RESEARCH ARTICLE

CNVkit: Genome-Wide Copy Number Detection and Visualization from Targeted DNA Sequencing

Eric Talevich[1,2,3], **A. Hunter Shain**[1,2,3], **Thomas Botton**[1,2,3], **Boris C. Bastian**[1,2,3]*

1 Department of Dermatology, University of California, San Francisco, San Francisco, California, United States of America, **2** Department of Pathology, University of California, San Francisco, San Francisco, California, United States of America, **3** Helen Diller Family Comprehensive Cancer Center, University of California, San Francisco, San Francisco, California, United States of America

* boris.bastian@ucsf.edu

Abstract

Germline copy number variants (CNVs) and somatic copy number alterations (SCNAs) are of significant importance in syndromic conditions and cancer. Massively parallel sequencing is increasingly used to infer copy number information from variations in the read depth in sequencing data. However, this approach has limitations in the case of targeted re-sequencing, which leaves gaps in coverage between the regions chosen for enrichment and introduces biases related to the efficiency of target capture and library preparation. We present a method for copy number detection, implemented in the software package CNVkit, that uses both the targeted reads and the nonspecifically captured off-target reads to infer copy number evenly across the genome. This combination achieves both exon-level resolution in targeted regions and sufficient resolution in the larger intronic and intergenic regions to identify copy number changes. In particular, we successfully inferred copy number at equivalent to 100-kilobase resolution genome-wide from a platform targeting as few as 293 genes. After normalizing read counts to a pooled reference, we evaluated and corrected for three sources of bias that explain most of the extraneous variability in the sequencing read depth: GC content, target footprint size and spacing, and repetitive sequences. We compared the performance of CNVkit to copy number changes identified by array comparative genomic hybridization. We packaged the components of CNVkit so that it is straightforward to use and provides visualizations, detailed reporting of significant features, and export options for integration into existing analysis pipelines. CNVkit is freely available from https://github.com/etal/cnvkit.

This is a *PLoS Computational Biology* software paper.

OPEN ACCESS

Citation: Talevich E, Shain AH, Botton T, Bastian BC (2016) CNVkit: Genome-Wide Copy Number Detection and Visualization from Targeted DNA Sequencing. PLoS Comput Biol 12(4): e1004873. doi:10.1371/journal.pcbi.1004873

Received: September 9, 2015

Accepted: March 16, 2016

Published: April 21, 2016

Data Availability Statement: All CNVkit coverage data files are available from GitHub (http://github.com/etal/cnvkit-examples). Corresponding raw sequencing data for the melanoma samples were deposited in the database of Genotypes and Phenotypes (dbGaP) under accession phs000977.v1.p1, and array CGH copy number data are in the Gene Expression Omnibus (GEO) under accession GSE55150. C0902 cell line raw data from the Botton 2013 study are in the Supporting Information files of that article. The authors of both studies may be contacted at boris.bastian@ucsf.edu.

Funding: This work was supported by the National Institutes of Health [R01 CA131524, P01 CA025874 to BCB]. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing Interests: The authors have declared that no competing interests exist.

Introduction

Copy number changes are a useful diagnostic indicator for many diseases, including cancer. The gold standard for genome-wide copy number is array comparative genomic hybridization (array CGH) [1, 2]. More recently, methods have been developed to obtain copy number information from whole-genome sequencing data ([3]; reviewed by [4]). For clinical use, sequencing of genome partitions, such as the exome or a set of disease-relevant genes, is often preferred to enrich for regions of interest and sequence them at higher coverage to increase the sensitivity for calling variants [5]. Tools have been developed for copy number analysis of these datasets, as well, including CNVer [6], ExomeCNV [7], exomeCopy [8], CONTRA [9], CoNIFER [10], ExomeDepth [11], VarScan 2 [12], XHMM [13], ngCGH [14], EXCAVATOR [15], CANOES [16], PatternCNV [17], CODEX [18], and recent versions of Control-FREEC [19] and cn.MOPS [20]. However, these approaches do not use the sequencing reads from intergenic and, usually, intronic regions, limiting their potential to infer copy number across the genome.

During the target enrichment, targeted regions are captured by hybridization; however, a significant quantity of off-target DNA remains in the library, and this DNA is sequenced and represents a considerable portion of the reads. Thus, off-target reads provide a very low-coverage sequencing of the whole genome, in addition to the high-coverage sequencing obtained in targeted regions. While the off-target reads alone do not provide enough coverage to call single-nucleotide variants (SNVs) and other small variants, they can provide useful information on copy number at a larger scale, as recently demonstrated by cnvOffSeq [21] and CopywriteR [22].

We developed a computational method for analysis of copy number variants and alterations in targeted DNA sequencing data that we packaged into a software toolkit. This toolkit, called CNVkit, implements a pipeline for CNV detection that takes advantage of both on– and off-target sequencing reads and applies a series of corrections to improve accuracy in copy number calling. We compare binned read depths in on– and off-target regions and find that they provide comparable estimates of copy number, albeit at different resolutions. We evaluate several bias correction algorithms to reduce the variance among binned read counts unlikely to be driven by true copy number changes. Finally, we compare copy ratio estimates by the CNVkit method and two competing CNV callers to those of array CGH, and find that CNVkit most closely agrees with array CGH. In summary, we demonstrate that both on– and off-target reads can be combined to provide highly accurate and reliable copy ratio estimates genome-wide, maximizing the copy number information obtained from targeted sequencing.

Design and Implementation

We implemented CNVkit as a Python 2.7 software package comprising a command-line program, `cnvkit.py`, and reusable library, `cnvlib`.

Software pipeline

The input to the program is one or more DNA sequencing read alignments in BAM format [23] and the capture bait locations or a pre-built "reference" file (Fig 1). All additional data files used in the workflow, such as GC content and the location of sequence repeats, can be extracted from user-supplied genome sequences in FASTA format using scripts included with the CNVkit distribution. The workflow is not restricted to the human genome, and can be run equally well on other genomes.

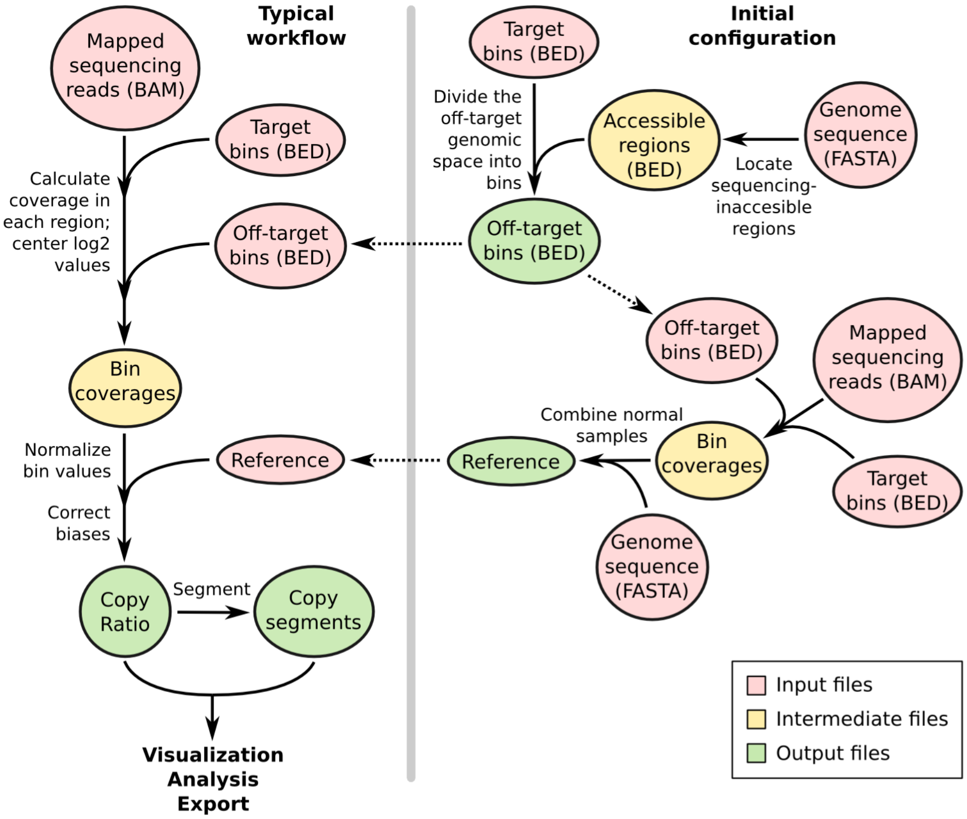

Fig 1. CNVkit workflows. The target and off-target bin BED files and reference file are constructed once for a given platform and can be used to process many samples sequenced on the same platform, as shown in the workflow on the left. Steps to construct the off-target bins are shown at the top-right, and construction of the reference is shown at the lower-right.

doi:10.1371/journal.pcbi.1004873.g001

CNVkit uses both the on-target reads and the nonspecifically captured off-target reads to calculate $\log_2$ copy ratios across the genome for each sample. Briefly, off-target bins are assigned from the genomic positions between targeted regions, with the average off-target bin size being much larger than the average on-target bin to match their read counts (Table 1). Both the on– and off-target locations are then separately used to calculate the mean read depth within each interval. The on– and off-target read depths are then combined, normalized to a reference derived from control samples, corrected for several systematic biases to result in a final table of $\log_2$ copy ratios. A built-in segmentation algorithm can be run on the $\log_2$ ratio values to infer discrete copy number segments. The $\log_2$ ratios and segments can then be used for visualization and further analyses supported by CNVkit, exported to other formats, and used with third-party software.

