Chapter 7

WISeXome: A within-sample comparison approach to detect copy number variations in whole exome sequencing data

Roy Straver, Marjan M. Weiss, Quinten Waisfisz, Erik A. Sistermans and Marcel J.T. Reinders

This chapter has been published in European Journal of Human Genetics. 2017 Dec;25(12):1354-1363. doi: 10.1038/s41431-017-0005-2.
Abstract

In clinical genetics, detection of single nucleotide variants (SNVs) as well as copy number variations (CNVs) is essential for patient genotyping. Obtaining both CNV and SNV information from WES data would significantly simplify clinical workflow. Unfortunately, the sequence reads obtained with WES vary between samples, complicating accurate CNV detection with WES. To avoid being dependent on other samples, we developed a within-sample comparison approach (WISExome). For every (WES) target region on the genome, we identified a set of reference target regions elsewhere on the genome with similar read frequency behavior. For a new sample, aberrations are detected by comparing the read frequency of a target region with the distribution of read frequencies in the reference set. WISExome correctly identifies known pathogenic CNVs (range 4 kb to 5.2 Mb). Moreover, WISExome prioritizes pathogenic CNVs by sorting them on quality and annotations of overlapping genes in OMIM. When comparing WISExome to four existing CNV detection tools we found that CoNIFER detects much fewer CNVs and XHMM breaks calls made by other tools into smaller calls (fragmentation). CODEX and CLAMMS seem to perform more similar to WISExome. CODEX finds all known pathogenic CNVs, but detects much more calls than all other methods. CLAMMS and WISExome agree the most. CLAMMS does, however, miss one of the known CNVs and shows slightly more fragmentation. Taken together, WISExome is a promising tool for genome diagnostics laboratories as the workflow can be solely based on WES data.

Introduction

Recent technological breakthroughs in DNA analysis methods have not only had a huge impact on genetic research, but also on Genome Diagnostics. Finding a genetic diagnosis is important as it helps patients and family members in understanding the disease, supports the search for possible treatments and determines reproductive options in subsequent pregnancies [1]. Currently, most patients with genetic disorders are tested through standard practices such as array-based techniques for detecting copy number variations (either by array comparative genomic hybridization, array-CGH, or by a single nucleotide polymorphism array, SNP-array) [2], or Sanger sequencing of single genes [3, 4], but these methods do not always provide a diagnosis. This has recently changed through the upswing of Next Generation Sequencing (NGS), which allows for the parallel sequencing of gene panels, whole exomes (WES, Whole Exome Sequencing), or whole genomes (WGS, Whole Genome Sequencing). Although WGS obtains nearly the whole genomic sequence of a patient, its costs are currently still too high for routine testing. As an affordable alternative, WES captures exon-specific regions, called targets, and uses target-specific probe sets to read-out these targets (Figure 1a). While providing a highly accurate way to obtain Single Nucleotide Variation (SNV) information [5], WES data does not allow for straightforward Copy Number Variation (CNV) analysis. The main reason being the non-uniform distribution of reads because: 1) target regions cover only 2% of the genome
[6, 7], and 2) the varying amplification efficiency of target regions [8, 10]. Moreover, this effect is not consistent over different samples as quality of DNA and environmental differences during sample preparation directly influence probe effectiveness [11]. Consequently, additional array analysis is still used to obtain CNV information for a patient. WES diagnostics would thus greatly benefit from a reliable tool to obtain CNV information from WES data only, as it would eliminate the need for separate analyses.

Basically, CNVs in WES data can be detected by comparing the read count for a region to the expected read count distribution for that region, representing possible variations in read counts when assuming a diploid genome for that region. When the observed read count for a region differs significantly from the expected read count, this region is designated to be aberrant, either amplified (when having more reads than expected) or deleted (when having less). Generally, the expected read count distribution is derived from a training set of normal (diploid) samples. Several methods for CNV detection in WES data have been developed [12–14], all based on this same principle. For clinical genetic diagnostics of rare diseases, it is important to also detect few and small CNVs, as opposed to mostly large and highly abundant CNVs in cancer [13]. As tools for cancer diagnostics aim to find the distribution of large CNVs as well as their ploidy, they lack the precision required to find intra-genic CNVs. CoNIFER [15] and XHMM [16] are generally believed to perform well for identifying CNVs in a genetic diagnosis setting, based on several comparison studies [10, 17, 18]. Recently, CLAMMS [19] and CODEX [20] have been introduced that claim to outperform CoNIFER and XHMM.

CNVs are generally detected using a training set of normal (diploid) samples to capture the expected read count distribution. Consequently, next to experimental variation, the expected read count distribution also captures the between-sample (biological) variations, which principally would not be necessary. Avoiding the incorporation of the between-sample variation in the expected read distribution potentially increases the sensitivity of CNV detection. We previously developed WISECONDOR for trisomy detection on cell-free fetal DNA based on a within-sample comparison to deal with the fluctuations in read distribution [21]. This within-sample comparison approach assumes that all target region amplifications within a sample undergo the same experimental variations. Hence, if we know which target regions, elsewhere on the genome, respond similarly to the experimental variation, these regions can serve as a reference set and the expected read count distribution can be derived from read counts across this set of reference target regions as measured within the same sample. Here, we present WISEExome: a CNV detection method for WES data based on this within-sample comparison principle. Figure 1 gives an overview of the complete procedure and a more elaborate description can be found in the methods section.

