ONLINE METHODS

CNV map. A previous study[14] described an ultra-high-density set of arrays, consisting of 42 million oligonucleotides tiled across the genome at an average of one probe every 56 bp, which were used to discover CNVs in 41 genomes, including the 3 individual genomes analyzed here, NA15510, NA12878 and NA10851. The CNVs in these 3 genomes were detected in two array-CGH experiments.

CNV calls. CNV calls were generated using the GADA segmentation algorithm[18] with parameters $t = 10$, a = 2.0, m = 10. Calls were not merged within samples before confidence interval design to maximize sensitivity to true breakpoints.

Confidence intervals. Here 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. As described in the main text, we used two likelihood-based methods for constructing a confidence interval on a breakpoint. The first (m1) 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:

$$\text{Like}(B) = \prod_{i \in L} Z(x_i; \hat{\mu}_1, \hat{\sigma}_1) + \prod_{i \in R} Z(x_i; \hat{\mu}_2, \hat{\sigma}_2)$$

where L is the set of all indices for probes that fall in the region left of the breakpoint, R the set of all indices that fall in the region right of the breakpoint, $\hat{\mu}_1$ and $\hat{\sigma}_1$ the estimates of the mean and s.d. of ratios in the left region, and $\hat{\mu}_2$ 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. We evaluate Like(B) over a grid of points, one point between each consecutive probe pair in the candidate region. We use Bayes' rule to transform the likelihoods into posterior probabilities, and we construct intervals centered on the breakpoint that are symmetric in probe space and contain 95% of this posterior. For full details of the algorithm, see **Supplementary Note**.

The second approach (m2) is predicated on a class of econometrics models known as structural change models. The methodology we use is based on a simple formulation of such a model—a linear model in which the value of the regression coefficient changes with time, but in a piecewise constant fashion. The intensity of the tth probe on the array, y_t, comes from a distribution specified by δ_j, the mean of the jth segment:

$$y_t = x_t' \delta_j + \varepsilon_t, \text{ for } t = T_{j-1} + 1, ..., T_j, j = 1, ..., m+1$$

where m is the number of breakpoints and the T_js are the locations of the breakpoints. An asymptotic theory for inference on this model has been presented[31], and this theory allows deduction of confidence intervals on the breakpoint estimator, $\hat{T}_j$. These confidence intervals are derived from a stochastic process composed of two independent Brownian motions; the drift parameter of the process is a function of the ratio of the two segment means, whereas the scale parameter is a function of the variance and covariance of probe intensities. We used functions in the R package 'strucchange' to calculate these confidence intervals[32]. To the best of our knowledge, the only prior genomics application of this method for constructing confidence intervals was in the study of yeast gene transcription[33].

Array design. We used the results from both methods for forming confidence intervals (m1 and m2, see above) to construct the target regions on the capture array. To merge confidence intervals constructed by each method within samples, we took the intersection of corresponding intervals, and we removed all 326 merged confidence intervals larger than 5 kb. We further merged confidence intervals between samples by taking the union of intervals. The net result of this process was a set of 3,712 target regions spanning 3.6 Mb of noncontiguous genomic sequence. The minimum, median and maximum target region size were 135 bp, 506 bp and 6 kb, respectively. We believe the optimal targeting strategy is to target 3–5 Mb of sequence on a single capture array (385k), with the condition that the minimal size for a targeted region be 135 bp.

Roche/NimbleGen's array design algorithm pads out any target region to a minimum length of 250 bp. After this padding, overlapping regions are consolidated. Probes are selected to be unique hits to the genome, allowing up to five insertions or deletions. There are 9,642 blocks of probes in total. Some target regions submitted for design to Roche did not yield any unique probes, and we have at least one block for 3,263 (88%) of the target regions. Thirty-three percent of these regions correspond to CNVs longer than 5 kb.

Capture and sequencing. The genomes of NA15510, NA10851 and NA12878 were pooled in equimolar amounts and then hybridized to the capture array according to the manufacturer's protocol. We used the FLX Standard chemistry for 454 sequencing.

Read mapping. We identified breakpoints by searching for reads with mappings split by structural variations[1]. We created two pipelines for mapping reads 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[34]. The second mapping approach used the command-line implementation of BLAT version 34 with all options set to their default values[35]. Full details of both mapping pipelines are presented in the **Supplementary Note**. The read mappings potentially reveal breakpoints down to the single-base level. There may be some ambiguity induced either by sequencing error or by microhomology at the breakpoint ends.