These steps are implemented entirely in CNVkit so that the complete workflow can be performed in a reasonable amount of time on a commodity workstation or laptop. The most computationally demanding step, read depth calculation, takes on the order of 20 minutes for an exome at 100-fold coverage or 2 minutes for a 293-gene target panel at 500-fold coverage using a single 3.7GHz CPU and a solid-state drive. Initial calculation of regional GC RepeatMasker content from the human genome takes about one minute, and all other steps complete in a few seconds at most. The implementation is designed to be memory-efficient, so that many samples can safely be run in parallel on a single machine.

Table 1. Binning statistics.

	TR		EX		C0902	
Statistic	**on-target**	**off-target**	**on-target**	**off-target**	**on-target**	**off-target**
Number of bins	8,216	19,434	301,249	55012	8,662	19,402
Total bin footprint (bp)	1,791,315	2,837,786,301	70,364,091	2,468,075,581	1,867,888	2,837,005,032
Mean bin size (bp)	218.0	146,021.7	233.6	44864.3	215.6	146,222.3
Min. bin size (bp)	36	10,012	114	6,000	42	10,089
1st quartile bin size (bp)	183	148,196	197	11,304	181	148,108
Median bin size (bp)	203	149,819	227	28,358	200	149,814.5
3rd quartile bin size (bp)	259	151,062	268	86,767	257	151,070
Max. bin size (bp)	397	223,781	399	134,972	398	224,678

The bins for the exome panel (EX) cover a slightly smaller total genomic footprint than the targeted panels (TR, C0902) because most introns are smaller than the minimum size allowed for off-target bins, and thus discarded from the exome bins, while the off-target bins in the targeted panels span both the introns and exons of non-targeted genes.

doi:10.1371/journal.pcbi.1004873.t001

Calculation of off-target intervals

Genomic intervals for counting off-target reads are initially calculated from the genomic positions of the targeted intervals. The CNVkit `antitarget` command accepts a list of targeted regions, in Browser Extensible Data (BED) or GATK/Picard interval list format, and divides the off-target regions between each target into large bins, typically on the order of 100 kilobases. As an optional input, separate lists of the sequencing-accessible chromosomal regions and low-mappability regions can be used to exclude telomeres, centromeres and other sequencing-inaccessible or unmappable repetitive regions from the off-target intervals when creating the off-target bins.

Each contiguous off-target region is divided into equal-sized bins such that the average bin size within the region is as close as possible to the size specified by the user. The user can select an appropriate off-target bin size by calculating the product of the average target region size and the fold-enrichment of sequencing reads in targeted regions, such that roughly the same number of reads are mapped to on– and off-target bins on average. In an effort to maximize the number of bins, CNVkit will deviate from the user-specified bin size to fit bins into small regions, such as introns, that are restricted in size. The user can also specify a lower limit on bin size to avoid evaluating very small off-target regions, where it is expected that too few reads would be captured to give a reliable estimate of copy number. Once a satisfactory set of off-target bins have been generated and saved as a BED file, the same BED file can be reused with CNVkit for copy number analysis of other samples prepared with the same library preparation protocol and sequenced on the same platform.

Estimation of copy number by read depth

The CNVkit `coverage` command computes the $\log_2$ mean read depth in each bin for a sample using an alignment of sequencing reads in BAM format and the positions of the on– or off-target bins in BED or interval list format. For each bin the read depths at each base pair in the bin are calculated and summed using pysam, a Python interface to samtools [23], and then divided by the size of the bin. The output is a table of the average read depths in each of the given bins $\log_2$-transformed and centered to the median read depth of all autosomes.

To produce the input BAM file, we recommend that an aligner such as BWA-MEM [24] be used with the option to mark secondary mappings of reads, and that PCR duplicates be flagged.

Construction of a copy number reference

The `reference` command estimates the expected read depth of each on– and off-target bin across a panel of control or comparison samples to produce a reference copy-number profile that can then be used to correct other test samples. At each genomic bin, the read depths in each of the given control samples are extracted. Read-depth bias corrections (see below) are performed on each of the control samples. In each bin, a weighted average of the $\log_2$ read depths among the control samples is calculated to indicate bins that systematically have higher or lower coverage, and the spread or statistical dispersion of log2 read depths indicates bins that have erratic coverage so that they can be de-emphasized at the segmentation step. A single paired control sample can also be used, or, in absence of any control samples, a "generic" reference can be constructed with a $\log_2$ read depth and spread of 0 assigned to all bins. In all cases a "male reference" can be specified in which the expected read depth of X chromosome bins is half that of the autosomes.

Additional information can be associated with each bin for later use in bias correction and segmentation. If the user provides a FASTA file of the reference genome at this step, the GC content and repeat-masked fraction of each binned corresponding genomic region are calculated. CNVkit calculates the fraction of each bin that is masked and records this fraction in an additional column in the reference file, along with GC, average log2 read depth, and spread.

As with the target and off-target BED files, once a satisfactory reference file has been generated, it can be reused with CNVkit for copy number analysis of other similar samples sequenced with the same platform and protocol.

Normalization of test samples to the reference

The `fix` command combines a single sample's on– and off-target binned read depths, removes bins failing predefined criteria, corrects for systematic biases in bin coverage (see below), subtracts the reference $\log_2$ read depths, and finally median-centers the corrected copy ratios.

Each bin is then assigned a weight to be used in segmentation and plotting. Each bin's weight is calculated according to bin size, difference from the global median coverage (if at least one control sample is provided), and the spread of normalized coverages in the control pool (if more than one control sample is provided). Finally, the overall variability of bin log2 ratio values is compared between on- and off-target bins, and the more variable of the two sets is downweighted in proportion.

Correction of coverage biases

Read depth alone is an insufficient proxy for copy number because of systematic biases in coverage introduced during library preparation and sequencing. For example, read depth is affected by GC content, sequence complexity and the sizes of individual targeted intervals [15, 19, 25]. To account for each of these potential biases in depth of read coverage, CNVkit uses a rolling median technique to recenter each on– or off-target bin with other bins of similar GC content, repetitiveness, target size or distance from other targets, independently of genomic location.

Systematic coverage biases may be largely removed simply by normalization to a reference of one or more representative normal samples, and subsequent corrections for these biases then have relatively little effect. However, even after normalization to a pooled reference, biases in coverage typically do persist in an individual sample and must still be removed.

Genomic GC content. DNA regions with extreme GC content are less accessible to hybridization and amenable to amplification during library preparation [26, 27]. The degree of GC bias can vary between samples due to differences such as the quality of each sample's DNA or efficiency of hybridization between library preparations. To remove this bias, CNVkit applies a rolling median correction (see below) to GC values on both the target and off-target bins, independently.

Sequence repeats. Repetitive sequences in the genome can complicate read-depth calculations, as these regions often show high variability in coverage from sample to sample [15]. This variability may be due to differences in the efficiency of the blocking step during library preparation (e.g. differences in the quantity of Cot-1 during blocking). The presence of sequence repeats serves as an indicator for regions prone to these biases.

In the reference genome sequences provided by the UCSC Genome Bioinformatics Site (http://genome.ucsc.edu/) and others, repetitive regions are masked out by RepeatMasker (http://repeatmasker.org). CNVkit calculates the proportion of each bin that is masked, similar to the method used in XHMM [13], and uses this information for bias correction. The CNVkit implementation applies the RepeatMasker correction to only the off-target bins. For most custom bait libraries, the on-target bins are much smaller, and usually are exonic, and therefore generally have no overlap with repeats. For those on-target bins that were identified as containing repeats (e.g. ∼7% in our custom target panel, see Results), we found them mostly entirely covered by the repeat, leaving very few intermediate points to infer a continuous trend for correction by the rolling median.