We show that WISEExome successfully replicates array analysis and compares favorably to other CNV detection tools. Together the results suggest that WISEExome can be used in diagnostics as replacement of array, removing the requirement for separate array analysis for WES diagnostics.
Materials & Methods

Sample preparation

Whole exome sequencing was performed as previously described [22]: Genomic DNA was isolated from blood for 336 samples and prepared using the SeqCap EZ Human Exome Library v3 kit (Roche, Basel, Switzerland), then sequenced using an Illumina HiSeq 2500. Reads were mapped with BWA (0.7.10) to Hg19 [23]. We removed duplicate reads (as marked by Picard Tools 1.111), reads with a mapping quality below 30, and reads that were not part of a read-pair. Samples were split into a training set of 319 samples and a test set of 17 samples.

Array analysis

Array analysis was carried out on the high-resolution CytoScan HD array platform (Affymetrix, a part of Thermo Fisher Scientific, Santa Clara, CA, USA) according to manufacturer’s protocols. This array consists of over 2.6 million copy number markers. Analysis was done using Nexus software (BioDiscovery, El Segundo, CA, USA), using SNPRank Segmentation with a minimum of 20 probes per segment and the significance threshold set at 1e-5.

WISExome

WISExome determines excess of target read counts (enriched or depleted) as compared to the expected target read count based on a reference set of target regions within the genome. In the following, we explain the different steps of WISExome, shown in Figure 1: 1) determining the target region counts from WES read data; 2) identification of the set of reference target regions for a target region; 3) CNV detection; and 4) fine-tuning identified CNVs using a segmentation algorithm. Finally, we explain WISExome’s scoring metric for called CNVs and its annotation of CNVs.

Target region read count

In WES data, DNA is fragmented, specific target regions within exons are captured, and subsequently enriched before sequencing (Figure 1a, target level). For every target region, there are several probes that are designed to recognize unique subsequences (Figure 1a, probe level). Paired-end sequencing of the probe-selected fragments covers fragments (Figure 1b), providing mappable reads. As experimental variation mostly influences the target region enrichment, we are interested in read counts for target regions instead of the probe, fragment or exon level (the latter is commonly used in other CNV detection tools). Mapped fragments were linked to the closest target region based on the distance between the center of the mapped fragment and the center of a target region (Figure 1b). To ensure specificity, we require an overlap of at least 20 bp of the mapped fragments.
with the linked target region. Target region counts, being the sum of mapped fragments linked to that target region, are normalized by the total target region count over the whole genome in the sample, resulting in target region count frequencies.

**Creating a reference target region set**

The basic idea behind the within-comparison approach is that for every target region, we find target regions on other parts of the genome that behave similarly to experimental variations; the reference target region set. By assuming a sparse number of CNVs, the target region read counts of the reference set represent the within-sample variation of a diploid read count for the associated target region. To find similarly varying target regions, we make use of a training set of samples with no known notable phenotypes. Note that this training set is only used once, to identify reference target regions. Because of lab specific changes in read depth behavior this set of reference samples is best obtained from the same lab as the test samples. During testing, we only use information of the test sample, and not the observed read counts in the training samples as other CNV detection tools do. For a target region of consideration, we correlate the observed read counts over all training samples to the read counts of all other target regions. Here, we used the squared Euclidean distance on the target read count frequencies across the 319 training samples (Figure1c), and selected the 100 target regions with the lowest distance to build-up the reference set for the target region of consideration. We do this for all target regions, so every target region has its own 100 reference target regions. For target regions on the X chromosome, we find their reference target regions considering read count variations across female training samples only. To avoid reference target regions overlapping the CNV, we require that reference target regions lie on other chromosomes than the target region for which the reference set is being build. To avoid that the reference target regions are not similar enough to the target region of consideration, we prune the list of 100 reference target regions. First, the mean and variance of the squared Euclidean distances of the closest reference target for every target region (i.e. the distances of the top-1 target regions in each reference set) is calculated. Then, for every reference set, we remove target regions from the reference set that have a distance larger than the mean plus three times the standard deviation of the top-1 distances ($z$-score $>3$). As a result, the number of reference target regions will differ for every target region, see Figure1d. Those that have less than 10 references are ignored in further analysis, and are denoted as unreliable target regions (Supplementary Figures S1, S2 and S3a show results do not change much when varying this setting).

