Accurate *de novo* and transmitted indel detection in exome-capture data using microassembly

Giuseppe Narzisi[1,2], Jason A O'Rawe[3,4], Ivan Iossifov[1], Han Fang[3,4], Yoon-ha Lee[1], Zihua Wang[1], Yiyang Wu[3,4], Gholson J Lyon[3,4], Michael Wigler[1] & Michael C Schatz[1]

We present an open-source algorithm, Scalpel (http://scalpel.sourceforge.net/), which combines mapping and assembly for sensitive and specific discovery of insertions and deletions (indels) in exome-capture data. A detailed repeat analysis coupled with a self-tuning *k*-mer strategy allows Scalpel to outperform other state-of-the-art approaches for indel discovery, particularly in regions containing near-perfect repeats. We analyzed 593 families from the Simons Simplex Collection and demonstrated Scalpel's power to detect long (≥30 bp) transmitted events and enrichment for *de novo* likely gene-disrupting indels in autistic children.

Although the analysis of single-nucleotide variations (SNVs) has become a standard technique to study human genetics[1], indels in DNA sequences cannot be detected as reliably[2,3]. Indels are the second most common source of variation in human genomes and the most common structural variant[4]. Within microsatellites (simple sequence repeats, SSRs, of 1- to 6-bp motifs), indels alter the length of the repeat motif and have been linked to more than 40 neurological diseases[5]. Indels also are an important genetic component in autism: *de novo* indels that are likely to disrupt the encoded protein are nearly twice as abundant in affected children than in their unaffected siblings[6].

Detecting indels is challenging for several reasons: (i) reads overlapping the indel sequence are more difficult to map[7] and may be aligned with multiple mismatches rather than with a gap; (ii) irregularity in capture efficiency and nonuniform read distribution increase the number of false positives; (iii) increased error rates make indel detection very difficult within microsatellites; and, as shown in this study, (iv) localized near-identical repetitive sequences can create high rates of false positives. For these reasons, the size of indels detectable by available software tools has been relatively small, rarely more than a few dozen base pairs[8].

Two major paradigms are currently used for detecting indels. The first and most common approach is to map all of the input reads to the reference genome using a read mapper (such as Burrows-Wheeler Alignment tool (BWA), Bowtie or Novoalign), although the available algorithms are not as effective for mapping across indels of more than a few base pairs. Advanced approaches use paired-end information to perform local realignments to detect longer mutations (tools include GATK UnifiedGenotyper[1] and Dindel[9]), although, in practice, their sensitivity is greatly reduced for longer (≥30 bp) variants. Split-read methods (such as Pindel[10] and Splitread[11]) can theoretically find deletions of any size, but they have limited power to detect insertions owing to the short read length of current sequencing technologies. The second paradigm consists of performing *de novo* whole-genome assembly of the reads and detecting variations between the assembled contigs and the reference genome[12,13]. Although it has the potential to detect larger mutations, in practice this paradigm is less sensitive because detecting indels requires a fine-grained and localized analysis to correctly report homozygous and heterozygous mutations. Recently, three approaches have been developed that use *de novo* assembly for variation discovery: GATK HaplotypeCaller, SOAPindel[14] and Cortex[15]. Another recent approach, TIGRA[16], also uses localized assembly, but it has been tailored for breakpoint detection without reporting the indel sequence.

We present a DNA sequence microassembly pipeline, Scalpel, for detecting indels within exome-capture data (**Fig. 1**). By combining the power of mapping and assembly, Scalpel carefully searches the de Bruijn graph for sequence paths (contigs) that span each exon. The algorithm includes an on-the-fly repeat composition analysis of each exon coupled with a self-tuning *k*-mer strategy.

Using simulated reads, we confirmed previous findings that nine standard algorithms have reduced power to detect large (≥30 bp) indels[14,15]: Scalpel, SOAPindel[14], GATK-HaplotypeCaller, GATK-UnifiedGenotyper, SAMtools[17], FreeBayes[18], Platypus (http://www.well.ox.ac.uk/platypus), lobSTR[19] and RepeatSeq[20] (**Supplementary Notes 1** and **2**, **Supplementary Figs. 1** and **2** and **Supplementary Tables 1–5**). We also performed a large-scale validation experiment involving ~1,000 indels from one single exome. The individual has a severe case of Tourette syndrome and obsessive-compulsive disorder (sample ID: K8101-49685); the exome was sequenced to ≥20× coverage over 80% of the exome target using the Agilent 44-Mb SureSelect capture protocol and Illumina HiSeq2000 paired-end reads, averaging 90 bp in length after trimming. Indels were called using the three pipelines that had performed best with our simulated reads: Scalpel v.0.1.1 beta, SOAPindel v.2.0.1 and GATK HaplotypeCaller v.2.4.3

[1]Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, Cold Spring Harbor, New York, USA. [2]New York Genome Center, New York, USA. [3]Stanley Institute for Cognitive Genomics, Cold Spring Harbor Laboratory, Cold Spring Harbor, New York, USA. [4]Stony Brook University, Stony Brook, New York, USA. Correspondence should be addressed to G.N. (gnarzisi@cshl.edu).

