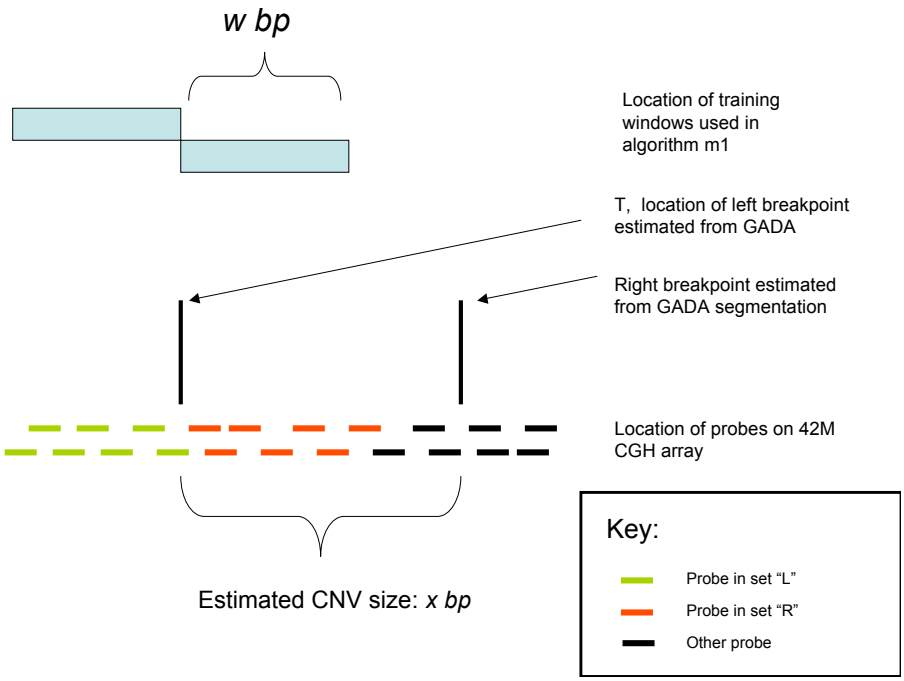
Supplementary Information for "Mutation spectrum revealed by breakpoint sequencing of human germline CNVs"

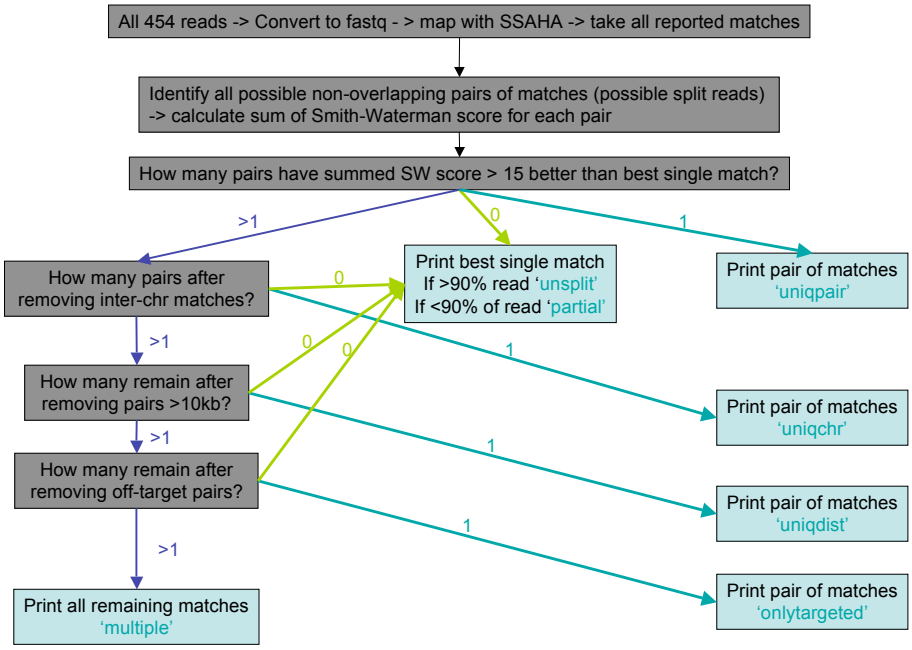
Donald F. Conrad[1], Christine Bird[1], Ben Blackburne[1], Sarah Lindsay[1], Lira Mamanova[1], Charles Lee[2], Daniel J. Turner[1], Matthew E. Hurles[1]