**Finding CNVs**

For each target region, its own reference set specifies an expected read count distribution. Hence, we can statistically test whether the observed read count of the target region points towards the region being aberrant or not. For this purpose, we use a $z$-score with the target region read count frequency as input, and with the mean and variance estimated
Figure 1: Caption on the next page.
a. Overview of all regions of importance for WES data. Genes (gray boxes) consist of exons (orange boxes) which are covered by target regions (light orange boxes), for which probes are designed (red boxes) that target a unique sub-sequence of the fragment. Paired-end reads (blue boxes) cover fragments. b. Determination of target region read count: a detected fragment is mapped on the reference genome, and consequently assigned to the nearest target region (orange box) based on the center of the fragment (gray lines). c. Selection of reference target regions based on the difference in read counts over a set of training samples. For illustrative purposes, we show only 1 sample and a few target regions on three different chromosomes. The selected reference target regions for the target regions on chromosome 2 are indicated by the straight colored lines. In this example only reference target regions are considered when the read count differs at most 1 read. Dotted lines are tested but showed a larger difference. d. Schematic overview of the selected reference target regions for the target regions on chromosome 2 as shown in panel c. e. Based on its reference set, a $z$-score for every target region can be calculated (level window size 1). $Z$-scores of neighboring target probes are aggregated using a combined $z$-score (here Stouffer’s $z$-score) for different numbers of neighbors, using odd window sizes from 3 to 15. f. Target regions with a significant $z$-score in any of the windows are marked and kept for further analysis (blue shaded boxes). g. For every target region a final $z$-score is determined by taking the maximal (positive or negative) $z$-score across all window-sizes (at the target region position). h. Stretches of significantly aberrated target regions (blue shaded boxes) are marked as a putative CNV segment. i/j/k. To fine-tune the borders of the CNV, the putative CNV segment is considered with all non-putative CNV regions on the same chromosome (i shows the extension, j prunes the putatively aberrated CNVs, k is the result). l. To find the exact borders of the CNV, the location of borders of the putative CNV are changed over all possible positions within and directly neighboring the putative CNV region (shaded blue lines in l). The segment that shows the largest difference in mean effect size within the segment compared to the mean effect size outside the segment is selected as the aberrated CNV segment (dark blue line in l). m. For visualization purposes, the $z$-scores (vertical axis) for each target region are plotted across their genomic position (x-axis). Designated aberrated target regions (regions within a detected CNV segment) are colored blue, others gray. The bottom line shows the position of the exons (orange boxes).
from the read count frequencies of the reference target regions, measured in the same sample. A target region with a \( z \)-score larger than 5.64 (family-wise error rate (FWER) corrected significance level of 0.05, see Supplementary Section SM1) is considered to be amplified, and a target region with a \( z \)-score smaller than -5.64 is considered to be deleted. Applying this procedure to the (normal) training samples showed that some target regions are frequently being called, probably due to either large variations in target amplification or because they are part of a common CNV. We decided to exclude target regions being called in more than 4 of the training samples from further analyses (also denoted as unreliable target regions). This removed 4,226 (1.15%) out of 366,795 target regions (Supplementary Figures S1, S2 and S3b show that WISExome is not sensitive to increasing this threshold). As CNVs generally will be larger than target regions, we improved sensitivity by aggregating \( z \)-scores of neighboring target regions. For every target region, we calculated a combined \( z \)-score (here, Stouffer’s Z) for differently sized windows (odd window sizes up to 15; larger windows did not change the results, data not shown), see Figure 1e/f. The \( z \)-score for the target region is then set equal to the largest (positive or negative) \( z \)-score across all windows, see Figure 1g. A target region is called aberrant if the aggregated \( z \)-score is significant (absolute value larger than 5.64), see Figure 1h. Consecutive called target regions then make up a CNV. Finally, we take the effect size of the aberration across the CNV into account. Hereto, first, the effect size of a target region is defined by the target region read count divided by the expected target region read count (based on the reference set). The effect size of the CNV is subsequently defined as the median effect size across the target regions it covers. Only CNVs that deviate more than (the arbitrarily chosen) cutoff of 35% from its expectation (i.e. CNV effect size smaller than 0.65 or larger than 1.35) are considered true CNVs.

Fine-tuning detected CNVs

Due to the aggregation of neighboring target regions into the \( z \)-score of a target region (the windowing), the borders of the CNVs will not be precise. For example, a strongly aberrant target region might cause that the aggregated \( z \)-score of neighboring target regions also becomes significant. In other words, the aggregation improved sensitivity at the cost of precision. Hence, we wanted to fine-tune the borders of every detected CNV to improve the precision of the CNV calls. For that we devised a segmentation algorithm. Every detected CNV is first extended with eight target regions on each side, Figure 1i. For all possible segmentations in this region (i.e. all possible start and end positions of the putative CNV), the mean effect size of target regions within the segmentation are compared to the effect sizes of all target regions on the same chromosome that are called un-aberrated, using a student’s \( t \)-test with a pooled variance (Supplementary Section SM2). The segmentation that maximizes this student’s \( t \)-test is chosen as the fine-tuned CNV, see also Figures 1i-l. Finally, we require that the mean effect size of the fine-tuned CNV, again to be at least 35%.
Quality score

Every call is annotated with a quality score reflecting the percentage of reliable target regions. For this score, we consider whether the neighboring target regions are unreliable. This is done because these unreliable neighboring target regions influence the fine-tuning of the CNV borders. The quality score for the CNV is then the number of reliable target regions minus the sum of the number of unreliable target regions covered by the CNV and the number of all unreliable neighboring target regions. Hence, a detected CNV with many unreliable neighboring target regions, or a detected CNV with many scattered unreliable target regions within the CNV will get low scores. The default setting for the CNV quality score we used was 6 (six more reliable target calls than unreliable calls).

CNV annotation

Additionally, calls are annotated with OMIM phenotype key scores of all (partially) overlapping genes using the OMIM API [24], i.e. the OMIM score for the CNV equals the maximum OMIM phenotype scores of the genes that overlap with the CNV. This score describes which method is used to link the gene to the disorder and reflects the certainty of a gene causing a specific disorder. For example, if the molecular basis of the disorder is known, this key is 3. If the gene is linked to a disorder through statistical methods only, the key is 2.

Other tools

We ran XHMM (downloaded from GitHub @ 18 June 2015), CoNIFER (version 0.2.2; released 9/17/2012), CODEX (GitHub, commit 3d40ac9 @ April 7 2017) and CLAMMS (GitHub, commit 3e19892 @ April 10 2017) according to their default settings. All tools were run on the same samples as WISExome, as described in Sample preparation. XHMM and CoNIFER do not distinguish between training and test samples, CODEX and CLAMMS used the same division in training and test samples as WISExome. Additional information on decisions and settings can be found in the Supplementary Section SM3.