Target density. We observed two distortions to read depth consistently occurring at the edges of each targeted interval (Fig 2): The "shoulders" of each interval showed reduced read depth due to incomplete sequence match to the bait, creating a negative bias in the observed read depth inside the interval near each edge; this effect was greatest for short intervals (Fig 2A). Some off-target capture also occurred in the "flanks" of the baited interval due to the same mechanism. Where targets are closely spaced or adjacent, this flanking read depth may overlap with a neighboring target, creating a positive bias in its observed read depth (Fig 2B). We accounted for the negative bias at the interval "shoulders" and the positive bias in the interval "flank" regions in a single model that describes the "density" of targets around a bin.

CNVkit's bias correction procedure needs only a monotonic function of the actual read depth bias, rather than the magnitude of the bias itself. For simplicity, we modeled the density

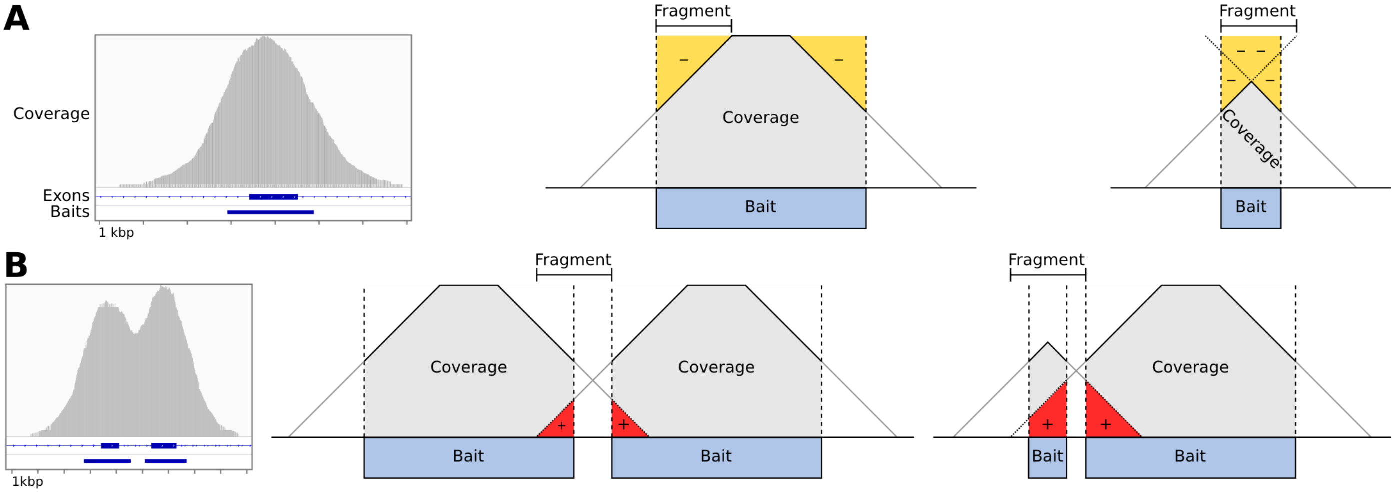

Fig 2. Baited region size and spacing affect read depth systematically. A: Example of typical coverage observed at a targeted exon, as viewed in IGV, and simplified geometric models of the negative coverage biases (yellow) that can occur as a function of the relative sizes of sequence fragments and the baited region. B: Coverage observed at two neighboring targeted exons, and models of the positive coverage biases (red) that can occur where intervals are separated by less than half the insert size of sequence fragments.

doi:10.1371/journal.pcbi.1004873.g002

biases as a linear decrease in read depth from inside the baited region to the same distance outside, calculated from the start and end positions of a bin and its immediate neighbors (Fig 2A and 2B). In the common case of no other targets within the window surrounding the given target, there is a one-to-one correspondence of the density value to target size. Thus, the density bias correction also accounts for the bias due to target size that has been described by others [15].

While the density bias can be significantly reduced by normalizing each sample to a reference, it may vary between samples due to differences in the insert sizes of sequence fragments introduced during the step of DNA fragmentation of the library preparation, and thus should still be accounted for even if a matched normal comparison exists. Density bias being related to the capture, CNVkit only applies this correction to the on-target bins.

Computational correction of biases. All of the information needed to calculate the biases at each bin is stored in the reference file. For each of the biases (GC content, repeat-masked fraction, target density), the bias value is calculated for each bin. Next, bins are sorted by bias value. A rolling median is then calculated across the bin $\log_2$ ratios ordered by bias value to obtain a midpoint $\log_2$ ratio value representing the expected bias for each bin. Finally, this value is subtracted from the original bin $\log_2$ ratio for the given sample to offset the observed bias. We also evaluated local regression (LOWESS) [28] and a Kaiser window function [29] in place of the rolling median to estimate the trend due to bias; all three functions produced similar fits on sample data, and we chose rolling median as the default for its simplicity and robustness.

Segmentation and calling absolute copy number

The sample's corrected bin-level copy ratio estimates can be segmented into discrete copy-number regions using the `segment` command. The bin log2 ratio values are first optionally filtered for outliers, defined as a fixed multiple of the 95th quartile in a rolling window, similar to BIC-seq [30]. The default segmentation algorithm used is circular binary segmentation (CBS) [31], via the R package PSCBS [32]. Alternatively, the HaarSeg algorithm [33] or Fused Lasso [34] can be used in place of CBS. In either case, the segmentation output is in a BED-like tabular format similar to that used for bin-level copy ratio tables.

Calling absolute copy number is implemented separately from segmentation. The `rescale` command, an optional step, can adjust a tumor sample's $\log_2$ ratios given an estimate of normal-cell contamination (separately derived from cell count or DNA content, see S2 Text), and can re-center the $\log2_2$ ratios by median, mode, or other measures of central tendency. The `call` command rounds the $\log_2$ ratios to the nearest integer absolute copy number given the normal ploidy of each chromosome, or directly maps segment $\log_2$ ratios to absolute copy number states given a set of numeric thresholds.

Data summarization, reporting and visualization

CNVkit generates several kinds of plots using the software libraries Biopython [35], Reportlab (http://www.reportlab.com/opensource/) and matplotlib (http://matplotlib.org):

- a "heatmap" of segmented results from multiple samples;
- a single-sample "scatter" plot of bin-level coverages with overlaid segments, either genome-wide or in selected chromosomal regions, optionally with single-nucleotide variant allele frequencies from a separately called Variant Call Format (VCF) file shown to indicate regions of loss of heterozygosity;

- a “diagram” of each chromosome drawn with bin-level copy ratios, segments, or both, labeled with the genes covered by copy number variants.

Copy number features can also be summarized as tabular text reports: Gene-level copy number information can be extracted with the `gainloss` command, and segmentation breakpoints that fall within a gene (possibly indicating translocation) with the `breaks` command. Statistics on the residual deviations of bin-level copy ratios from the segmentation calls are calculated per-sample with the `metrics` command, and per-segment with `segmetrics`.

Integration and compatibility with other software

To ease integration into a variety of workflows and pipelines, CNVkit can convert between its native, BED-like file format and formats supported by other software. In particular, the standard SEG format used by GenePattern [36] and Integrative Genomics Viewer [37], and others is supported for both import and export, while standard BED, VCF and Clustered Data Table (CDT) and the native formats of Java TreeView [38] and Nexus Copy Number (BioDiscovery Inc.) are only exported. The per-target coverages reported by the CalculateHsMetrics script in Picard tools (http://picard.sourceforge.net/) can be imported as an alternative to CNVkit’s `coverage` command. Import and export compatibility with the tumor heterogeneity analysis program THetA2 [39] is implemented to allow fully automated estimation of tumor cell fraction and subclones.

Application wrappers are available for Galaxy [40], DNAnexus, and Docker. CNVkit is also included in the best-practices sequencing analysis pipeline bcbio-nextgen (https://bcbio-nextgen.readthedocs.org/en/latest/) and can be used with the ensemble structural-variant caller MetaSV [41].

Results

We evaluated our method on DNA sequencing data from targeted sequencing of the melanoma cell line C0902 [42] and two sets of samples, referred to here as “TR” and “EX”, derived from a recent study of advanced melanomas [43]:

- Targeted sequencing (“TR”) of 82 samples, paired tumor and normal tissue from archived microdissected FFPE of 41 melanoma patients, sequenced with a custom 293-gene target capture protocol.
- Exome sequencing (“EX”) of 20 samples, paired fresh frozen tumor and matching blood samples from 10 melanoma patients, sequenced with a whole-exome capture protocol.

Sequencing methods are described in S1 Text.