[1] Wellcome Trust Sanger Institute, Hinxton, Cambridge, UK

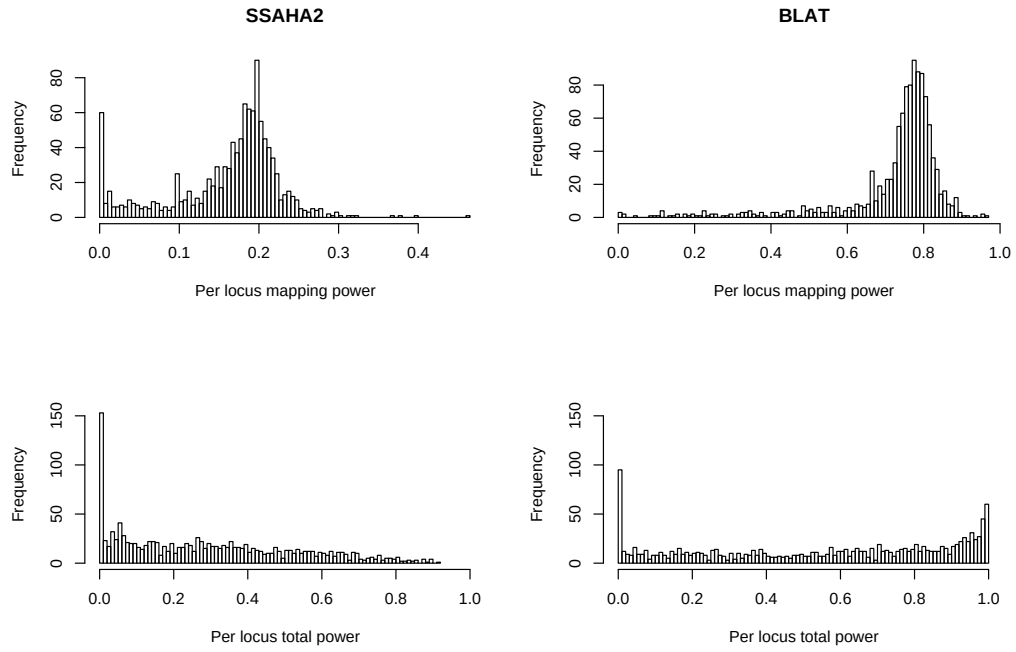
[2] Department of Pathology, Brigham and Womens Hospital and Harvard Medical School, Boston, MA, USA