RECEIVED 15 APRIL; ACCEPTED 11 JULY; PUBLISHED ONLINE 17 AUGUST 2014; DOI:10.1038/NMETH.3069

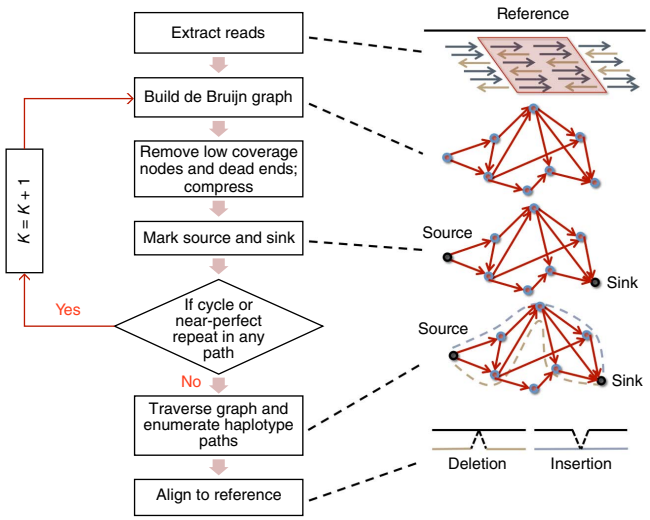

Figure 1 | Overview of the Scalpel algorithm workflow. Extracted reads include well-mapped reads, soft-clipped reads, and reads that fail to map but are anchored by their mate. The assembled sequences are aligned to a reference using the standard Smith-Waterman-Gotoh alignment algorithm with affine-gap penalties.

(Online Methods). Interestingly, there was only ~37% concordance among calls made by all of the pipelines, and each method reported hundreds of indels unique to that pipeline (**Fig. 2a**), which is in close agreement with a recent analysis[2]. An update for GATK to v.3.0 was released after our initial validation experiments, but we also assessed its accuracy with a second blinded resequencing experiment (**Fig. 2b**, **Supplementary Note 3**, **Supplementary Fig. 3** and **Supplementary Tables 6** and **7**).

From the concordance rate alone, it is hard to judge the quality of indels unique to each pipeline, as these could represent either superior sensitivity or poor specificity. The size distribution of indels called by the HaplotypeCaller (v.2.4.3) had a bias toward deletions, whereas SOAPindel had a bias toward insertions (**Fig. 2b**). Scalpel and HaplotypeCaller (v.3.0) instead showed a well-balanced distribution in agreement with other studies of human indel mutations[8].

We further investigated the performance of the algorithms by a focused resequencing of a representative sample of indels using the more recent 250-bp Illumina MiSeq sequencing protocol (Online Methods). On the basis of the data depicted in **Figure 2a**, we selected a total of 1,000 indels according to the following categories: 200 random indels from the intersection of all pipelines, 200 random indels specific to the respective pipelines, and 200 random indels of size ≥30 bp from the union of all indels detected by the three algorithms.

To avoid possibly ambiguous representation, we 'left normalized'[2] indel positions. However, some ambiguity can still remain, especially within microsatellites, so we computed validation rates using two different approaches. These were (i) position based, in which an indel is considered valid if a mutation with the same coordinate exists in the validation data (**Fig. 3a**), and (ii) exact match, in which an indel is considered valid if there is a mutation with the same coordinate and sequence in the validation data (**Fig. 3b** and **Supplementary Data 1**).

As expected, indels detected by all pipelines had a high validation rate, and their sizes followed a log-normal distribution (**Supplementary Fig. 4**). However, the validation rate varied dramatically for each tool. Respectively, only 22% and 55% of the HaplotypeCaller (v.2.4.3)- and SOAPindel-specific indels could be validated even when the less strict position-based approach was used, whereas 77% of Scalpel-specific indels were true positive. For the long indels: less than 10% called by SOAPindel and HaplotypeCaller passed validation (**Fig. 3c** and **Supplementary Table 8**). The new version of GATK (v.3.0) has largely removed the bias toward deletions (**Fig. 2b**), but Scalpel still outperformed HaplotypeCaller (**Supplementary Note 3**). Scalpel showed a substantially higher validation rate (76%) for longer indels (>5 bp) than HaplotypeCaller v.3.0 (27%).