For each panel of targets, on– and off-target genomic regions are each partitioned into bins (Table 1) in which unique reads are counted in the initial step of copy number estimation. The read counts and percentages in on– and off-target regions for each of these samples are shown in S1 Table.

Correction of systematic biases in read depth improves copy ratio estimates

While normalization to a reference reduces the coverage biases attributable to GC content, repetitive sequence, and target density introduced by library preparation and sequencing, the extent of each of these systematic biases varies from sample to sample (Fig 3), requiring additional correction measures of the residual biases.

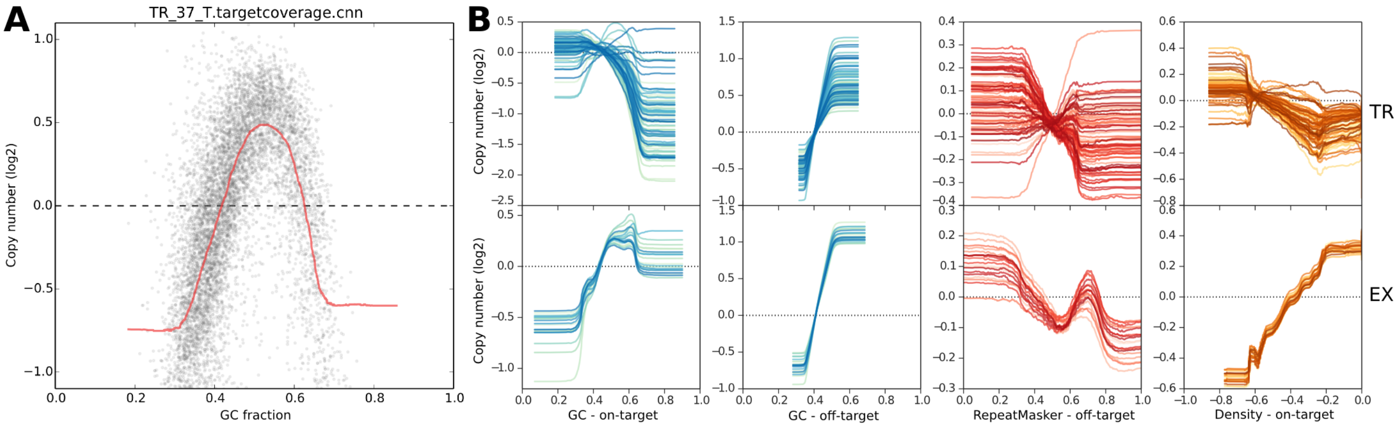

Fig 3. Bin read depths are systematically biased by GC content and other factors. A: GC coverage bias follows a unimodal distribution in sample TR_37_T. Target bins are sorted according to bin GC fraction (x-axis), and the uncorrected, median-centered $\log_2$ bin read depths are plotted (y-axis). A rolling median of the bin $\log_2$ read depths in order of GC value is drawn in red, showing a systematic deviation from 0 in the selected sample. B: Trendlines summarize each bias type in each sample. TR and EX samples are shown in the top and bottom rows, respectively. Columns show biases due to GC content in target bins and off-target bins, repeat content in off-target bins, and density bias in target bins.

doi:10.1371/journal.pcbi.1004873.g003

We evaluated the effect of each of our bias corrections by comparing the final segmented copy number data, separately determined with all corrections enabled, to the bin-level read depths or $\log_2$ ratios for on– and off-target bins at each processing step (Fig 4). For each sample in the TR and EX cohorts, we used CNVkit to perform each of the corrections described above sequentially to estimate bin-level $\log_2$ ratios. First, we subtracted the uncorrected, median-centered $\log_2$ read depth of each on– and off-target bin from the corresponding $\log_2$ copy ratio values of the final segmentation to obtain the deviations of each bin from our final estimate of true $\log_2$ copy ratio. We repeated this calculation at each of the subsequent steps of bias correction: (i) after GC bias correction; (ii) after the density and repeat corrections; and (iii) after normalizing to a pooled reference.

In these results, the deviation values decreased monotonically across all steps, indicating that each step of corrections reduces random deviations from the true copy number signal/value. The spread of deviation values also decreased overall, indicating that the improvements are seen consistently and are reliable; even outlier data points (representing samples with poor overall sequencing quality) were consistently improved. The greatest improvements were seen from GC bias correction and normalization to the pooled reference.

While each step reduced deviations of off-target bins similarly between the two cohorts, the on-target bins in the EX cohort appear to exhibit more variation in read depths that is independent of GC and targeting density, but consistently removed by refererence normalization. This difference between the two cohorts may be due to differences in the target capture kits' probe design, the diversity of genes captured, and the type of samples sequenced. In particular, the Nimblegen custom panel used for TR primarily captures average-sized genes that are amenable to hybridization, while the Agilent exome panel used for EX captures nearly all protein-coding genes.

We also found that the deviation of the off-target bins was inversely related to the off-target bin size, or equivalently, directly related of the number of reads captured in each off-target bin. Thus, by choosing the off-target bin size to match the average read counts for on-target bins, we ensured that the deviations or random error in the read counts per bin was similar between on– and off-target bins.

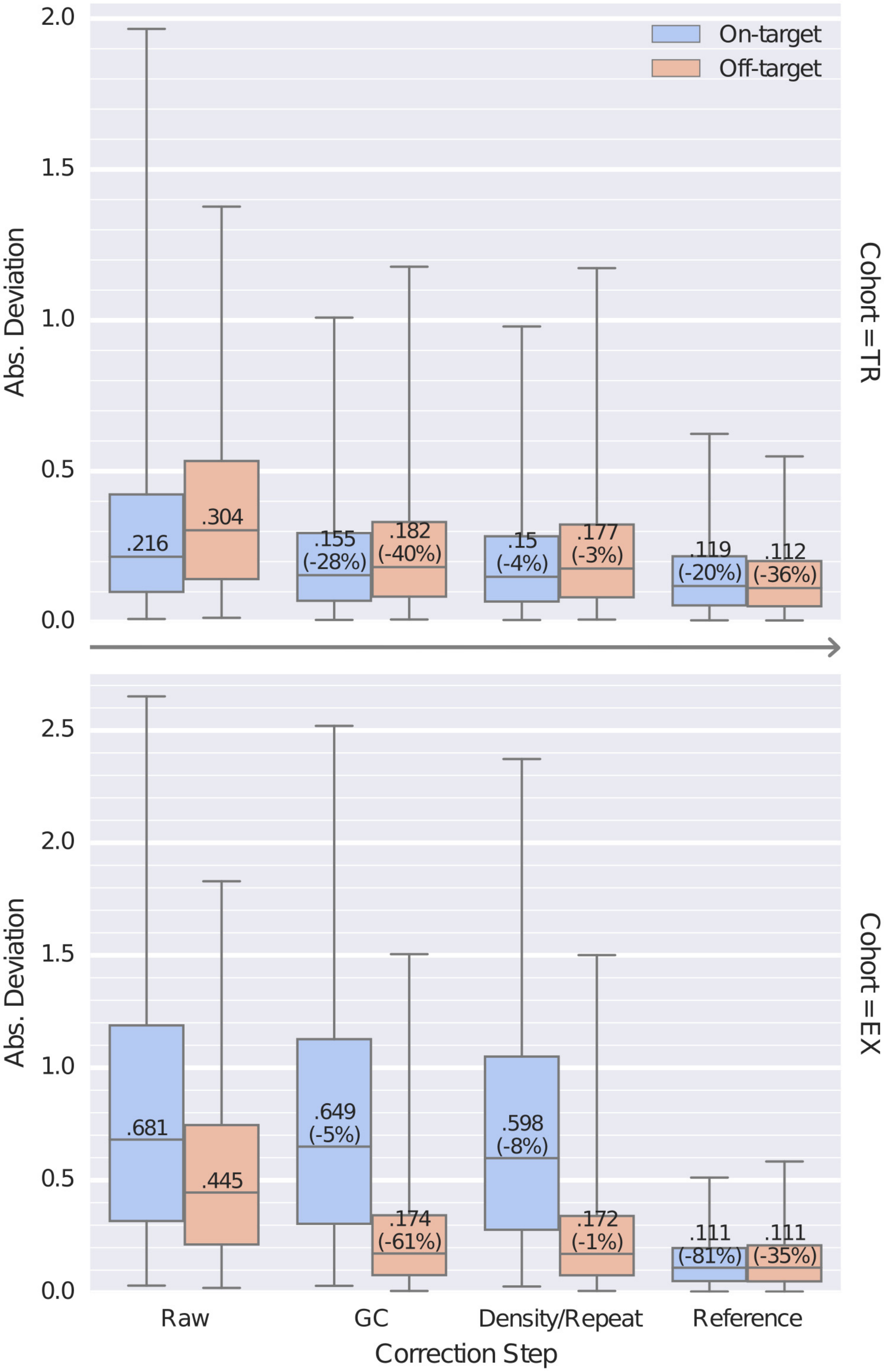

Fig 4. Bias corrections reduce the extraneous variation in bin read depths. Distributions of the absolute deviation of on– and off-target bins from the final, segmented copy ratio estimates are shown as box plots at each step of bias correction for all samples in the TR and EX sequencing cohorts. At each step, for on- and off-target bins separately, boxes show the median and interquartile range of absolute deviations and whiskers show the 95% range. Steps shown are the initial median-centered $\log_2$ read depth ("Raw"), correction of GC bias ("GC"), correction of on-target density and off-target repeat biases ("Density/Repeat"), and normalization to a pooled reference ("Reference").

doi:10.1371/journal.pcbi.1004873.g004

Validation by array CGH and FISH assays

We validated the segmented copy ratio estimates by CNVkit with respect to two widely used methods for copy number measurement, array CGH (Agilent 4x180K) and fluorescence in situ hybridization (FISH) (see S1 Text). For this validation we used the C0902 cell line, derived from a melanoma.