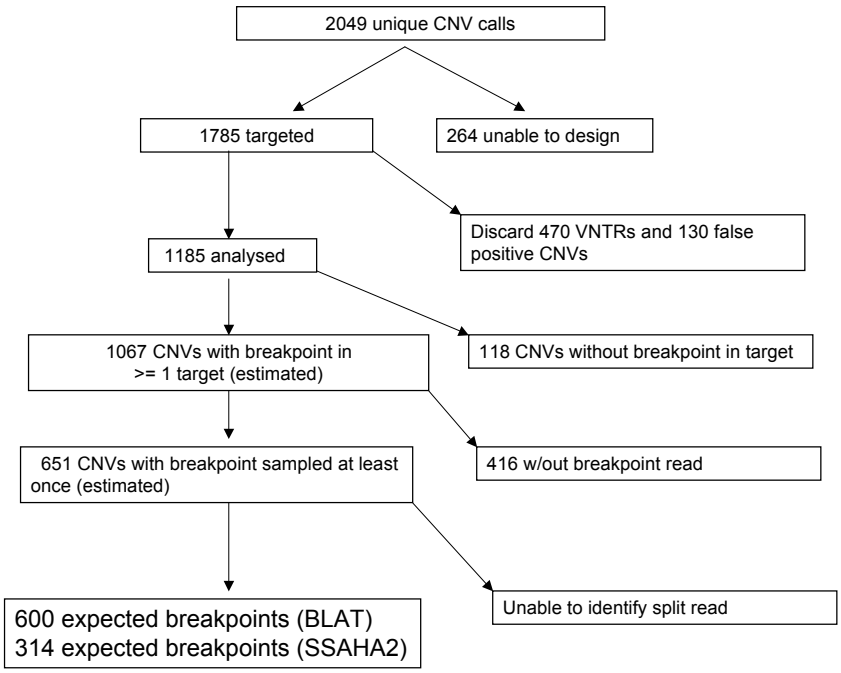
Supplementary Figure 1: **Overview of parameters and data used in calculating confidence intervals.** Here we present a visual guide to the parameters used by the "m1" algorithm for constructing breakpoint confidence intervals. The algorithm takes as input the locations and associated intensities of the CGH probes from the 42M CGH experiment, as well as estimated breakpoint locations for each CNV (generated by a CNV discovery algorithm; we used GADA for this analysis). For a full description of the algorithm see Supplementary Note.



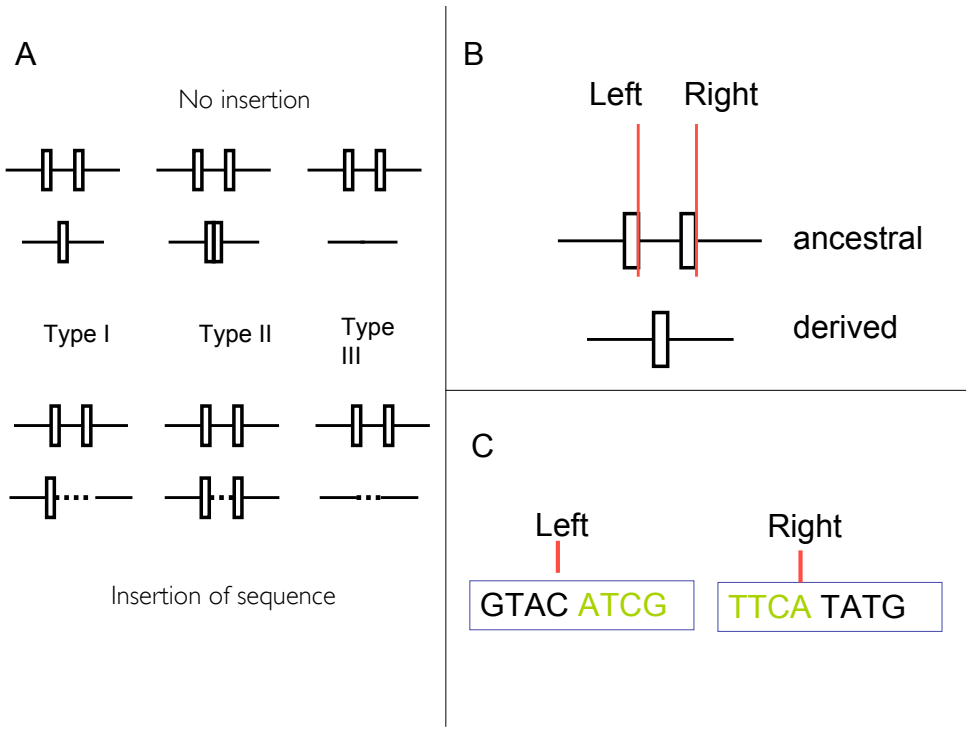
Supplementary Figure 2: **Overview of SSAHA2 mapping pipeline for identifying split reads.** For each read it is assessed whether it maps fully to the reference sequence or whether a pair of split-reads is a better alignment (Smith-Waterman score > 15 better than for the best single alignment). If multiple pairs of split-reads pass this threshold, least likely pairs are filtered out sequentially to arrive at either a single best split-read pair for each read, or, once all filters have been applied, a set of equally good split-reads.