Results

Replication array analysis

To test for compliance with array and MLPA analysis, we tested WISExome on 17 test samples with at least one known pathogenic CNV each (20 CNVs total) as identified by array analysis (18 CNVs) or MLPA (2 CNVs, kit P170-B2 for the APP gene, MRC-Holland, Amsterdam, The Netherlands). We were able to correctly identify all known pathogenic CNVs as shown in Supplementary Table S1 and Supplementary Figure S4. There are differences in start and end positions, but these are mostly because the array...
platform has probes in inter-exonic regions whereas the WES-probes lie in exons only. On average, WISEXome finds 33 calls per sample without size-filters, and 15 calls per sample when filtering at a minimum of 15 kb (standard array resolution in current clinical practice), which is comparable to the array analysis.

As several tools have been developed to call CNVs from exome data, we compared WISEXome to XHMM, CoNIFER, CODEX and CLAMMS (using default settings, see Methods). An overview of how their calls overlap with known CNVs, as validated by array analysis, is shown in Figure 2, Supplementary Figure S4 and Supplementary Table S1.

CoNIFER made few calls and only five overlapped the known CNVs in our test data. Adjusting the SVD argument yielded slight variations in results, but no setting was satisfactory. One of the few calls CoNIFER did make is shown in Figure 2a. While other tools, except for WISEXome, found a smaller area to be affected by the known CNV, CoNIFER appears to overestimate the CNV length.

XHMM identified most known CNVs partially (Figure 2a, 2b) and failed to identify one known deletion and three duplications (numbered 13, 15, 17 and 19 in Supplementary Table S1, respectively), one missed CNV is shown in Figure 2c, all results are shown in Supplementary Figure S4. Most notably, XHMM tends to break up single CNV calls of the array in multiple small regions.

CODEX showed strong similarities to WISEXome in results; it calls all CNVs known from the array analysis (Supplementary Table S1, Supplementary Figure S4). While generally extremely sensitive, Figure 2a shows that CODEX only found the rightmost half of the CNV, while WISEXome, XHMM and CoNIFER agree on the upstream start position for this CNV.

CLAMMS also showed very similar behavior to WISEXome (Supplementary Table S1, Supplementary Figure S4). However, CLAMMS was unable to identify one known CNV in our dataset, shown in Figure 2a. This CNV was not missed by any of the other tools, including CoNIFER.

Differences in detected regions

As it is unlikely that the array caught all CNVs in our data, we extended our comparison to include all calls made by the selected tools. Figure 3a shows an overview of the size of the detected regions by the different tools, whereas Figure 3b shows the number of detected regions. Most tools detect roughly the same number of regions, except CODEX which makes considerably more calls. WISEXome and CoNIFER show relatively few and large calls. CLAMMS and XHMMs show fragmented calling behavior which is reflected by their small median call size (Figure 3a, Supplementary Figure S5).

To find out how the detected regions differ between tools, we compared the size of the union of the exons detected by a set of tools with the size of the intersection of the detected exons by these tools. When tools agree, their intersection should equal their union. The results for every possible combination of tools is shown in Figure 4b. The
Figure 2: Detected CNV segments by WISExome (blue), CLAMMS (orange), CODEX (green), XHMM (Red) and CoNIFER (purple) for three known pathogenic CNVs according to the array analysis (brown). The region is annotated by the array probes (cyan), genes (citron), exons (gray) and target regions (pink). CNVs shown here are marked in supplementary Table S1 as numbers 18 (panel a), 7 (panel b) and 19 (panel c). (a) CLAMMS failed to identify this region as aberrated. WISExome, XHMM and CoNIFER mark roughly the same area as part of a CNV, while CODEX made a relatively small call. (b) WISExome, CLAMMS and CODEX are in near perfect agreement on the CNV. XHMM shows a very fragmented call, failing to identify several affected genes in between its calls. CoNIFER does not make a call. (c) WISExome, CLAMMS and CODEX all identify the region to the right as aberrated, while the CNV should have been shorter according to the Array. Both XHMM and CoNIFER fail to make a call.
Figure 3: a. Boxplot showing the size distribution (vertical axis) of calls made by the five CNV detection tools (horizontal axis). The annotated numbers next to boxplots shows the median size for that tool. CoNIFER has the least fragmentation in its calls, and XHMM the most. CLAMMS calls are also fragmented and have a median size of only 1133 bp. To improve visibility of the box plots, outliers above 250,000 were cropped. Supplementary Figure S5a shows this boxplot without the crop. b. Number of amplifications and deletions per tool after thresholding, clearly showing CODEX’s huge number of calls compared to other tools.

Differences in detected genes and exons

Figure 4a shows the comparisons between WISExome and any of the other tools with respect to the number of genes detected as aberrated, whereas Supplementary Figure S6 shows these comparisons for aberrated exons. A gene/exon is called aberrated when a call of a tool shows at least one base pair overlap with that gene/exon. It immediately becomes clear that WISExome and CLAMMS behave relatively similar, although CLAMMS misses a few more genes found detected by array analysis (Figure 4a). CODEX found several of the array based detected genes missed by WISExome, but at the cost of a huge set of unique calls. Both CoNIFER and XHMM miss out on a lot of genes detected by the array analysis. Fragmentation by CLAMMS and XHMM can also be seen in the relatively small number of detected genes and exons (shown in Supplementary Figure S7) compared to the total number of calls as shown in Figure 3b.