We further divided the results to separately report the validation rate for indels within microsatellites. SOAPindel showed an appreciably higher rate of false positives within microsatellites (**Fig. 3a**,**b**). When microsatellites were excluded, the performance of SOAPindel and HaplotypeCaller declined, whereas Scalpel's validation rate was only slightly reduced (**Fig. 3a**,**b**). The relative abundance of indels within microsatellites called by each tool is also shown (**Fig. 3a**,**b**); HaplotypeCaller seemed to filter against these. Finally, when we switched from the position-based approach to the exact-match approach, there was a notable reduction in the validation rate for indels within microsatellites. This phenomenon is due to their high instability and higher error rates, and in fact it is not unusual to have more than one candidate mutation at a microsatellite locus.

We further inspected the sequence composition of all false positive long indels. Specifically, we reanalyzed the 129 SOAPindel invalid long mutations using Scalpel. The majority of these mutations (115) overlapped repeat structures where the reference contained a perfect or near-perfect repeat (**Supplementary Fig. 5**). In contrast, of the 62 false positive long indels from HaplotypeCaller, only 16 overlapped a repeat. The remaining false positive deletions appear to be due to an aggressive approach used by the algorithm when processing soft-clipped reads (in which bases are 'trimmed'

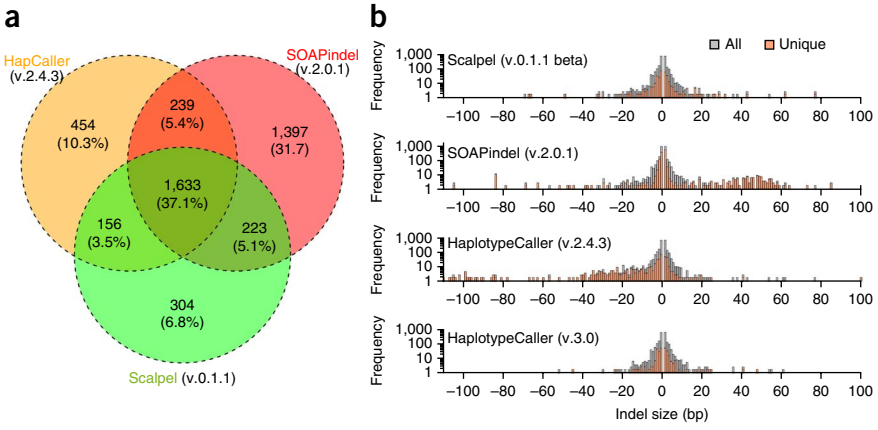

Figure 2 | Concordance of indels between pipelines. (**a**) Venn diagram showing the percentage of indels shared among the three pipelines. (**b**) Size distribution for indels called by each pipeline. 1,000 indels from five categories were analyzed by focused resequencing.