Supplementary Figure 3: **Mapping power and total power.** The total power of our given pipeline (based on SSAHA2 or BLAT) to detect a CNV breakpoint is a function of both the sampling power and mapping power of the algorithm being used. We used simulations to estimate the sampling power, mapping power, and total power of our particular experiment to detect breakpoints for each CNV targeted by our capture array (simulations described in Supplementary Note). Top row: For each CNV locus that contained mapped reads in the real data, we estimated the power of the SSAHA2 and BLAT pipelines to accurately classify a split read containing a simple deletion breakpoint from that locus. Bottom row: we present the distribution of per-locus total power to detect simple deletion breakpoints for all validated, non-VNTR loci. Left panel- SSAHA2 results, right panel- BLAT results.



Supplementary Figure 4: **Overview of experimental power.** This flow chart summarizes, step-by-step, the impact of our experimental design on the power to identify CNV breakpoints. The regions targeted for study corresponded to 2049 unique CNVs but several factors lowered the number of assayable CNVs even before considering the sequencing data. Probe selection for the capture array was able to successfully produce at least one block of capture probes for 3,263 (88%) of the target regions, corresponding to 1785 CNVs (Methods). We know that approximately 15% of the CNVs targeted on the array are false positives, and about 9% of CNVs will not have a target region containing a breakpoint (we estimate that the CIs cover a breakpoint 70% of the time, $(1 - 0.7)^2 = 0.09$). Furthermore, the complex repeat structure of VNTRs makes obtaining accurate estimates of their breakpoints extremely challenging with short reads, and we excluded the 25% of loci with over 50% of their sequence contained in VNTR annotation from analysis. Combining these figures, and accounting for overlap among categories, we believe the number of CNVs for which we could possibly sequence a breakpoint to be 1067. The last two steps in the flow diagram show the loss of breakpoints due to sampling and mapping power, respectively.



Supplementary Figure 5: **Annotation of microhomology at CNV breakpoints.** Here we present our system for describing patterns of homology around deletion breakpoints. (A) classes of microhomology. Illustrations are provided for both the case of deletion with no subsequent insertion of sequence (top) and with insertion of sequence (bottom). In each class the breakpoints of the deletion occur within or immediately adjacent to a pair of homologous sequences, represented by rectangles. It may be biologically relevant to distinguish among breakpoint sequences where 1, 2 or 0 copies of the homology remains following the deletion (what we have named type I, II and III, respectively). In the present study we only annotate "Type I" microhomology; we developed a simple rule-based system for marking the location of breakpoints in the presence of microhomology. (B) In the presence of microhomology, the breakpoints ("Left" or "Right") are always placed to the right of the microhomology. (C) the position of the left break always defines the first base before the mutated sequence, and the position of the right break defines the last base within the mutated sequence. Here mutated sequence is colored green.

Supplementary Note

1 Confidence Intervals

Here, we describe in detail the "m1" algorithm for constructing breakpoint confidence intervals. The algorithm takes as input the locations and associated intensities of the CGH probes from the 42M CGH experiment, as well as estimated breakpoint locations for each CNV (generated by a CNV discovery algorithm; we used GADA for this analysis). A rough visual guide to the parameters and regions is presented in Supplementary Figure 1.