To further zoom in on differences, supplementary Figure S8 shows the overlap between WISExome, CLAMMS and CODEX. This figure again shows that CODEX calls significantly more genes/exons aberrated than other tools. WISExome and CLAMMS show a
a. Union vs Intersection: All Exons

b. Figure 4: Caption on the next page.
a. Venn diagrams showing overlap based on the detected aberrated genes between WISExome and any of the four other CNV detection tools, as well as overlap with the known CNVs according to the array analysis. For known CNVs, CLAMMS shows a strong overlap with WISExome but adds no unique calls that overlap with the array, whereas WISExome shows 15 unique calls wrt CLAMMS that do overlap with the array results. CODEX does find 24 more affected genes (than WISExome) overlapping with the array data, but has a significantly larger set of affected genes in total. XHMM misses half of the genes called by the array analysis, and CoNIFER even misses three quarters of the known aberrated genes. An exon level based equivalent of this figure is shown in Supplementary Figure S6. 

b. Plot showing the overlap in detected exons between any combination of tools. The horizontal axis shows the size of the union of the exons detected by a set of tools, and the vertical axis shows the size of the intersection of the exons detected by the same set of tools. Lines marked with numbers show $y = ax$, where $a$ is the number shown. When the set of tools agree the size of the intersection should equal the size of the union, which would put the marker on the line marked with 1. A unique color and shape is used for every tool, and plotted on top of each other for every tool involved in a combination. For example, the top-right is marked by both CLAMMS and CODEX, meaning that the unions and intersection of these two tools are considered. The bottom right shows the result when comparing all five tools. Due to the number of affected exons called by CODEX, two main groups can be observed: the combinations on the right of the plot that include CODEX, and the combinations on the left of the plot that exclude CODEX. Next, vertically another two groups can be observed: all combinations including CoNIFER or XHMM below an intersection of size 1000, and combinations between WISExome, CLAMMS and CODEX above this line. WISExome and CLAMMS are most similar since the size of their intersection is $1/4$th of the size of their union, which is considerably larger than any other combination. A similar plot where overlap among genes is considered is shown in Supplementary Figure S11.
larger overlap, but WISExome’s unique calls are more overlapping with array results than CLAMMS’ unique calls (also visible in Figure 4a).

**Prioritizing WISExome calls**

Within WISExome, we made it possible to prioritize calls based on its scoring scheme that measures the difference between the amount of reliable and unreliable called target regions (Methods, and Supplementary Figure S9 shows a distribution of scores). Prioritizing calls based on this score puts all 20 known pathogenic CNVs at the top of the list with minor exceptions; one CNV was ranked second (CNV 9 in Supplementary Figure S4), two CNVs were ranked 5th (CNV 15 and 18), and the smallest CNV was ranked 25th (CNV 20). We further annotated calls with their potential pathogenicity based on the OMIM phenotype key of the underlying genes (Methods). By lowering CNVs that have no genes with a high OMIM phenotype key, and thus are not known to be involved in any syndrome, a large fraction of the calls can be ignored (Supplementary Figure S10), leaving only a few calls (5 calls on average) per sample to be inspected in more detail. Note that this prioritization will also rank common copy number variations lower as those are not expected to harbor pathogenic genes.

**Conclusion**

We developed a new CNV detection methodology for WES data that uses a within-sample comparison approach to capture the expected read count distributions across the genome. The benefit of this approach is that we only need to describe experimental variation and not in-between sample variation, making the method more accurate than previous approaches. We have shown that this new methodology, called WISExome, reliably reproduces array results, opening the possibility to perform genome diagnoses using WES data exclusively, in contrast to current practice where WES analysis still has to be combined with array analysis.

We compared WISExome to four existing CNV detection tools: CoNIFER, XHMM, CODEX, CLAMMS. From these tools, CoNIFER deviates the most, calling few CNVs in general, and it detects only a few CNVs that result from clinical interpretation of the array analysis. XHMM breaks calls made by other tools into smaller calls (fragmentation) and misses a few CNVs detected by array analysis. Note that fragmentation of calls might lead to overlooking pathogenic genes.

CODEX found all CNVs detected by array analysis, but made significantly more calls than any of the other tools, resulting in the detection of many more genes and exons than other tools. CLAMMS showed results similar to WISExome, both in the start and stop positions of the CNVs resulting from array analysis, as well as the numbers of detected genes and exons. Yet, CLAMMS shows more fragmentation, which can be clearly seen in the size distribution of the calls. Additionally, CLAMMS missed one of the CNVs detected by array analysis that was found by all other tools (sample 18, Figure 2a).
From this comparison, we conclude that WISExome performs consistently well over the different analysis that we investigated, i.e. it shows a proper balance between accuracy and specificity, without fragmentation of regions. These are favorable aspects when applying a CNV detection tool within the clinic. Nevertheless, we have observed considerable variability between tools, and we recommend to run multiple tools in parallel for clinical practice.