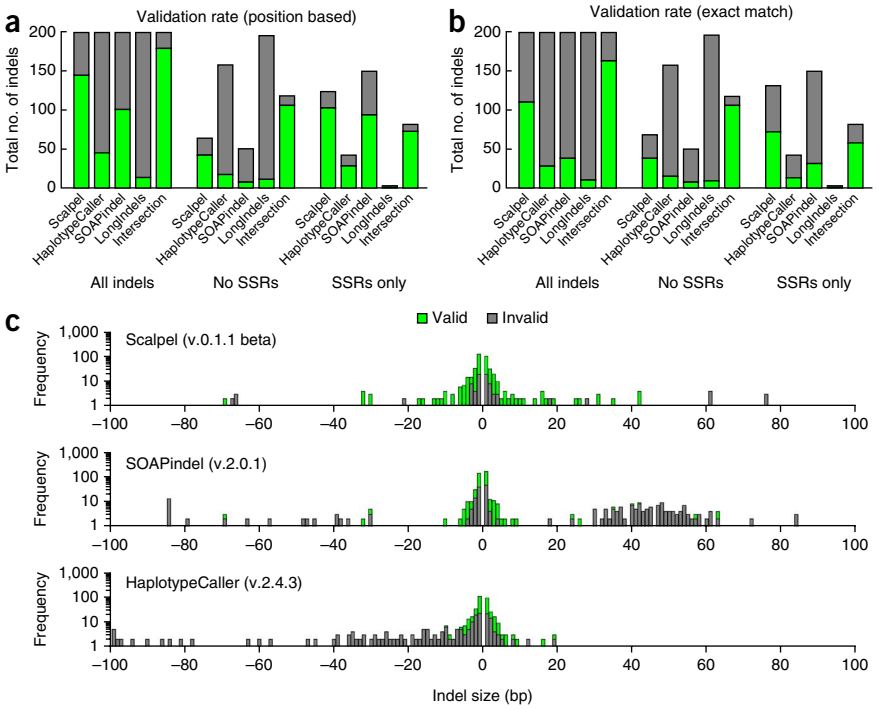

Figure 3 | MiSeq validation. (**a**,**b**) Ratio of valid and invalid indel calls based on position-based (**a**) or exact matches (**b**) for the indicated tools, for indels of size ≥30 bp from the union of the mutations detected by all three pipelines ("LongIndels") and for indels in the intersection ("Intersection"). Validation for all indels ("All indels"), validation for only the indels within microsatellites ("SSRs only") and validation for indels that are not within microsatellites ("No SSRs") are shown. (**c**) Stacked histogram of validation rate by indel size for each variant caller.

at the end of the read to optimize their alignment score). The soft-clipped reads in false positive indels for HaplotypeCaller were highly variable and are conjectured to be mapping artifacts of reads from different genomic locations (**Supplementary Fig. 6**). Finally, we investigated the relationship between the false discovery rate and characteristic features (for example, χ^2 score and coverage) for 614 indels detected by Scalpel and validated by resequencing (**Supplementary Note 4** and **Supplementary Figs. 7** and **8**). In addition to highlighting the common trends, this analysis provides recommendation on how to select a χ^2 score cutoff to achieve a given false discovery rate.

Using Scalpel we detected a total of 3.3 million indels in exomes from 593 families from the Simons Simplex Collection (SSC), corresponding to an average of ~1,400 (= 3,388,139/(4 × 593)) mutations per individual. Accounting for population frequencies, there were 27,795 distinct transmitted indels across the exomes. We found close agreement to the size distributions reported by Montgomery *et al.*[8] using low-coverage whole-genome data from 179 individuals (**Supplementary Fig. 9a**). Direct comparison to indels detected by the GATK-UnifiedGenotyper–based mapping pipeline used by Iossifov *et al.*[6] showed that Scalpel has superior power to detect longer insertions (**Supplementary Fig. 10**). To estimate Scalpel's ability to discover transmitted mutations, we performed targeted resequencing of 31 long (≥30 bp) transmitted indels. Excluding indels that failed to sequence (4), 21 passed validation (out of 27), which gives a 78% true positive rate. Three of the indels that did not pass validation were indeterminate with ambiguous alignments because they were either too long (≥70 bp) or embedded in a repetitive region.

Within the coding sequence, frame-preserving indels were more abundant than frameshifts (**Supplementary Fig. 9b**). In agreement with MacArthur and Tyler-Smith[21], we detected a large number of transmitted loss-of-function variants in protein-coding genes. Frameshift mutations were found at lower frequency in the population when located in protein-coding sequences compared to intronic regions (**Supplementary Fig. 9c**). Finally, we observed an enrichment of deletions over insertions (**Supplementary Table 9** and **Supplementary Fig. 11**), with an overall 2:1 ratio across all annotation categories. Similar trends were reported in previous studies[8,22].

We reanalyzed the data on autistic children and unaffected siblings[6,23,24] with Scalpel with the goal of examining *de novo* likely gene-disrupting (LGD) mutations. We confirmed an overabundance of frameshift mutations in autistic patients[6], predicted additional candidates and extended the analysis to a larger number of families. Our reanalysis of a previous study with 200 SSC families[23] reported an enrichment of 11 LGD indels in autistic children compared to 4 in their healthy siblings. In targeted resequencing of 102 candidate indels, we confirmed 84 as *de novo* mutations, invalidated 11 and failed to sequence 7, giving an 82% *de novo* positive predictive rate.

In order to focus the list of candidate genes, we excluded mutations that are common in the population and used stringent coverage filters (Online Methods) to select a total of 97 high-quality *de novo* indels (**Supplementary Data 2**). Even after extending the population size from 343 (ref. 6) to 593, the same 2:1 enrichment for LGD mutations was confirmed: 35 frameshifts in autistic children versus 16 in siblings ($P = 0.01097$, exact binomial test) (**Supplementary Tables 10** and **11**, **Supplementary Note 5** and **Supplementary Figs. 12–16**); other smaller studies have come to similar conclusions[23–25]. This result also holds for a larger collection of 1,303 SSC families (unpublished data, G.N. *et al.*). All together, in agreement with the previously reported results[6], we found a notable overlap between the LGD target genes and the 842 genes related to the protein FMRP[26], whose mutation is associated with autism. Specifically, 8 out of 35 LGDs in autistic children overlapped with the 842 FMRP-associated genes.

METHODS

Methods and any associated references are available in the online version of the paper.