We define a breakpoint as the physical position at which there is a transition in relative copy number between the test and reference sample in a CGH experiment. This method is based on treating the CGH intensity data in the vicinity of a breakpoint as a simple mixture of two normal distributions, which leads to the following likelihood function for the breakpoint location,

$$Like(B) = \prod_{i \in L} Z(x_i; \hat{\mu_1}, \hat{\sigma_1}) + \prod_{i \in R} Z(x_i; \hat{\mu_2}, \hat{\sigma_2}) \quad (1)$$

where L is the set of all indices for probes that fall in the region of copy normal sequence outside of the breakpoint, R the set of all indices that fall in the region of copy variant sequence inside the breakpoint, $\hat{\mu_1}$ and $\hat{\sigma_1}$ and the estimates of the mean and standard deviation of ratios in the left region, $\hat{\mu_2}$ and $\hat{\sigma_2}$ the analogous estimates for the right. Z is the probability distribution function for a normal distribution with the specified mean and variance.

In what follows we will describe the procedure for constructing a confidence interval for the left breakpoint of a CNV; the procedure for the right breakpoint is analogous. In order to estimate values for the μ's and σ's, we first create training sets of probes that are likely drawn from the region outside the CNV (set L) and probes that are likely drawn from inside the CNV (set R).

- Define the training window size in basepairs, w, to be the minimum of the following two numbers: 10000, and the size of the CNV, x.
- Define L to be the indices of all probes within w bp to the left of the estimated breakpoint location.
- Define R to be the indices of all probes within w bp to the right of the breakpoint. Using our data, set R is guaranteed to contain at least 10 probes, as we required aneuploidy at 10 consecutive probes to call a CNV in the discovery project. If there are fewer than 3 probes in set L then the algorithm ends and no CI is reported.
- Estimate $\hat{\mu_1}$ and $\hat{\sigma_1}$ as the mean and standard deviation of probe intensities in set L; and likewise $\hat{\mu_2}$ and $\hat{\sigma_2}$ as the mean and standard deviation of probe intensities in set R.

- Call the *a priori* estimate of the breakpoint location, T. Define the domain of Like(B) as $(T - w, T + w)$.

We evaluate Like(B) over a grid of points, one point between each consecutive probe pair with an edge in this domain. We use Bayes' rule to transform the resulting likelihoods into posterior probabilities, and construct intervals centered on the breakpoint that are symmetric in probe space and contain 95% of this posterior.

1.1 Confidence Intervals: Results

We estimated confidence intervals for breakpoints of all 1,174 CNVs detected in the CGH experiment comparing NA12878 and NA10851, and 1,304 CNVs detected in the CGH experiment comparing NA15510 and NA10851 (i.e. 2,348 and 2,608 CIs in total), using both methods m1 and m2. After merging CIs within samples and between samples, and accounting for breakpoints for which CIs could not be successfully designed, we obtained a set of 3,712 target regions spanning 3.6Mb of non-contiguous genomic sequence. The minimum, median, and maximum target region sizes were 135bp, 506bp, and 6kb respectively.

To assess the accuracy of these confidence intervals, we compiled published sequenced CNV breakpoints and identified breakpoints corresponding to the targeted CNVs (see Methods in main paper). We found 194 sequenced breakpoints for the NA15510 experiment and 156 for NA12878, using a threshold of 70% reciprocal overlap to declare two events as identical. The median error between the GADA-estimated breakpoints and the true breakpoints was roughly equal to the distance between probes on the CGH array (median 54bp for NA15510, 62bp for NA12878), suggesting that half of the estimated breakpoint locations are localized to the exact pair of probes. Using a more relaxed threshold of 50% reciprocal overlap to declare identity led to slightly larger error estimates (245 events in NA15510 with 71bp median error; 199 events in NA12878 with 88bp median error).

The two CI-estimation methods perform similarly well, with m1 perhaps slightly more efficient in terms of breakpoints/basepair. Method m1 produced confidence intervals spanning 101.69kb (Left CIs) and 100.38kb (Right CIs) that covered 247 and 240 breakpoints respectively. Method m2 produced confidence intervals spanning 91.625kb (Left CIs) and 93.33kb (Right CIs) that covered 221 and 237 breakpoints respectively.