Note that the within-comparison approach might suggest that WISExome is independent for different sequencing or enrichment protocols, and that it would be possible to run WISExome without (re)training in different centers. Our experiences with WISExome and WISECONDOR (the first method in which we introduced the within-comparison approach, although designed for detecting chromosomal aberrations in cell-free DNA) indicates that even the within-comparison scheme is influenced by differences between centers. That is, although the within-comparison scheme works across different centers and sequencing technologies, we have seen performance improvements when reference bins are determined using training samples that were processed similar to the test cases. This should, however, be tested more thoroughly by setting up an inter-center comparison. For now, we advocate to create a new reference set table when there is a change in center or protocol. However, we suspect WISExome can be trained with a relatively small training set (until now we typically used 200-300 samples). Furthermore, we would like to stress that by using a within-comparison scheme our tool is likely still capable of detecting CNVs in cases where the read distribution of a sample deviates strongly from the reference data. More so than when the sample is compared to the reference set directly, as the reference targets from the within comparison scheme vary accordingly within the sample.

Finally, we introduced two ways to prioritize calls made by WISExome. One that expresses the quality of the call, which is directly influenced by the size of the call, the number of genes and the amount of probe targets, and another that annotates the call with OMIM phenotype keys. The prioritization of calls based on their score and annotation, allows geneticists to quickly zoom in on the most likely candidate genes, making WISExome an extremely useful diagnostic tool. To even further reduce the overall time spent per sample for a geneticist additional filtering of calls might be done by, for example, filtering common CNVs, or annotating calls using other databases, such as the Copy Number Variation in Disease (CNVD) database [25], or the Database of Genomic Variants [26].

Taken together, WISExome provides an alternative to array analysis with a quick and easy workflow for geneticists that includes a prioritization scheme for calls that improves the diagnostic relevance.
References


### Table S1: Overview of all known pathogenic CNVs in 17 test samples.

The first two columns indicate CNV index and sample identifier. The array based information is shown in the next four columns, showing the chromosome (Chr), start and stop positions (based on Hg19), and whether the call was an amplification or a deletion (Amp/Del). Next, for every tool the start and end positions for calls overlapping the array calls are shown, as well as the number of calls that overlapped. Whenever multiple calls were made, the reported start position is the first base pair position of the first call with overlap, the stop position is the last base pair of the last call overlapping the array CNV, and the number of overlapping calls is shown in the corresponding Calls column.

<table>
<thead>
<tr>
<th>ID</th>
<th>Sample</th>
<th>Chr</th>
<th>Start</th>
<th>Stop</th>
<th>CNV</th>
<th>Amp/Del</th>
<th>WISE</th>
<th>XHMM</th>
<th>CODEX</th>
<th>CLAMMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>15</td>
<td>23805000</td>
<td>28771100</td>
<td>Del</td>
<td>7866980</td>
<td>7869175</td>
<td>7867119</td>
<td>5</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>9</td>
<td>214387</td>
<td>4566187</td>
<td>Amp</td>
<td>64174</td>
<td>4566131</td>
<td>36404</td>
<td>456326</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>7</td>
<td>500000</td>
<td>2838078</td>
<td>Del</td>
<td>283325</td>
<td>2838333</td>
<td>20781</td>
<td>282343</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>15</td>
<td>3307136</td>
<td>32954230</td>
<td>Snv</td>
<td>3092872</td>
<td>3240457</td>
<td>12</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>E</td>
<td>8</td>
<td>8774423</td>
<td>8754518</td>
<td>Amp</td>
<td>8750463</td>
<td>8319362</td>
<td>12</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>F</td>
<td>X</td>
<td>6448777</td>
<td>8146498</td>
<td>Amp</td>
<td>6461945</td>
<td>8093512</td>
<td>11</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>G</td>
<td>20</td>
<td>5400500</td>
<td>2388997</td>
<td>Del</td>
<td>5902535</td>
<td>212286</td>
<td>17</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>H</td>
<td>22</td>
<td>2072371</td>
<td>2195132</td>
<td>Amp</td>
<td>2075521</td>
<td>2166325</td>
<td>22</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>I</td>
<td>6</td>
<td>34551339</td>
<td>146280217</td>
<td>Amp</td>
<td>3456998</td>
<td>146280217</td>
<td>134572405</td>
<td>148350806</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>J</td>
<td>7</td>
<td>6010735</td>
<td>6019366</td>
<td>Amp</td>
<td>6023386</td>
<td>6434453</td>
<td>602362</td>
<td>6231554</td>
<td>8</td>
</tr>
<tr>
<td>11</td>
<td>K</td>
<td>3</td>
<td>23542268</td>
<td>25376261</td>
<td>Amp</td>
<td>23542372</td>
<td>25368195</td>
<td>23540245</td>
<td>2364828</td>
<td>12</td>
</tr>
<tr>
<td>12</td>
<td>L</td>
<td>4</td>
<td>27525861</td>
<td>27543138</td>
<td>Amp</td>
<td>2753167</td>
<td>2784039</td>
<td>2752157</td>
<td>7</td>
<td>-</td>
</tr>
<tr>
<td>13</td>
<td>M</td>
<td>16</td>
<td>28732228</td>
<td>28982277</td>
<td>Amp</td>
<td>2873998</td>
<td>28982277</td>
<td>2873998</td>
<td>28982277</td>
<td>1</td>
</tr>
<tr>
<td>14</td>
<td>N</td>
<td>1</td>
<td>23780746</td>
<td>23800083</td>
<td>Amp</td>
<td>23781166</td>
<td>237997180</td>
<td>23781166</td>
<td>237997180</td>
<td>1</td>
</tr>
<tr>
<td>15</td>
<td>O</td>
<td>15</td>
<td>2818074</td>
<td>2809938</td>
<td>Del</td>
<td>2777558</td>
<td>2967517</td>
<td>1</td>
<td>-</td>
<td>0</td>
</tr>
<tr>
<td>16</td>
<td>P</td>
<td>9</td>
<td>107589595</td>
<td>107585350</td>
<td>Amp</td>
<td>10758666</td>
<td>107585350</td>
<td>10758666</td>
<td>107585350</td>
<td>3</td>
</tr>
<tr>
<td>17</td>
<td>Q</td>
<td>16</td>
<td>89831821</td>
<td>89841352</td>
<td>Amp</td>
<td>8983110</td>
<td>8984234</td>
<td>8984205</td>
<td>89842197</td>
<td>1</td>
</tr>
<tr>
<td>18</td>
<td>R</td>
<td>5</td>
<td>14022318</td>
<td>14028656</td>
<td>Del</td>
<td>14022318</td>
<td>14028656</td>
<td>14022318</td>
<td>14028656</td>
<td>1</td>
</tr>
<tr>
<td>19</td>
<td>S</td>
<td>5</td>
<td>1500958</td>
<td>150098482</td>
<td>Del</td>
<td>1500958</td>
<td>150098482</td>
<td>1500958</td>
<td>150098482</td>
<td>1</td>
</tr>
<tr>
<td>20</td>
<td>T</td>
<td>5</td>
<td>7867073</td>
<td>7871862</td>
<td>Del</td>
<td>7866980</td>
<td>7870595</td>
<td>7866980</td>
<td>7870595</td>
<td>1</td>
</tr>
</tbody>
</table>
Figure S1: Venn diagram of overlapping genes (left) and exons (right) when running WISExome with three settings of pre-filtering of target regions. WISExome’s default setting is: pruning target regions with less than 10 reference target regions, as well as target regions that are called in more than 4 training samples. Top row shows two different settings for the number of reference targets: (1) HALF: prune target regions with less than 5 reference target regions, and (2) DOUBLE: prune regions with less than 20 reference target regions. The bottom row shows two different settings for the number of calls in the training set: (1) HALF: prune target regions that occur more than 2 times in the training set, (2) DOUBLE: prune regions that occur more than 8 times in the training set.
Figure S2: Similar to Figure S1, showing the effect of two of the pre-filtering thresholds. Here, the calls of WISExome for all the different setting are first filtered to require overlap with the regions detected by the array analysis. This way we can inspect how the different filter settings influence results that overlap with the array analysis.
Figure S3: Effect of thresholds on the minimum number of reference target regions (a) and the maximal number of calls within the training set (b) on the number of reliable target regions (vertical axis).
Tool comparison for CNV: 6, Sample: E, Type: Amplification

