonic. Subfigure B shows that the largest group of CNV calls correspond to targets that map uniquely (mappability of 1.0, generated with 36-mers on Hg19). There is also a small group of CNV calls that correspond to targets which map to multiple regions in the reference (mappability of zero). A great portion of these calls was generated with ExomeDepth (red).

The scatter plots in Figure S20 show that the exclusive CNV calls indicate certain biases on our data of each tool. panelCN.MOPS has generated a great number of very large CNV calls and a great number of total CNV calls. ExomeDepth called many CNVs on targets that have a low mappability, which means that these fragment count data have a high probability to represent different regions in the reference genome than aimed for. The CNV calls of only CoNVaDING show no specific bias, except that there are many. clearCNV is not restricted to high mappability targets and has also generated some very large CNV calls but shows no specific bias.

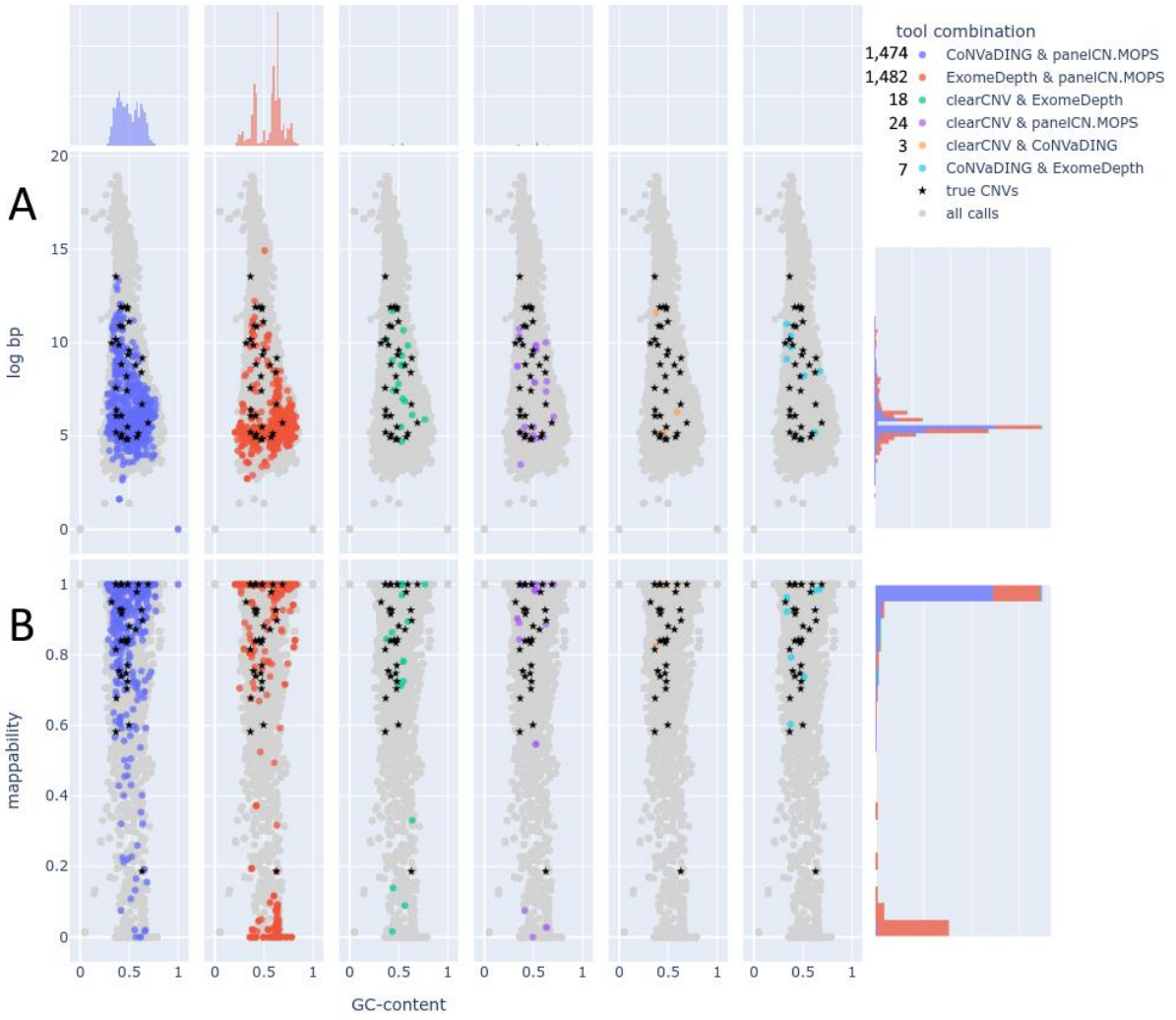

Fig. S21. Scatter plot with marginal distributions of all called CNVs with two tools' support. Only CNV calls that were called by exactly two tools are shown in this Figure. Each combination of two tools is represented by a color. The scatter plots in subfigure A show the GC-content vs. the size in log bp. Subfigure B shows GC-content vs. mappability (36-mers on Hg19). The number of CNV calls present here are shown in the legend. 3,008 calls are presented in total. Most CNV calls presented here are either in the 'CoNVaDING & panelCN.MOPS' (purple-blue) set (1,474 calls) or in the 'ExomeDepth & panelCN.MOPS' (red) set (1,482). Any other tool combination has no great overlap. Almost all called CNVs in low mappability targets are in the 'ExomeDepth & panelCN.MOPS' (red) subset. It can be seen in the marginal distribution of the size in log bp that CNV calls of targets with size 180 or 240 bp are present in very high numbers (180 bp: 501; 240 bp: 855). They are single-exon CNV calls. The larger calls are mostly multi-exonic.

In Figure S21, the overlap of CNV calls that were generated by exactly two tools is shows a much greater portion for the 'CoNVaDING & panelCN.MOPS' and 'ExomeDepth & panelCN.MOPS'. The latter shows again a bias as it does for the exclusive ExomeDepth calls in Figure S20. This bias is that the greater portion of CNV calls are done on targets with a very low mappability. The other tool combinations have only a very few CNV calls ins them, so that it is hard to make any statements about them.