We compared the performance of the most efficient breakpoint-capturing scheme to a fixed window strategy. The right break confidence intervals designed with m1 spanned a total of 100.38kb, for an average window size of 286bp. A strategy using fixed 286bp CIs centered on the estimated break locations captured virtually as

many breakpoints as the variable window design (233 versus 240), but this is attributable to the fact that the average confidence interval size contains information about the typical uncertainty about the breakpoint location.

2 Split Read Ontology

Prior to mapping the sequence reads, we devised a coding system to comprehensively categorise reads which do not align contiguously to the reference genome and thus potentially span CNV breakpoints. We term these reads with discontinuous mappings "split reads" if both ends of the reads are mapped to the reference, otherwise they are "partially aligned". Each end of a split read is annotated with a mapping location and strand of alignment, encoded as a 5 character string: mapping order of $5'$ end of read (1st or 2nd), mapping order of $3'$ end, mapping strand of $5'$ end (+ or -), mapping strand of $3'$ end, and a symbol indicating whether both ends map to the same (C) or different (X) chromosomes. Therefore a 12++C split would indicate the $5'$ (1) and $3'$ (2) ends of the split read mapping to the top strand of the same chromosome (C), and a $21 - +$X would indicate the right end mapping to the bottom strand of one chromosome and the left end mapping to the top strand of a different chromosome (X). For simple deletions, we expect the mappings of the split read to be on the same chromosome, on the same strand, and in the same order on the chromosome as in the read (i.e. 12++C or 21−−C), while for tandem duplications (longer than a sequence read), we expect the mappings of the split read on the same chromosome, on the same strand, but in the reverse order in the read as on the chromosome (i.e. 21++C or 12−−C).

3 Mapping Pipelines

Breakpoints were identified by searching for reads with mappings split by structural variations. We created two pipelines for mapping reads in order to maximize sensitivity to a wide range of breakpoint structures. In both pipelines reads were mapped to the whole human genome assembly (NCBI36) The first approach, using SSAHA2, was designed to detect breakpoints from a wide variety of mutation processes: deletions, tandem duplications and dispersed duplications (Supplementary Figure 2). In order to allow for split mappings, the SSAHA2 options were tuned to produce multiple hits per read (-454 -seeds 2 -diff -1 -kmer 12 -skip 5). Duplicate reads were identified and removed by searching for those that produced identical mapping coordinates. From these mappings, all possible pairs with fewer than 15 overlapping bases in the read were considered. These pairs were filtered to remove those that had a combined Smith-Waterman score less than 15 better

than the best unsplit alignment. If this filtering resulted in a single pair, this was reported. Then inter-chromosome splits, those larger than 10kb, and those further than 500bp from the target region were successively removed until a single pair remained, which was then reported. If multiple pairs remained after this filter, these were all reported with an appropriate tag. This filtering process classifies split reads mapping within 500bp of the defined target regions as on-target.

The second mapping approach used the command line implementation of BLAT version 34 with all options set to their default value. For all reads with at least one mapping, we ordered these mappings by BLAT score (an ad hoc measure of alignment quality) and retained only the highest ranked mapping. Define the alignment footprint of a read to be the region from the mapping position of the first base of the read to the mapping position of the last base of the read. Best-hit reads with an alignment footprint on the genome greater than 500bp were retained as possible split reads. Candidate split reads with fewer than 90% of their bases aligned were discarded. The subset of remaining candidate split reads whose alignment footprint has greater than 50% reciprocal overlap with a CNV targeted on the capture array were classified as true split reads.

The read mappings potentially reveal breakpoints down to the single-base level. There may be some ambiguity induced by either sequencing error, or by microhomology at the breakpoint ends.

4 Power Analysis: Methods

In the experiment described in the main text, we targeted 1785 CNVs for capture and sequencing, and recovered breakpoints for approximately 18% of these. While this yield appear low, we cannot make precise statements about our expected result without understanding the power of our experiment to recover CNV breakpoints.