Tool comparison for CNV: 7, Sample: F, Type: Deletion

Tool comparison for CNV: 8, Sample: G, Type: Amplification

Tool comparison for CNV: 9, Sample: H, Type: Amplification

Tool comparison for CNV: 10, Sample: I, Type: Amplification
Tool comparison for CNV: 11, Sample: J, Type: Amplification

Tool comparison for CNV: 12, Sample: K, Type: Amplification

Tool comparison for CNV: 13, Sample: L, Type: Deletion

Tool comparison for CNV: 14, Sample: J, Type: Amplification

Tool comparison for CNV: 15, Sample: M, Type: Deletion
Figure S4: Detected CNV segments for all tools for all CNVs reported by clinical geneticist using array analysis. CNV numbers and sample identifiers match those in Supplementary Table S1. The region is annotated by the array probes (cyan), genes (citron), exons (gray) and target regions (pink).
Figure S5: The same boxplot as shown in Figure 3a in the manuscript, except no cropping was performed. Numbers inside the figure annotate the actual median for every tool.
Figure S6: Overlap of exons affected by calls by WISExome, array analysis and either CLAMMS, CODEX, XHMM, or CoNIFER.
Figure S7: Number of genes and exons affected by amplifications and deletions per tool after thresholding. (a) Number of genes affected by calls per tool. (b) Number of exons affected by calls per tool.

Figure S8: Overlap of (a) genes and (b) exons affected by calls by WISExome, CLAMMS and CODEX.
Figure S9: Density plots for the quality scores that are generated by the different CNV detection tools for the calls that they generate. CoNIFER is excluded as it does not provide a quality score per region. (a) WISExome’s, (b) CLAMMS’s, (c) CODEX’s and (d) XHMM’s quality score distributions. Every distribution is fitted with a kernel density estimate.
Figure S10: Effect on number of calls per sample when filtering on WISExome’s quality score and OMIM phenotype key $\geq 3$. Data is shown for all calls (gray), and for amplifications (blue) and deletions (red) separately. Light colors show the effect of applying only the quality filter, darker colors show the amounts of calls when overlap with at least one gene with an OMIM phenotype key of $\geq 3$ is also required.

Figure S11: Plot showing the overlap in detected genes between any combination of tools. The figure is the equivalent of Figure 4b where overlap in exons is shown instead. For a detailed description of the figure we therefore refer to Figure 4b.
Figure S12: The scree plot used to determine the value of the SVD variable for CoNIFER.
SM1. Corrected significance threshold

Here we derive the used thresholds for significance.
First, we applied FWER using these settings:

\[
\begin{align*}
\alpha & \ 0.05 \quad \text{Significance threshold} \\
N & \ 366795 \quad \text{Number of targets} \\
w & \ 8 \quad \text{Number of windows}
\end{align*}
\]

Multiplying the probe count by the number of windows tested we obtained the total number of tests \( T \) done:

\[
T = N \cdot w
\]  

(1)

We determined the \( p \)-value required for this number of tests:

\[
\frac{1}{T} \cdot \alpha
\]  

(2)

Which equals to:

\[
\frac{\alpha}{N \cdot w}
\]  

(3)

Using the numbers for our settings we obtain:

\[
\frac{0.05}{366795 \cdot 8} = 1.7e - 8
\]  

(4)

Looking up the corresponding \( z \)-score for a two-tailed \( p \)-value of 1.7e-8 we found a corresponding \( z \)-score threshold of 5.64.
SM2. Student’s $t$-test with a pooled variance

This part is meant to explain the mathematical background to the segmentation algorithm we applied to fine-tune the borders of detected aberrations.