We compared CNVkit and array CGH copy ratios across the whole genome (Fig 5A). Segmentation by CBS yielded 70 segments from the CNVkit bins and 146 segments from the array CGH probes. The median absolute deviation (MAD) of the residual bin– or probe-level $\log_2$ ratio values from the corresponding array CGH segment means was 0.2002 by array CGH and 0.1531 by CNVkit.

One multi-copy gene-level deletion was detected by array CGH but not by CNVkit. In the gene CEBPA, a 27.5-kilobase loss with a $\log_2$ copy ratio of -1.5691 is detected by 8 array CGH probes, but the corresponding CNVkit bins showed neutral copy number. These bins cover sequence regions with very high GC content (73–82%) and the CNVkit reference indicated an expected read depth significantly below the genome-wide average, which may have masked any true copy number loss at this locus in the sequencing data. Further comparison of the CNV calls made by array CGH and CNVkit is presented in the next section and in S1 Text.

Next, we used FISH to determine the absolute copy number at loci harboring cancer-relevant genes: ALK, ROS1, MET, BRAF and RET.

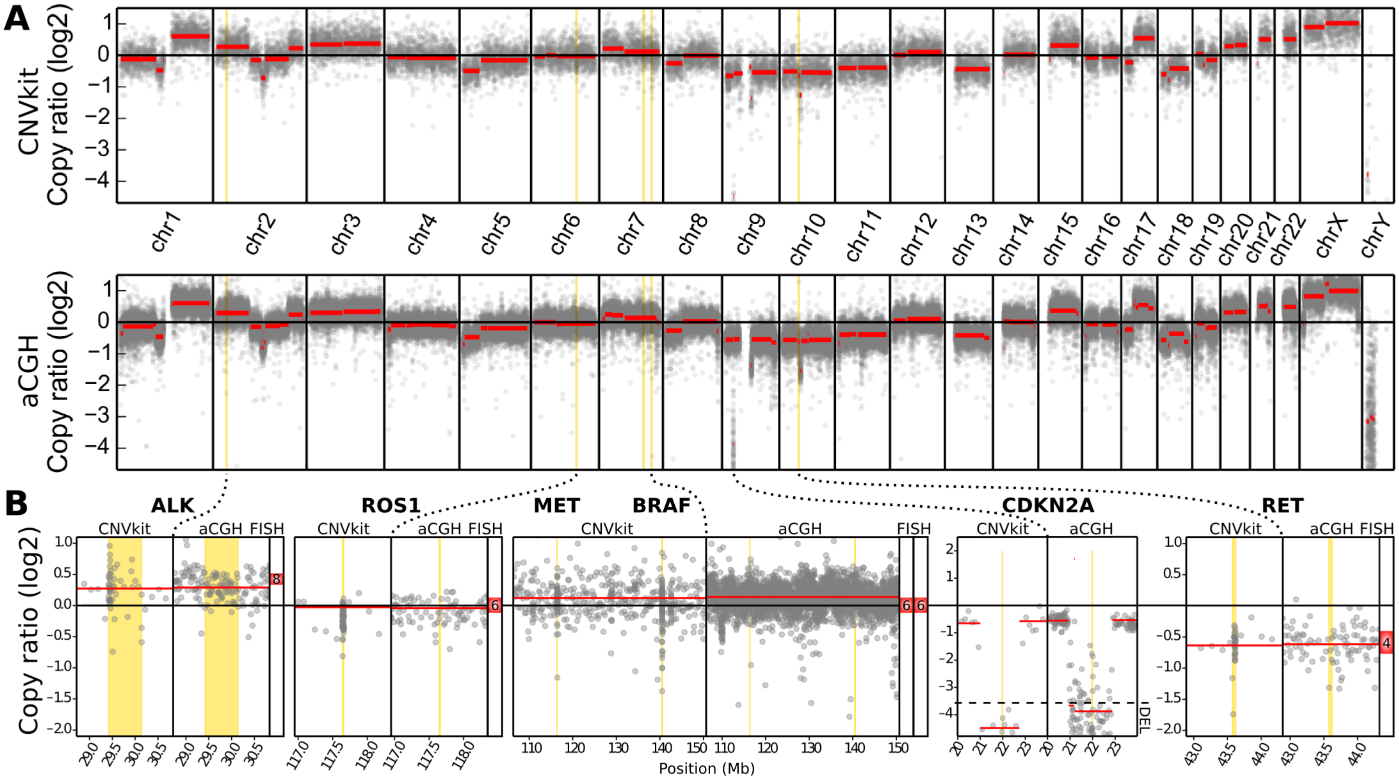

Fig 5. CNVkit copy ratios agree with experimental results array CGH and FISH on cell line DNA. A: Whole-genome profiles of $\log_2$ copy ratio by CNVkit (top) and array CGH (bottom) are shown. B: Genes additionally assayed by FISH are labeled with the detected absolute copy number. At CDKN2A, $\log_2$ ratios below the marked level of -3.58 indicate the site is entirely deleted in the majority of cells.

doi:10.1371/journal.pcbi.1004873.g005

We compared the $\log_2$ ratios obtained by both CNVkit and array CGH to the average signal counts per nucleus obtained by FISH. We transferred the average FISH signal counts into $\log_2$ copy ratios, by calculating the difference between the $\log_2$ of their average nuclear signal counts and the $\log_2$ of the cell's ploidy, which we determined to be 6n. In all five of the genes assayed by FISH, the copy ratio inferred by CNVkit is close to the average value observed by FISH (Fig 5B).

Comparison to related software

CNVkit is the first CNV caller to automatically combine copy number information from both on– and off-target regions. These two sources of copy number information have been separately considered in other methods and their software implementations. In particular, CONTRA [9] implemented a pipeline for inferring copy number from targeted regions alone, and CopywriteR [22] recently demonstrated that copy number information can be obtained from off-target reads alone, but neither attempted to combine the off-target and on-target information. Like CNVkit, CONTRA and CopywriteR both use the CBS algorithm to perform segmentation, and both report segment means without requiring an integer copy number value—a feature essential for reporting CNVs in heterogeneous samples. We therefore selected CONTRA and CopywriteR for evaluation alongside CNVkit on the same targeted and whole-exome sequencing datasets presented earlier in this text.

We performed array CGH on each sample in the TR cohort using the Agilent 180K array, and on the EX cohort using the Agilent 1-million-probe array. We used the GenePattern server [36] to segment the array CGH $\log_2$ ratio values by CBS. The analysis pipelines for CNVkit version 0.7.6, CopywriteR version 1.99.3 and CONTRA version 2.0.6 were run with default settings (see S1 Text). Each of the methods was evaluated using all of the available approaches for constructing a reference: All normal samples pooled (supported by CNVkit and CONTRA), matched tumor-normal pairs (all three methods), and tumor-only calling with no normal reference (CNVkit and CopywriteR). The CNVkit pipeline completed the fastest in all cases, while CopywriteR and CONTRA generally required about 2–4 times as long as CNVkit.

We compared the CNV calls from each program to those obtained by array CGH. Our primary interest in this evaluation was to see how accurately each method estimates copy ratio at targeted genes, as the inclusion of these genes in a target panel implies that they are the genomic regions of the most interest. We took the differences in segmented $\log_2$ ratio estimates by array CGH and CNVkit, CONTRA or CopywriteR at each of the targeted genes, and plotted the distributions of these values for comparison (Fig 6). We also calculated the median, 2.5-percentile and 97.5-percentile of each of these distributions to identify the prediction interval (PI) in which 95% of estimates by each method typically deviate from that of array CGH (S1 Table).