Modeling the power of the experiment is potentially complex with many variables - including read lengths, number of reads, sequence context, type of rearrangement (deletion vs duplication), complexity of rearrangement (non-templated sequence, multiple gains and loses in short stretch). Instead of doing a general power analysis exploring all possible parameter values, we simulated the power of our experiment, specifically- conditioning on the number of reads that mapped on target, and the length of the reads and their mapping locations within the targets. We explore over uncertainty in the true breakpoint location at each CNV.

Say that there are K reads sampled from a given CNV region. We factor our power to detect the breakpoint of that CNV into two components:

$$\Pr(\text{detect breakpoint}) = C * \sum_{i=1}^{K} [\Pr(\text{read } i \text{ contains breakpoint}) \times \Pr(\text{identifying breakpoint } | \text{read } i \text{ contains breakpoint})], \quad (2)$$

where C is a normalization constant. We refer to the first component as *sampling power* and the second component as *mapping power*. We calculate an approximation to the first component, Pr(sequencing a read with the breakpoint), using just the configuration of reads that we recovered (their number and mapping location). The second probability is a function of the mapping pipeline: it is the probability that, given a split read, we can correctly infer that it came from a CNV breakpoint. We designed simulation studies to estimate these probabilities, as well as the total power of our experiment.

4.1 Sampling power

To estimate sampling power, we assume the number, size and location of reads mapping to each CNV target is fixed, with values corresponding to that observed in the real data. Therefore we will estimate the sampling power to be 0 for a CNV with no mapped reads in the real data. We conservatively estimate the number of chromosomes carrying the CNV allele to be the call frequency in the CGH data of NA15510 and NA12878; thus each CNV will be present as a heterozygote in one or two of the three samples pooled for the pulldown.

We estimate the sampling power separately for each CNV target, and these individual power estimates can then be combined to give an overall estimate of experiment-wide power. For each CNV target we simulate a breakpoint uniformly at random within the target region (corresponding to either the left or right end of the CNV). We then count the number of reads from observed data that overlap this breakpoint. This procedure is done repeatedly, simulating 100 breakpoints for each target; power for a given target is estimated as the proportion of repeats for which at least one read contains the breakpoint.

4.2 Mapping power

Mapping power should be sensitive to the sequence context of the rearrangement, the location of the split within the read, and read size; additionally it will depend on the mutation model. In order to broadly assess the effect of different mutation processes on mapping power, we generated a set of split reads from 4 different mutation models. We simulated split reads from every validated, non-VNTR CNV, and the length of the reads at a CNV is simulated from the distribution of read lengths observed at that CNV in the real data. The four sets are

1. Deletions with 0bp of non-templated sequence inserted
2. Deletions with 5bp of non-templated sequence inserted
3. Deletions with 30bp of non-templated sequence inserted
4. Tandem Duplication with 0bp of non-templated sequence inserted

As we have used two distinct mapping pipelines, one using SSAHA2 and one using BLAT, we use this simulation framework to estimate the mapping power of both pipelines. For the following analyses, we only analyze those CNVs that which contained at least 1 mapped read in the real data. We define the power of breakpoint detection for all other CNVs to be 0.

5 Power Analysis: Results

5.1 Sampling Power

As we have modeled the experiment, the sampling power is a function of sequence coverage (number of bases sequenced/number of bases targeted) and the number chromosomes carrying the CNV.

Our first set of simulations investigated power to detect a CNV breakpoint conditional on the length and number of reads sequenced from each target region. Integrating over uncertainty in CNV breakpoint location, we estimated that our pulldown data gave us an average of 69% power to capture a CNV breakpoint across all loci analysed. We estimate that we have 90% power to capture a breakpoint when we have an average of about 2x haploid sequence coverage across both target regions of a CNV.

Our experimental design used a pool of genomic DNA from three unrelated individuals, meaning that in the case of a single heterozygous CNV we had only a 1/6 chance of sampling from the chromosome with the breakpoint. Simulating over a range of pooling sizes from 1 to 4 diploid samples, and assuming that the CNV allele is present on a single chromosome in each pool, we estimated that our average power to sequence a CNV breakpoint was $77\%, 68\%, 61\%$, and 55%, respectively. We conclude that our strategy of pooling 3 samples reduces our power by at most 15%.