For this we use the following variables:

- $X_1$ samples on the segmented region (considered to be aberrated)
- $X_2$ samples outside the segmented region (not aberrated)
- $m_1$ mean of samples of $X_1$
- $s_1^2$ variance of samples of $X_1$
- $m_2$ mean of samples of $X_2$
- $s_2^2$ variance of samples of $X_2$

We use the following formula for a $t$-test with unequal sample sizes and equal variance:

$$t - test = \frac{(m_1 - m_2)}{s_p}$$

where:

$$s_p = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 1}}$$

For $X_2$ we sample values from the whole chromosome except for the candidate aberrated regions. Consequently, $n_2$ is very large in comparison to $n_1$. This provides the rule:

$$n_2 \gg n_1 > 1$$

with this rule, we can approximate the $\sqrt{\cdot}$ term in the denominator of the $t$-test (equation S1), i.e.

$$\sqrt{\frac{1}{n_1} + \frac{1}{n_2}} = \sqrt{\frac{1}{n_1} + \frac{1}{\infty}} = \sqrt{\frac{1}{n_1}} = \frac{1}{\sqrt{n_1}}$$

Now the $t$-test can be rewritten to:

$$t - test = \frac{\sqrt{m_1}}{s_p(m_1 - m_2)}$$

Rewriting $s_p$:

$$s_p = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 1}}$$
Again, using the rule stated in equation (S7), \( n_2 \gg n_1, n_2 \gg 1 \), this can be approximated as follows:

\[
sp = \sqrt{\frac{(n_1 - 1)}{n_2} s_1^2 + \frac{(n_2 - 1)}{n_2} s_2^2}
\]  

(11)

Using that \( n_2 \gg 1 \), the first term disappears. The ratio in the second term equals 1. From that it follows that the pooled variance becomes:

\[
sp = s_2
\]  

(12)

Since the non aberrated region on the chromosome is very large \( s_2 \) will be nearly constant when the segment \((X_1)\) changes size. Therefore, when maximizing the \( t \)-test \( sp \) can be considered constant:

\[
t^* = \frac{\sqrt{n_1}}{sp} (m_1 - m_2) \rightarrow \sqrt{n_1} (m_1 - m_2)
\]  

(13)

Our implementation maximizes the difference between the mean of the segment and all other, not aberrated, target regions on the same chromosome, multiplied by the square root of the length of the segment. From the deduction shown here, it follows our segmentation equals optimization between two sets using the \( t \)-test.
SM3. Settings used for other tools

**XHMM:** Samples supplied to XHMM (downloaded from GitHub @ 18 June, 2015) were the same BAM files as prepared for WISExome testing. XHMM was run according to the tutorial. The quality filter was set to Q_{SOME} \geq 60, as was suggested in the tutorials. XHMM counts the read coverage per exon and employs Principal Component Analysis (PCA) to remove most of the technical variations over a set of samples. XHMM takes several heuristics into account such as exome-wide CNV rates, length distributions and the distance between target regions, all of these were set to default values. XHMM could not distinguish between training and test samples, thus allowing test samples to influence their own results and other samples through both the PCA and sample comparisons.

**CoNIFER:** CoNIFER (version 0.2.2; released September 17, 2012) was provided all samples involved in this project. Settings were kept at defaults, with the --svd variable set to 4. Seeing the results were missing many of the known positives, we also tried running with SVD values of 2 up to and including 6 to ensure we did not misinterpret the scree plot (Supplementary Figure S12), but these alternative settings did not change the results significantly. Just like XHMM, CoNIFER could not distinguish between training and test samples, thus allowing test samples to influence their own results and other samples through both the SVD and sample comparisons.

**CODEX:** Codex (GitHub, commit 3d40ac9 @ April 7, 2017) was run according to the vignette and tutorials supplied. As we have a clear group of affected individuals and controls, we used the normalize2 function which allowed us to specify the healthy control group. CODEX appears to make many calls allowing the end user to set a threshold. As their paper states, CODEX found significantly more calls than XHMM (2-fold) and CoNIFER (10-fold) in their tests, our finding appears in line with the original authors statements. As we were unable to find a suggestion for this threshold in the manual and tutorials, we decided not to apply one to influence the results as little as possible.

**CLAMMS:** We ran CLAMMS (GitHub, commit 3e19892 @ April 10, 2017) with all default settings and suggested values. Based on the readme, we applied Q_{EXACT} \geq 0 as a threshold. Results shown here were created using all reference data as reference set. We did not manually pick a reference set per sample (as suggested in the readme), as this would make results too dependent on user interaction. We tried the automated k-tree that was also described, but using the settings shown in the examples results became far worse, likely due to using a too small reference set per test sample.

For tools that provided a scoring per region, we plotted a distribution of these values in Supplementary Figure S9.