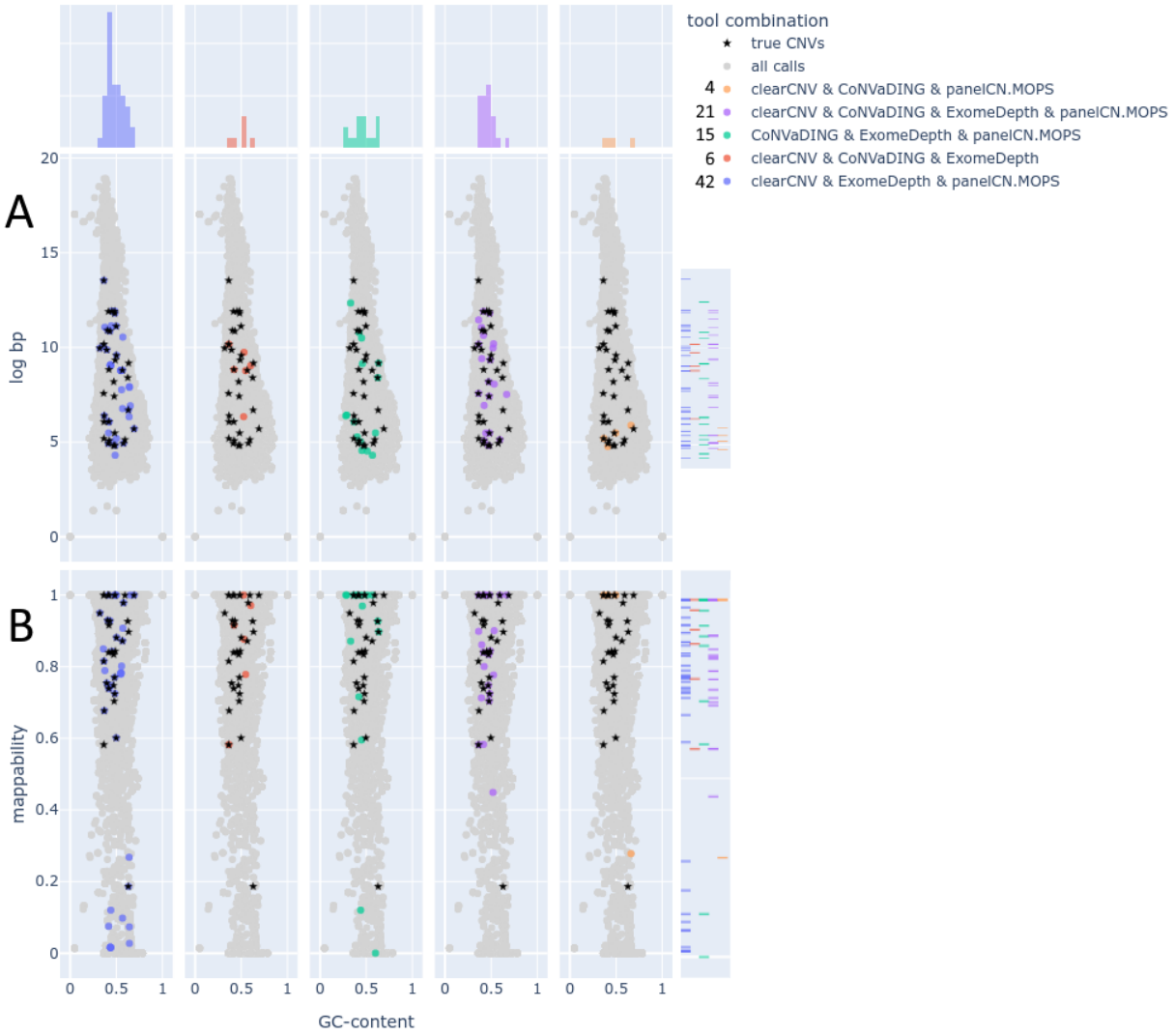

Fig. S22. Scatter plot with marginal distributions of all validated CNV calls of three or four tools' support. Only CNV calls that were called by three or more tools were taken to qPCR-validation. 88 CNV calls were selected this way. The number of CNV calls in each tools combination is found in the legend. Each combination of two tools is represented by a color. The scatter plots in subfigure A show the GC-content vs. the size in log bp. Subfigure B shows GC-content vs. mappability (36-mers on Hg19. The marginal distributions of size in log bp and mappability are rug plots due to low numbers.

Figure S.22 shows that compared to the rest, there was only a small group of 88 CNV calls selected for validation via qPCR. This was due to our limited capacities to do these validations. Even though the numbers are not very high here, it can be seen that the bivariate distribution of CNV calls on their targets' mappability has changed a bit. There is no great group that has a mappability close to zero.

S10 Supplementary data description

The first table (https://github.com/bihealth/clear-cnv-supplementary/blob/master/data/all_cnv_calls.xlsx) summarizes all CNV call by all four tools (clearCNV, CoNVaDING, ExomeDepth, and panelCN.MOPS). The columns describe the following features: **chr**: chromosome; **start**: start coordinate on the chromosome from 5' end; **end**: end coordinate from 5' end; **aberration**: type of CNV call. Only DEL for deletion and DUP for duplication are called; **score_[tool]:** the score assigned to a CNV call by the specified tool; **ranks_[tool]:** the rank we computed from a tool's score to compare across all tools.

The second table (https://github.com/bihealth/clear-cnv-supplementary/blob/master/data/validated_cnv_calls.xlsx) summarizes all CNV calls that were validated via qPCR. All selected CNV calls were done by at least three of the selected tools. The columns describe the following features: **chr**: chromosome; **start**: start coordinate on the chromosome from 5' end; **end**: end coordinate from 5' end; **aberration**: type of CNV call. Only DEL for deletion and DUP for duplication are called; **ranks_[tool]:** the rank we computed from a tool's score to compare across all tools; **[tool]**: the CNV was called by the specified tool; **size**: the size of the called CNV in bp; **qPCR_validated**: the status of the qPCR validation. 'not' means that the result indicated no change in the copy number. 'ambiguous' means that the result

indicated no wild type nor copy number variant. ‘contrary’ means that the opposite CNV was indicated in the qPCR result. ‘no_sample’ means that the sample was not available.