Accession codes. Sequence Read Archive: SRX265476 (HiSeq data); SRX386284 (MiSeq data).

Note: Any Supplementary Information and Source Data files are available in the online version of the paper.

ACKNOWLEDGMENTS

The project was supported in part by the US National Institutes of Health (R01-HG006677) and US National Science Foundation (DBI-1350041) to M.C.S. and by the Cold Spring Harbor Laboratory (CSHL) Cancer Center Support Grant (5P30CA045508), the Stanley Institute for Cognitive Genomics and the Simons Foundation (SF51 and SF235988) to M.W. The DNA samples used in this work are included within SSC release 13. Approved researchers can obtain the SSC population data set described in this study by applying at https://base.sfari.org/. We thank S. Eskipehlivan for the technical assistance with the MiSeq validation experiments. We thank M. Bekritsky, S. Neuburgerand, M. Ronemus, D. Levy, B. Yamron and B. Mishra for helpful discussions and comments on the paper. We thank R. Aboukhalil for testing the software.

AUTHOR CONTRIBUTIONS

G.N. developed the software and conducted the computational experiments. G.N. and M.C.S. designed and analyzed the experiments. Y.W. assisted in designing the primers and performed the MiSeq validation experiments. J.A.O. designed the primers and analyzed the MiSeq data. H.F. and J.A.O. assisted with the computational experiments for the comparative analysis between different variant-detection pipelines. G.J.L. planned and supervised the experimental design for indel validation. Z.W. designed the primers and performed experiments for the validation of *de novo* and transmitted indels in the SSC. I.I., Y.-h.L. and M.W. assisted with the analysis of the SSC. G.N. and M.C.S. wrote the manuscript with input from all authors. All of the authors have read and approved the final manuscript.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

1. DePristo, M.A. *et al. Nat. Genet.* **43**, 491–498 (2011).
2. O'Rawe, J. *et al. Genome Med.* **5**, 28 (2013).
3. Zook, J.M. *et al. Nat. Biotechnol.* **32**, 246–251 (2014).
4. Mullaney, J.M., Mills, R.E., Pittard, W.S. & Devine, S.E. *Hum. Mol. Genet.* **19**, R131–R136 (2010).
5. Pearson, C.E., Edamura, N.K. & Cleary, J.D. *Nat. Rev. Genet.* **6**, 729–742 (2005).
6. Iossifov, I. *et al. Neuron* **74**, 285–299 (2012).
7. Li, H., Ruan, J. & Durbin, R. *Genome Res.* **18**, 1851–1858 (2008).
8. Montgomery, S.B. *et al. Genome Res.* **23**, 749–761 (2013).
9. Albers, C.A. *et al. Genome Res.* **21**, 961–973 (2011).
10. Ye, K., Schulz, M.H., Long, Q., Apweiler, R. & Ning, Z. *Bioinformatics* **25**, 2865–2871 (2009).
11. Karakoc, E. *et al. Nat. Methods* **9**, 176–178 (2012).
12. Li, Y. *et al. Nat. Biotechnol.* **29**, 723–730 (2011).
13. Li, H. *Bioinformatics* **28**, 1838–1844 (2012).
14. Li, S. *et al. Genome Res.* **23**, 195–200 (2013).
15. Iqbal, Z., Caccamo, M., Turner, I., Flicek, P. & McVean, G. *Nat. Genet.* **44**, 226–232 (2012).
16. Chen, K. *et al. Genome Res.* **24**, 310–317 (2014).
17. Li, H. *et al. Bioinformatics* **25**, 2078–2079 (2009).
18. Garrison, E. & Marth, G. Haplotype-based variant detection from short-read sequencing. Preprint at http://arxiv.org/abs/1207.3907v2 (2012).
19. Gymrek, M., Golan, D., Rosset, S. & Erlich, Y. *Genome Res.* **22**, 1154–1162 (2012).
20. Highnam, G. *et al. Nucleic Acids Res.* **41**, e32 (2013).
21. MacArthur, D.G. & Tyler-Smith, C. *Hum. Mol. Genet.* **19**, R125–R130 (2010).
22. Sjödin, P., Bataillon, T. & Schierup, M.H. *PLoS ONE* **5**, e8650 (2010).
23. Sanders, S.J. *et al. Nature* **485**, 237–241 (2012).
24. O'Roak, B.J. *et al. Nature* **485**, 246–250 (2012).
25. Neale, B.M. *et al. Nature* **485**, 242–245 (2012).
26. Darnell, J.C. *et al. Cell* **146**, 247–261 (2011).