With CNVkit, the best estimates were consistently obtained using a pooled reference, then by a "generic" reference, while reference-free calling remained competitive in all cohorts. In the TR cohort CNVkit performed best overall, though when restricted to reference-free calling CNVkit and CopywriteR performed similarly (PI = 0.464 and 0.462, respectively); it is striking to note that in this cohort CopywriteR performed better with no reference than with a single matched normal reference. In the EX cohort CNVkit and CopywriteR achieve reference-free performance (PI = 0.36 and 0.382), and improved by a similar degree using a reference (CNVkit pooled PI = 0.287, CopywriteR paired PI = 0.27). CONTRA did not produce better results than CNVkit or CopywriteR under any conditions, and in the TR cohort and cell line, pooling the reference appeared to exacerbate the inconsistencies apparent in the paired normals.

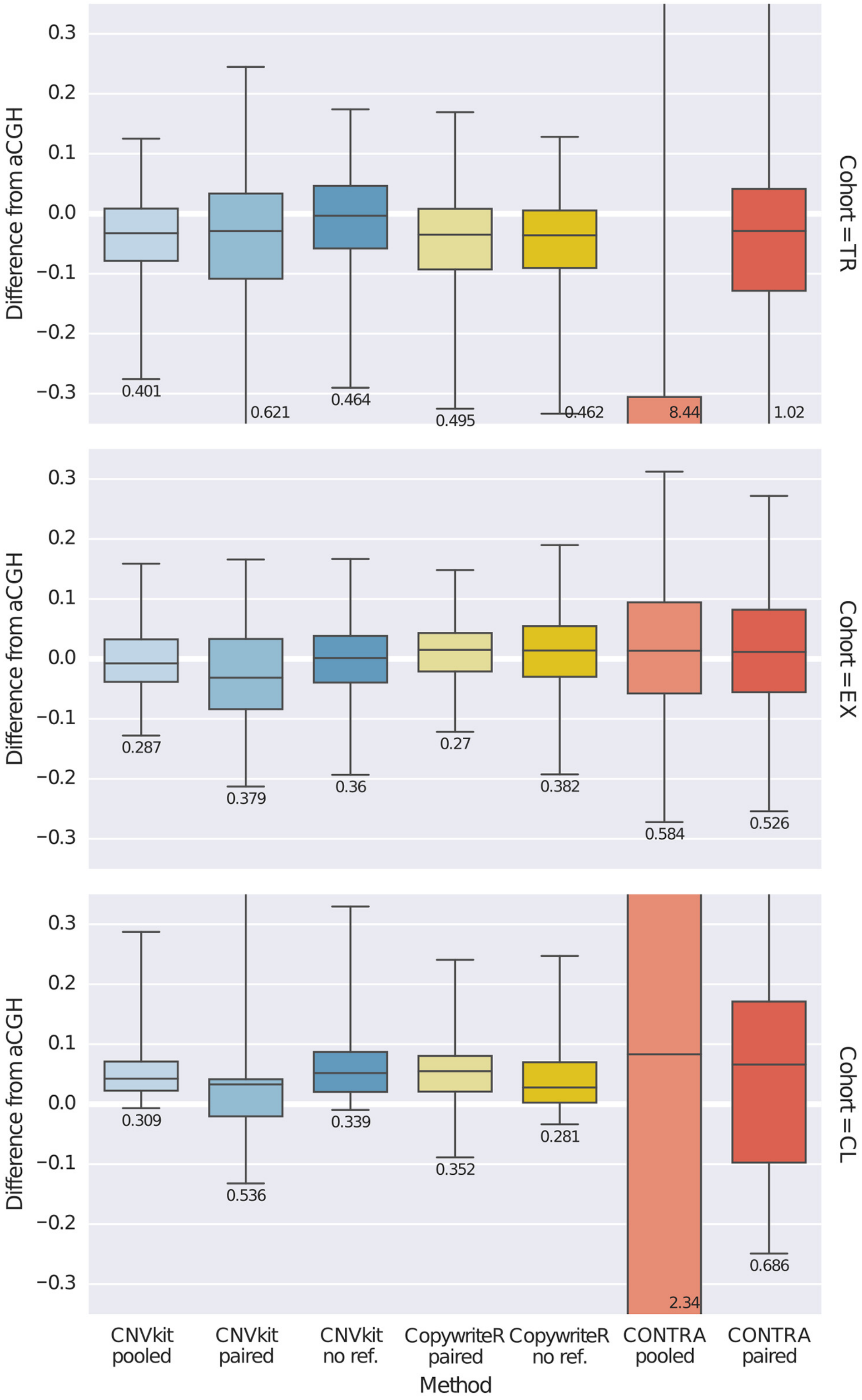

Fig 6. Comparison of CNVkit and other methods to array CGH. Log$_2$ ratio estimates by CNVkit, CONTRA and CopywriteR were compared to those by array CGH at each of the targeted genes in the TR and EX cohorts as well as the C0902 cell line sample (CL). The distribution of differences of segmented log$_2$ ratio estimates by each caller from that of array CGH at each targeted gene is shown as a box plot, where each box shows the median and interquartile range of absolute deviations, whiskers show the 95% range, and the

magnitide of the 95% range (prediction interval) is printed under the box plot. Columns are CNV callers, and rows are the TR and EX cohorts and C0902 sample on which the callers were evaluated.

doi:10.1371/journal.pcbi.1004873.g006

We also investigated genome-wide CNV calling of each method, quantifying performance in terms of precision (specificity) and recall (sensitivity). For the C0902 cell line we derived absolute integer copy numbers from the segmentation obtained by array CGH, CNVkit, CopywriteR, and CONTRA. Treating the array CGH calls as the truth set, we then compared the deletions and duplications from each caller at each copy number state between each caller and array CGH using BEDtools [44], using calls with at least 50% overlap as matches, and calculated the precision and recall for gains and losses of at least one copy and at least two copies (Fig 7). To evaluate performance on larger and smaller CNVs separately, we split the array CGH calls into subsets with CNV sizes above and below 5 megabases, the median size of CNV calls by array CGH, and recalculated precision and recall within each subset. As a check on these results, we also calculated precision and recall across each basepair in all CNVs in lieu of the 50% overlap criterion. As with the gene-level analysis presented above, under most of these metrics CNVkit appears to be competitive with or superior to the other methods.

This evaluation merely considers how well the copy number estimates by several callers agree with array CGH, and ignores key advantages that CNVkit offers—i.e. the ability to efficiently infer copy number from both on– and off-target genomic regions simultaneously, and CNVkit's extreme flexibility in composing and summarizing the analyses. Nonetheless, CNVkit consistently performs at least as well as, and in some cases much better than, similar software under a range of conditions while maximally extracting copy number resolution from deep sequencing data.

Availability and Future Directions

CNVkit source code is freely available from https://github.com/etal/cnvkit under the Apache License 2.0 (http://www.apache.org/licenses/LICENSE-2.0). Documentation is available at http://cnvkit.readthedocs.org/ and as S2 Text. Instructions and data files for recreating the analyses presented here are available at http://github.com/etal/cnvkit-examples.

CNVkit provides robust and efficient implementations of methods to improve estimates of copy number from high-throughput sequencing data, making use of both on– and off-target reads from hybrid captures. The flexible design also allows CNVkit to be readily adapted to different sequencing platforms such as Ion Torrent systems (Thermo Fisher Scientific Inc.), and to integrate well into existing analysis pipelines.

The software library underlying CNVkit serves as a basis for developing and benchmarking a variety of approaches to call, analyze and visualize copy number, not unlike the GenomicRanges framework in Bioconductor [45]. The library's modular design accommodates multiple methods for copy ratio normalization, bias correction and segmentation, and can easily incorporate new methods at any point in the workflow. In particular, we are exploring additional normalization and segmentation approaches within CNVkit to better support whole-genome sequencing and targeted amplicon capture, in which off-target reads are not available to improve copy number estimates. Another current avenue of development is using single-nucleotide polymorphism allele frequencies to assign allele-specific copy number, detect copy-number-neutral loss of heterozygosity, and investigate the structure of tumor heterogeneity in terms of absolute copy number and ploidy in each subclonal cell population.