5.2 Mapping Power

We simulated about 750,000 split reads each from four different mutation models: simple deletions, deletions with 5 or 30bp of non-templated sequence, and simple

tandem duplications. For all simulations the percent of reads mapping to the genome with both pipelines was extremely high, and interestingly, this percentage was not affected much by the amount of non-templated sequence inserted within deletion breakpoints.

SSAHA2. The power to correctly classify a split read did not appear to vary greatly across regions, contrary to what would be expected if power was highly dependent on sequence context (Supplementary Figure 3). Surprisingly, the sensitivity of our pipeline is about 14% per-read for deletions. For comparison, 0.25% of our real (unsimulated) mapped reads were identified as CNV reads; taken in the context of our simulation results this suggests that we may be missing as many as 85% of the breakpoint reads in the data. The per-locus mapping power for simple deletions is 16%. Interestingly, duplication reads intrinsically have more mappings per read (1.6) than deletion reads (1.32) with the SSAHA2 pipeline. The power to detect duplication split reads appeared slightly lower, on average, than deletion split reads.

After mapping with the SSAHA2 pipeline, reads are placed into one of 5 mutually exclusive categories: deletion, tandem duplication, interchromosomal duplication, unsplit, and "other". This last class includes split reads with multiple ambiguous mappings and reads for which only one end is alignable. Interestingly, about 60% of the split reads from all four mutation models were classified as "other", suggesting that additional breakpoints could be mind from this category in our real data.

BLAT. The estimated per-read sensitivity of the BLAT pipeline was 57% for deletions with no NTS and essentially the same for deletions with 30bp of NTS. The estimated per-locus sensitivity is higher: on average 77% of deletion breakpoint reads from a validated, non-VNTR locus will be identifiable (Supplementary Figure 3). In striking contrast to the SSAHA pipeline, we have virtually no power to successfully map simple tandem duplications, which is an expected property of the BLAT alignment model.

Summary. We interpret the results from both pipelines in the following manner: BLAT, which was designed for identifying gapped alignments (often with multiple gaps) during mapping of cDNAs, has high sensitivity for finding deletion split reads. The BLAT model is restrictive in that all pieces of an aligned read must align to the same chromosome, and in the same 5'-3' orientation, and our increased sensitivity to deletions comes at the expense of identifying other types of rearrangements. SSAHA2 provides a more flexible model that does have power to detect tandem and dispersed duplications, as well as more complex deletions,

but this flexibility necessitates a drop in sensitivity in order to control the false discovery rate.

5.3 Total Power

With estimates of both per-locus mapping power and sampling power in hand, we estimated the overall power of the experiment using the relationship in Equation 2. This includes all sites targeted on the array. The per locus power distribution using the SSAHA2 and BLAT pipelines is plotted in Supplementary Figure 3. The average per-locus total power for SSAHA2 is estimated to be 29%, and 62% for BLAT.

Several aspects of real data are not captured by these simulations. Perhaps most importantly we have not introduced any sequencing error into the simulated reads. Second, split reads are more divergent from the reference genome than unsplit reads, and we have not attempted to model the effect that this lower sequence homology may have on capture efficiency. The impact of the former could be addressed through simulation, and the impact of the latter could possibly be estimated from real data.

We have only limited information about the design of the capture array, and specifically about placement of probes within our target regions. It is possible that there are significant gaps in coverage within any target region that may reduce power, especially if such gaps are biased towards the location of CNV breakpoints. We assume that there is no error in placement of our target regions and that all target regions contain breakpoints.

A summary of the entire capture experiment is presented in Supplementary Figure 4. Based on these results we can see that the experimental design can easily account for the relatively low proportion of CNVs with inferred breakpoints. The simulation framework we describe here could be used to inform experimental design going forward, exploring the impact of features such as pooling, read length and sequence depth on breakpoint yield.