ONLINE METHODS

The Scalpel pipeline. Scalpel is designed to perform localized microassembly of specific regions of interest in a genome with the goal of detecting insertions and deletions with high accuracy. It is based on the de Bruijn graph assembly paradigm, in which the reads are decomposed into overlapping *k*-mers and directed edges are added between *k*-mers that are consecutive within any read[27]. **Figure 1** shows the high-level structure of the pipeline. (1) The pipeline begins with a fast alignment of the reads to the reference genome using BWA[28,29]. Importantly, these alignments are not directly used to call variations but only to localize the analysis by identifying all the reads that have similarity to a given locus. Reads are then extracted in the region of interest (for example, exon) including: (i) well-mapped reads, (ii) soft-clipped reads, and (iii) reads that fail to map but are anchored by their mate. The latter two classes correspond to locations where the mapper encountered trouble aligning the reads, especially because of the large indels present, so it is necessary to include them in the assembly. (2) Once localized, the algorithm computes an on-the-fly assembly of the reads in the current region using the de Bruijn graph paradigm: specifically, reads are decomposed into overlapping *k*-mers (starting with a default $k = 25$), and the associated graph is constructed. (3) Using the reference sequence, one source node and one sink node are selected according to the procedure described later in the "Graph traversal" section. (4) An on-the-fly analysis of the repeats in each region is used to automatically select the *k*-mer size to be used for the assembly, described in section "Repeat analysis." (5) The graph is then exhaustively examined to find end-to-end paths that span the region. (6) After the sequences are assembled, they are aligned to the reference to detect candidate mutations using a sensitive gapped sequence aligner based on the Smith-Waterman algorithm[30] targeted at the reference window. Finally, the above assembly process is applied using a sliding-window approach over each target region. By default a window size of 400 bp is used with a sliding factor of 100 bp. The sliding-window strategy is fundamental to handle the highly nonuniform read distribution across the target (**Supplementary Fig. 17**). A window size of 400 bp is large enough to assemble the majority of the exons into a single contig: ~95% of the human exon-targets are shorter than 400 bp (**Supplementary Fig. 18**); however, each assembly task is small enough for using in-depth techniques to optimize the assembly.

Graph construction. Two critical components of the Scalpel algorithm are (i) construction of the de Bruijn graph and (ii) detection of sequence paths spanning the targeted region. Reads aligning to the region are extracted and decomposed into overlapping *k*-mers. In order to model the double-stranded nature of the DNA, a bidirected de Bruijn graph is constructed[31,32]. The graph is then compressed by merging all nonbranching chains of *k*-mers into a single node. Tips and low-coverage nodes are removed according to input threshold parameters to remove obvious sequencing errors. Note that, differently from a traditional de Bruijn graph assembler, Scalpel does not use any threading strategy to resolve collapsed repeats. In principle, threading would allow resolution of repeats whose lengths are between *k* and the read length. However, we observed in both real and simulated data that, owing to the localized graph construction, if a bubble were not covered end to end by the reads, threading would either disconnect the graph or introduce errors. Repeats are instead handled differently, as explained in the next section.

Repeat analysis. Due to the highly nonuniform read-depth distribution across the targeted region and the presence of near-perfect repeats that can mislead the assembly (**Supplementary Note 6** and **Supplementary Fig. 19**), Scalpel implements a detailed repeat composition analysis coupled with a self-tuning *k*-mer strategy. Specifically, when assembling a window, Scalpel inspects both the base-pair composition of the corresponding reference sequence and the resulting de Bruijn graph for the presence of cycles in the graph or near-perfect repeats in the assembled sequences. If a repeat structure is detected, the graph is discarded and a larger *k*-mer is selected. This process continues until a maximum *k*-mer length is reached, which is a function of the read length. If no *k*-mer value can be chosen to avoid the presence of repeats, the region is skipped and the next available region from the sliding window scheme is analyzed. This conservative strategy reduces the number of false positive calls in highly repetitive regions and skips less than 2% of possible windows in the human exome. Note also that, once *k* is selected by the self-tuning *k*-mer strategy, the graph is 'repeat free', and there is no need to use threading to resolve small repeats.

The proposed self-tuning *k*-mer strategy is similar to the dynamic approach used by SOAPindel and TIGRA to reconnect a broken path in low-coverage regions. However, SOAPindel searches for unused reads with gradually shorter *k*-mers until a path is formed or the lower bound on *k*-mer length has been reached; in TIGRA the user can specify the list of *k*-mers to use (by default only two: 15 and 25). Scalpel instead starts from a small *k*-mer value (input parameter) first and then gradually increases it such that the smallest possible *k*-mer value that generates a 'repeat-free' graph is used for each region. This strategy better handles repetitive sequences, highly polymorphic regions and sequencing errors: source and sink have a higher chance to be selected (see "Graph traversal"), and a smaller *k*-mer reduces the chance of fragmented assembly in low-coverage regions.