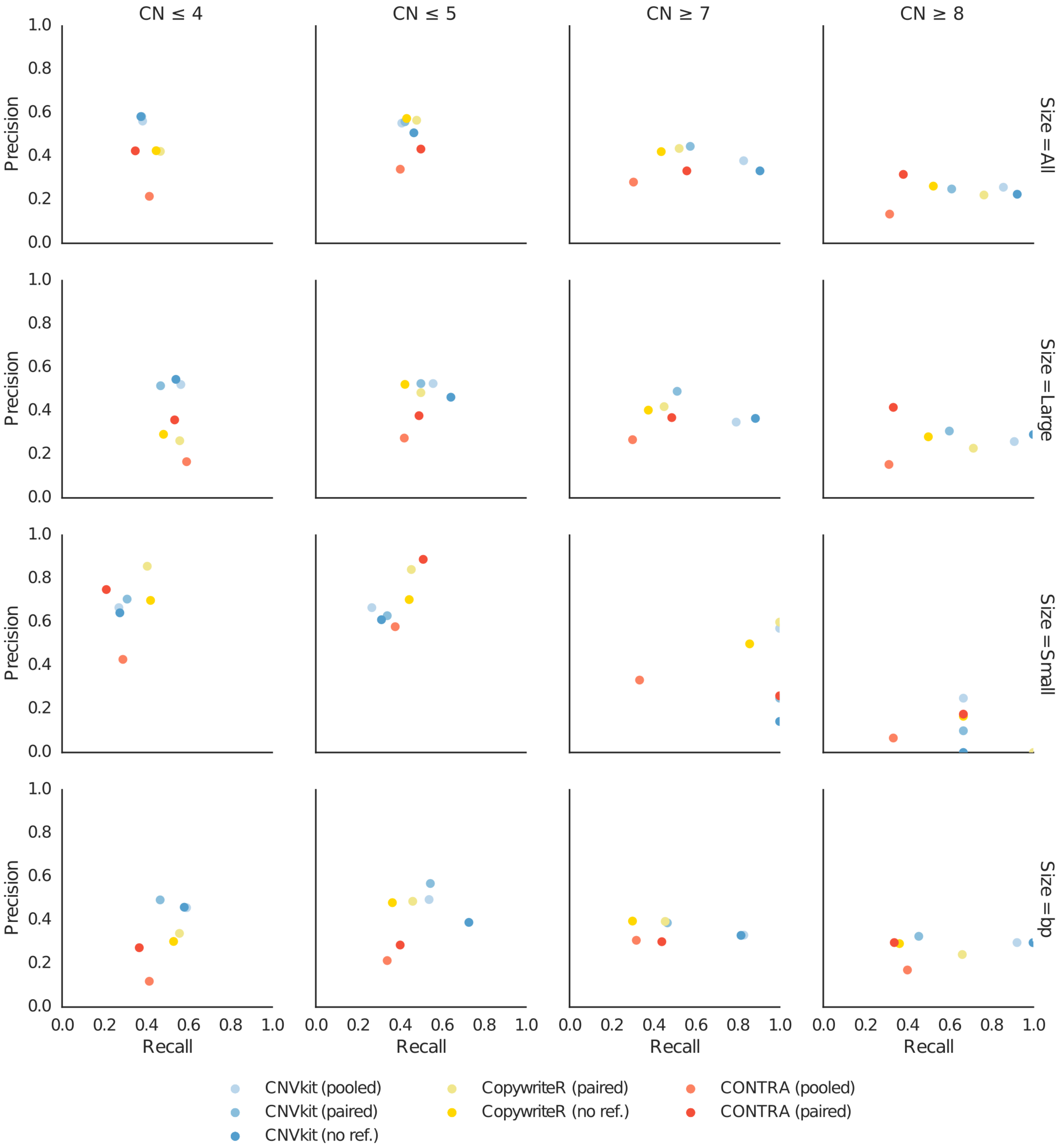

Fig 7. Precion and recall of absolute copy number calls. CNV calls obtained using each sequencing-based method are compared to those determined by array CGH to calculate precision and recall under several criteria for the C0902 cell line sample. Columns show detection of each copy number level versus the neutral hexaploid state. Rows show criteria for comparison: all CNVs, CNVs larger than 5 MB, CNVs smaller than 5MB, all CNV basepairs. Each subplot shows the calculated precision and recall of CNVkit, CopywriteR and CONTRA with each supported reference.

doi:10.1371/journal.pcbi.1004873.g007

Supporting Information

S1 Table. Sequencing metrics. Read counts, average coverage and hybridization capture metrics obtained for each sample in the TR and EX cohorts using Picard CalculateHsMetrics.
(TSV)

S1 Text. Supporting methods. Descriptions of the experimental procedures and materials used in this study.
(PDF)

S2 Text. Documentation. An up-to-date copy is maintained online at http://cnvkit.readthedocs.org/.
(PDF)

Acknowledgments

We thank Henrik Bengtsson for helpful suggestions on the use of PSCBS, and members of the Bastian lab for comments on the manuscript and feedback during the development of CNVkit.

Author Contributions

Conceived and designed the experiments: ET AHS TB BCB. Performed the experiments: ET TB. Analyzed the data: ET AHS TB. Contributed reagents/materials/analysis tools: ET AHS TB BCB. Wrote the paper: ET AHS TB BCB.

References

1. Pinkel D, Segraves R, Sudar D, Clark S, Poole I, Kowbel D, et al. High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. Nature Genetics. 1998 Oct; 20(2):207–11. doi: 10.1038/2524
2. Pinkel D, Albertson DG. Array comparative genomic hybridization and its applications in cancer. Nature Genetics. 2005 Jun; 37 Suppl(May):S11–7. doi: 10.1038/ng1569 PMID: 15920524
3. Yoon S, Xuan Z, Makarov V, Ye K, Sebat J. Sensitive and accurate detection of copy number variants using read depth of coverage. Genome Research. 2009 Sep; 19(9):1586–92. doi: 10.1101/gr.092981.109 PMID: 19657104
4. Zhao M, Wang Q, Wang Q, Jia P, Zhao Z. Computational tools for copy number variation (CNV) detection using next-generation sequencing data: features and perspectives. BMC Bioinformatics. 2013 Jan; 14 Suppl 1(Suppl 11):S1. doi: 10.1186/1471-2105-14-S11-S1
5. Dahl F, Stenberg J, Fredriksson S, Welch K, Zhang M, Nilsson M, et al. Multigene amplification and massively parallel sequencing for cancer mutation discovery. Proceedings of the National Academy of Sciences of the United States of America. 2007 May; 104(22):9387–92. doi: 10.1073/pnas.0702165104 PMID: 17517648
6. Medvedev P, Fiume M, Dzamba M, Smith T, Brudno M. Detecting copy number variation with mated short reads. Genome Research. 2010 Nov; 20(11):1613–22. doi: 10.1101/gr.106344.110 PMID: 20805290
7. Sathirapongsasuti JF, Lee H, Horst BAJ, Brunner G, Cochran AJ, Binder S, et al. Exome sequencing-based copy-number variation and loss of heterozygosity detection: ExomeCNV. Bioinformatics. 2011 Oct; 27(19):2648–54. doi: 10.1093/bioinformatics/btr462 PMID: 21828086
8. Love MI, Myšičková A, Sun R, Kalscheuer V, Vingron M, Haas Sa. Modeling read counts for CNV detection in exome sequencing data. Statistical Applications in Genetics and Molecular Biology. 2011 Jan; 10(1). doi: 10.2202/1544-6115.1732 PMID: 23089826
9. Li J, Lupat R, Amarasinghe KC, Thompson ER, Doyle Ma, Ryland GL, et al. CONTRA: copy number analysis for targeted resequencing. Bioinformatics. 2012 May; 28(10):1307–13. doi: 10.1093/bioinformatics/bts146 PMID: 22474122
10. Krumm N, Sudmant PH, Ko A, O'Roak BJ, Malig M, Coe BP, et al. Copy number variation detection and genotyping from exome sequence data. Genome Research. 2012 Aug; 22(8):1525–32. doi: 10.1101/gr.138115.112 PMID: 22585873

11. Plagnol V, Curtis J, Epstein M, Mok KY, Stebbings E, Grigoriadou S, et al. A robust model for read count data in exome sequencing experiments and implications for copy number variant calling. Bioinformatics. 2012; 28(21):2747–2754. doi: 10.1093/bioinformatics/bts526 PMID: 22942019

12. Koboldt DC, Zhang Q, Larson DE, Shen D, McLellan MD, Lin L, et al. VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. Genome Research. 2012 Mar; 22 (3):568–76. doi: 10.1101/gr.129684.111 PMID: 22300766

13. Fromer M, Moran JL, Chambert K, Banks E, Bergen SE, Ruderfer DM, et al. Discovery and statistical genotyping of copy-number variation from whole-exome sequencing depth. American Journal of Human Genetics. 2012 Oct; 91(4):597–607. doi: 10.1016/j.ajhg.2012.08.005 PMID: 23040492