Power simulations. We considered the power of our experiment to locate a CNV breakpoint to be comprised of two parts: sampling power and mapping power. In the simulation of sampling power, our goal was to estimate the probability that a breakpoint was captured by a read at each targeted CNV locus. We assumed that the location and length of each read, as provided by the original SSAHA2 mappings, were known without error. We simulated 1,000 random breakpoints for each CNV and calculated the proportion of breakpoints that are covered by at least one read in our data.

To quantify mapping power we simulated about 400 split reads from each CNV and recorded the percentage of simulated reads for which the breakpoint is correctly identified by SSAHA2. The power to map a read containing a CNV breakpoint will depend upon the size of the read, the location of the breakpoint within the read, the mutation process forming the read and the mapping algorithm. Here we explored four mutation models, including deletions with a range of amounts of nontemplated sequence (0–30 bp) and tandem duplications. CNV breakpoints were assumed to be randomly distributed within target intervals, and the length of reads from a given target region was simulated from the distribution of reads observed at the target in the real data.

We derived estimates of the total number of breakpoints that we expected to identify (the 'total power') by combining information on both sampling and mapping power. Additional details on power simulations can be found in the **Supplementary Note**.

Analysis of breakpoints. To determine a consensus of the breakpoint at the base-pair level, we assembled the fasta sequences of all split reads and the targets they intersect in a gap4 database[36], allowing joins only between the reads and not between the targets (allowing a maximum 10% mismatch). Contigs were successfully assembled for 315 deletions and all 3 tandem duplications. We investigated each target locus by joining the sequences of reads and targets manually using the 'find repeats' tool of gap4 to view matches in a direct or inverted orientation. Inserted sequence was defined as any extra sequence between the breakpoints in the reads that was not present in the reference sequence. The sequences and sizes of flanking regions, inserted bases and regions of microhomology were recorded. Although there are multiple possible patterns of homology around a breakpoint—depending on whether homology is observed 5′ or 3′ of the breaks, or both, and on whether it is observed on the ancestral or derived chromosomes, or both—we use a very specific definition (what we call 'type I' microhomology, **Supplementary Fig. 5**). We used BLAT to match inserted sequences >19 bases back to the reference genome to determine whether the inserted sequence was part of the reference. This included regions containing more complex patterns, such as those with inverted sequence within the breakpoint.

Validation. Some split reads also showed evidence of inverted or inserted sequence, or both, in some cases originating from >3 kb away from the breakpoint. We validated these complex rearrangement structures by PCR

doi:10.1038/ng.564

and capillary resequencing in the test individuals. Primer sequences for these experiments are provided in **Supplementary Table 2**.

External data. We compiled known breakpoints from five major sequencing-based studies of CNV; these comprised 3,731 CNV observations greater than 400 bp, some of which represent the same events observed multiple times[1–3,26,37]. We matched 423 CNVs from the genome-wide tiling oligo–CGH study to this list of events, using a threshold of 70% reciprocal overlap to declare two events as identical.

URLs. Manufacturer's protocol for capture array hybridization is available from Roche/NimbleGen Systems, http://www.NimbleGen.com.

31. Bai, J. & Perron, P. Computation and analysis of multiple structural change models. *J. Appl. Econom.* **18**, 1–22 (2003).
32. Zeileis, A., Kleiber, C., Kramer, W. & Hornik, K. Testing and dating of structural changes in practice. *Comput. Stat. Data Anal.* **44**, 109–123 (2003).
33. Huber, W., Toedling, J. & Steinmetz, L.M. Transcript mapping with high-density oligonucleotide tiling arrays. *Bioinformatics* **22**, 1963–1970 (2006).
34. Ning, Z., Cox, A.J. & Mullikin, J.C. SSAHA: a fast search method for large DNA databases. *Genome Res.* **11**, 1725–1729 (2001).
35. Kent, W.J. BLAT–the BLAST-like alignment tool. *Genome Res.* **12**, 656–664 (2002).
36. Staden, R., Beal, K.F. & Bonfield, J.K. The Staden package, 1998. *Methods Mol. Biol.* **132**, 115–130 (2000).
37. Wheeler, D.A. *et al.* The complete genome of an individual by massively parallel DNA sequencing. *Nature* **452**, 872–876 (2008).