Graph traversal. Once a valid de Bruijn graph is constructed, Scalpel examines the graph to find end-to-end sequence paths that span the target window. Because the coverage from exome-capture data are highly variable, a special selection algorithm is used to find the edges of each window where coverage is present. First, two nodes in the graph are labeled as 'source' and 'sink' according to the following procedure: the reference sequence of the target region is scanned left to right to detect the first sequence of *k* bases that exactly matches one of the *k*-mers from the nodes in the graph; this node will be marked as the source. In a similar fashion the sink node is detected by scanning the reference sequence right to left. Because every region is first inspected for repeats, source and sink can be safely selected at this stage. The automated strategy used by Scalpel to select the boundaries of the reference sequence improves upon TIGRA's approach, in which the reference region is selected only on the basis of input parameters. After the source and sink nodes are identified, all possible source-to-sink paths are enumerated up to a maximum number (default: 100,000) using a depth-first search (DFS) traversal of the graph, similarly to the SUTTA assembly algorithm[33]. Note that because the regions to assemble are very small, time and space

computational complexities associated with large-scale whole-genome assembly are not relevant, and an exact brute-force strategy can be efficiently applied.

If there are no repeat structures in the graph, all the candidate paths are enumerated and aligned to the portion of the reference sequence delimited by source and sink *k*-mers using the standard Smith-Waterman-Gotoh alignment algorithm with affine-gap penalties. The list of candidate mutations is then generated. Under typical conditions, the assembler reports a single path for homozygous mutations and two paths for heterozygous mutations. For example, if the sample had an insertion in only one of the two haplotypes, the assembler would discover the indel and also the unmodified reference sequence. Note that a traditional sequence assembler would have selected only one of these two paths (usually with higher coverage) and discarded the other one. Scalpel instead examines both paths to distinguish: for example, between homozygous and heterozygous mutations. However, in practice, various factors in real data complicate the detection process, and sometimes multiple paths are reported in the case of more exotic variations. For example, the Illumina sequencing platform is particularly error prone around microsatellites (for example, homopolymer runs) and, as a consequence, multiple candidate alleles are elucidated by the data at these loci. Highly polymorphic regions are also prone to generate multiple paths and could be computationally demanding: if the distance between multiple nearby mutations is larger than the (automatically) selected *k*-mer value, each of the associated bubbles in the graph will give rise to two different paths. Finally, it is important to note that SNVs are also computed by Scalpel, but they are not reported in output. SNVs are used internally for important downstream analysis of the variants called in order: for example, to compute coverage around the indels and to correctly characterize the zygosity of the mutations.

Exome-capture data. Exome capture for the sample K8101-49685 was carried out using the Agilent 44-Mb SureSelect protocol and then sequenced on Illumina HiSeq2000 with average read length of 100 bp. More than 80% of the target region was covered with a depth of 20 reads or more. All of the HiSeq data have been submitted to the Sequence Read Archive (http://www.ncbi.nlm.nih.gov/sra/) under project accession number SRX265476.

MiSeq validation. A total of 1,400 indels were selected for MiSeq validation over the course of this study. MiSeq validation was initially performed on 1,000 indels before the release of GATK v.3.0 (see the Results section for detailed selection criteria). After the release of GATK v.3.0, we selected an additional 400 indels for MiSeq validation (see **Supplementary Note 3** for detailed selection criteria). Out of these 400 indels, 215 were covered with more than 1,000 reads in the initial MiSeq data set or in another MiSeq data set reported by O'Rawe *et al.*[2]. Thus, the second MiSeq validation experiment was performed on the remaining 185 indels. For both of the MiSeq validation experiments performed during the course of this study, PCR primers were designed using Primer 3 (http://primer3.sourceforge.net/) to produce amplicons ranging in size from 200 to 350 bp, with indels of interest located approximately in the center of each amplicon. Primers were obtained from Sigma-Aldrich in 96-well mixed-plate format, 10 μmol/L dilution in Tris per oligonucleotide. Upon arrival, all primers were tested for PCR efficiency using a HapMap DNA sample (catalog ID NA12864, Coriell Institute for Medical Research) and LongAmp *Taq* DNA polymerase (New England Biolabs). PCR products were visually inspected for amplification efficiency using agarose gel electrophoresis. For the validation experiment, this same PCR was performed using sample K8101-49685 genomic DNA as template. PCR product was verified on E-Gel 96 gels (Invitrogen) and subsequently pooled for ExoSAP-IT (Affymetrix) cleanup. The cleanup product was further purified using QIAquick PCR Purification Kit (Qiagen) and quantified by Qubit dsDNA BR Assay Kit (Invitrogen). Library construction for the MiSeq Personal Sequencer platform (Illumina) was performed according to the Illumina TruSeq DNA Sample Prep LS protocol (for the initial 1,000 indels) and TruSeq Nano DNA Sample Preparation Guide (for the additional 185 indels), omitting the DNA fragmentation step. Finally, before we loaded the sample onto the MiSeq machine, the quality and quantity of the sample was again verified using the Agilent DNA 1,000 Kit on the Agilent Bioanalyzer and with quantitative PCR (Kapa Biosystems). This protocol generated high-quality 250-bp reads (paired end) with an average coverage of 47,018× (**Supplementary Fig. 20**). The reads were aligned with BWA-MEM (v.0.7.5a) to the reference human genome hg19. The alignment was sorted with SAMtools (v.0.1.18), and PCR duplicates were marked with the Picard tool set (v.1.91). Indels were realigned with the GATK (version v.2.6-4) using the IndelRealigner, and base quality scores were recalibrated. Variants were then called with GATK UnifiedGenotyper. All of the MiSeq data have been submitted to the Sequence Read Archive under project accession number SRX386284.