14. Gartner JJ, Davis S, Wei X, Lin JC, Trivedi NS, Teer JK, et al. Comparative exome sequencing of metastatic lesions provides insights into the mutational progression of melanoma. BMC Genomics. 2012 Jan; 13(1):505. doi: 10.1186/1471-2164-13-505 PMID: 23006843

15. Magi A, Tattini L, Cifola I, D'Aurizio R, Benelli M, Mangano E, et al. EXCAVATOR: detecting copy number variants from whole-exome sequencing data. Genome Biology. 2013 Jan; 14(10):R120. doi: 10.1186/gb-2013-14-10-r120 PMID: 24172663

16. Backenroth D, Homsy J, Murillo LR, Glessner J, Lin E, Brueckner M, et al. CANOES: Detecting rare copy number variants from whole exome sequencing data. Nucleic Acids Research. 2014; 42(12):1–9. doi: 10.1093/nar/gku345

17. Wang C, Evans JM, Bhagwate AV, Prodduturi N, Sarangi V, Middha M, et al. PatternCNV: a versatile tool for detecting copy number changes from exome sequencing data. Bioinformatics. 2014 Sep; 30 (18):2678–2680. doi: 10.1093/bioinformatics/btu363 PMID: 24876377

18. Jiang Y, Oldridge Da, Diskin SJ, Zhang NR. CODEX: a normalization and copy number variation detection method for whole exome sequencing. Nucleic Acids Research. 2015 Mar; 43(6):e39–e39. doi: 10.1093/nar/gku1363 PMID: 25618849

19. Boeva V, Zinovyev A, Bleakley K, Vert JP, Janoueix-Lerosey I, Delattre O, et al. Control-free calling of copy number alterations in deep-sequencing data using GC-content normalization. Bioinformatics. 2011 Jan; 27(2):268–9. doi: 10.1093/bioinformatics/btq635 PMID: 21081509

20. Klambauer G, Schwarzbauer K, Mayr A, Clevert DA, Mitterecker A, Bodenhofer U, et al. cn.MOPS: mixture of Poissons for discovering copy number variations in next-generation sequencing data with a low false discovery rate. Nucleic Acids Research. 2012 May; 40(9):e69. doi: 10.1093/nar/gks003 PMID: 22302147

21. Bellos E, Coin LJM. cnvOffSeq: detecting intergenic copy number variation using off-target exome sequencing data. Bioinformatics. 2014 Sep; 30(17):i639–45. doi: 10.1093/bioinformatics/btu475 PMID: 25161258

22. Kuilman T, Velds A, Kemper K, Ranzani M, Bombardelli L, Hoogstraat M, et al. CopywriteR: DNA copy number detection from off-target sequence data. Genome Biology. 2015 Dec; 16(1):49. doi: 10.1186/s13059-015-0617-1 PMID: 25887352

23. Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, et al. The Sequence Alignment/Map format and SAMtools. Bioinformatics. 2009 Aug; 25(16):2078–2079. doi: 10.1093/bioinformatics/btp352 PMID: 19505943

24. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. arXiv preprint arXiv. 2013 Mar; 00(00):3. Available from: http://arxiv.org/abs/1303.3997.

25. Boeva V, Popova T, Lienard M, Toffoli S, Kamal M, Le Tourneau C, et al. Multi-factor data normalization enables the detection of copy number aberrations in amplicon sequencing data. Bioinformatics. 2014 Jul;p. 1–8.

26. Aird D, Ross MG, Chen WS, Danielsson M, Fennell T, Russ C, et al. Analyzing and minimizing PCR amplification bias in Illumina sequencing libraries. Genome Biology. 2011 Jan; 12(2):R18. doi: 10.1186/gb-2011-12-2-r18 PMID: 21338519

27. Benjamini Y, Speed TP. Summarizing and correcting the GC content bias in high-throughput sequencing. Nucleic Acids Research. 2012 May; 40(10):e72. doi: 10.1093/nar/gks001 PMID: 22323520

28. Cleveland WS. Robust Locally Weighted Regression and Smoothing Scatterplots. Journal of the American Statistical Association. 1979 Dec; 74(368):829–836. doi: 10.1080/01621459.1979.10481038

29. Kaiser J, Schafer R. On the use of the I0-sinh window for spectrum analysis. IEEE Transactions on Acoustics, Speech, and Signal Processing. 1980 Feb; 28(1):105–107. doi: 10.1109/TASSP.1980.1163349

30. Xi R, Hadjipanayis AG, Luquette LJ, Kim TM, Lee E, Zhang J, et al. Copy number variation detection in whole-genome sequencing data using the Bayesian information criterion. Proceedings of the National Academy of Sciences of the United States of America. 2011 Nov; 108(46):E1128–36. doi: 10.1073/pnas.1110574108 PMID: 22065754

31. Olshen AB, Venkatraman ES, Lucito R, Wigler M. Circular binary segmentation for the analysis of array-based DNA copy number data. Biostatistics. 2004 Oct; 5(4):557–72. doi: 10.1093/biostatistics/kxh008 PMID: 15475419

32. Olshen AB, Bengtsson H, Neuvial P, Spellman PT, Olshen RA, Seshan VE. Parent-specific copy number in paired tumor-normal studies using circular binary segmentation. Bioinformatics. 2011 Aug; 27 (15):2038–46. doi: 10.1093/bioinformatics/btr329 PMID: 21666266

33. Ben-Yaacov E, Eldar YC. A fast and flexible method for the segmentation of aCGH data. Bioinformatics. 2008 Aug; 24(16):i139–45. doi: 10.1093/bioinformatics/btn272 PMID: 18689815

34. Tibshirani R, Wang P. Spatial smoothing and hot spot detection for CGH data using the fused lasso. Biostatistics. 2008 Jan; 9(1):18–29. doi: 10.1093/biostatistics/kxm013 PMID: 17513312

35. Cock PJA, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, et al. Biopython: freely available Python tools for computational molecular biology and bioinformatics. Bioinformatics. 2009 Jun; 25(11):1422–1423. doi: 10.1093/bioinformatics/btp163 PMID: 19304878

36. Reich M, Liefeld T, Gould J, Lerner J, Tamayo P, Mesirov JP. GenePattern 2.0. Nature Genetics. 2006 May; 38(5):500–1. doi: 10.1038/ng0506-500 PMID: 16642009

37. Thorvaldsdóttir H, Robinson JT, Mesirov JP. Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. Briefings in Bioinformatics. 2013 Mar; 14(2):178–92. doi: 10.1093/bib/bbs017 PMID: 22517427

38. Saldanha AJ. Java Treeview–extensible visualization of microarray data. Bioinformatics. 2004 Nov; 20 (17):3246–8. doi: 10.1093/bioinformatics/bth349 PMID: 15180930

39. Oesper L, Mahmoody A, Raphael BJ. THetA: inferring intra-tumor heterogeneity from high-throughput DNA sequencing data. Genome Biology. 2013 Jul; 14(7):R80. doi: 10.1186/gb-2013-14-7-r80 PMID: 23895164

40. Blankenberg D, Von Kuster G, Bouvier E, Baker D, Afgan E, Stoler N, et al. Dissemination of scientific software with Galaxy ToolShed. Genome Biology. 2014 Jan; 15(2):403. doi: 10.1186/gb4161 PMID: 25001293

41. Mohiyuddin M, Mu JC, Li J, Bani Asadi N, Gerstein MB, Abyzov A, et al. MetaSV: an accurate and integrative structural-variant caller for next generation sequencing. Bioinformatics. 2015 Aug; 31 (16):2741–2744. doi: 10.1093/bioinformatics/btv204 PMID: 25861968

42. Botton T, Yeh I, Nelson T, Vemula SS, Sparatta A, Garrido MC, et al. Recurrent BRAF kinase fusions in melanocytic tumors offer an opportunity for targeted therapy. Pigment cell & melanoma research. 2013 Nov; 26(6):845–51. doi: 10.1111/pcmr.12148

43. Shain AH, Garrido M, Botton T, Talevich E, Yeh I, Sanborn JZ, et al. Exome sequencing of desmoplastic melanoma identifies recurrent NFKBIE promoter mutations and diverse activating mutations in the MAPK pathway. Nature Genetics. 2015;. doi: 10.1038/ng.3382 PMID: 26343386

44. Quinlan AR, Hall IM. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics (Oxford, England). 2010 Mar; 26(6):841–2. doi: 10.1093/bioinformatics/btq033

45. Lawrence M, Huber W, Pagès H, Aboyoun P, Carlson M, Gentleman R, et al. Software for computing and annotating genomic ranges. PLoS Computational Biology. 2013 Jan; 9(8):e1003118. doi: 10.1371/journal.pcbi.1003118 PMID: 23950696