Alignment. Sequencing reads from K8101-49685 exome-capture data were aligned using BWA (v.0.6.2-r126) with default parameters to the human reference hg19. Alignments were converted from SAM format to sorted, indexed BAM files with SAMtools (v.0.1.18). The Picard tool set (v.1.91) was used to remove duplicate reads. These BAM files were used as input for all the indel callers used in this study. Reads coming from the resequencing experiments were also aligned using BWA. However, if the indel approaches half the size of the read length, even after target resequencing, mapping the reads containing the indel is problematic. The problem is emphasized if the indel is located toward the ends of the read (instead of in the middle). To avoid this problem we aligned sequencing reads containing long indels (≥30 bp) using Bowtie2 (ref. 34) instead of BWA. Bowtie2 offers an end-to-end alignment mode that searches for alignments involving all of the read characters, also called an 'untrimmed' or 'unclipped' alignment. Specifically, we used the following parameter settings: "--end-to-end --very-sensitive --score-min L,-0.6,-0.6 --rdg 8,1 --rfg 8,1 --mp 20,20".

Variant calling. Indels for K8101-49685 were called using Scalpel, GATK HaplotypeCaller and SOAPindel as follows.

Scalpel. Scalpel (v.0.1.1 beta) was run on the indexed BAM using the following parameter setting: "--single --lowcov 1 --mincov 3 --outratio 0.1 --intarget". Indels showing high coverage unbalance were then removed (χ^2 *k*-mer score >20).

GATK. GATK software tools (v.2.4-3 and v.3.0) were used for improvement of alignments and genotype calling and refining with recommended parameters. BAM files were realigned with

doi:10.1038/nmeth.3069

the GATK IndelRealigner, and base quality scores were recalibrated by the GATK base quality recalibration tool. Genotypes were called by the GATK UnifiedGenotyper and HaplotypeCaller. According to the GATK recommendations, the Variant Quality Score Recalibration (VQSR) was not used for the K8101-49685 single-exome experiment. Also, hard filtering criteria (such us "QD < 2.0, ReadPosRankSum < -20.0 FS > 200.0") were not used for v.2.4.3 as they were aggressively removing long indels from the HaplotypeCaller calls (**Supplementary Note 3**), but they were used instead for the more recent HaplotypeCaller from GATK v.3.0. It is important to report that the bias toward deletions for HaplotypeCaller has been extensively reduced with the release of GATK 2.8 and 3.0 in December 2013 and March 2014 after our initial MiSeq resequencing experiments were completed (**Supplementary Note 3**). However, many research groups have already employed the older version of HaplotypeCaller for genetic studies and are still extensively using it. We expect GATK, along with our own and other algorithms, to improve over time as new insights are made into the mutation mechanisms and error profiles.

SOAPindel. SOAPindel (v.2.0.1) was run on the indexed BAM file using default parameters. According to SOAPindel documentation, putative indels are initially assumed to be located near the unmapped reads whose mates mapped to the reference genome. SOAPindel then executes a local assembly (*k*-mer = 25 by default) on the clusters of unmapped reads. The assembly results were aligned to reference in order to find the potential indels. To distinguish true and false positive indels, SOAPindel generates Phred quality scores, which take into consideration the depth of coverage, indel size, number of neighboring variants, distance to the edge of the contig, and position of the second different base pair. Only those indels filtered by internal threshold are retained in the final indel call set.

Finally, for all pipelines, we selected only indels located within the regions targeted by the exome-capture protocol.

The Simons Simplex Collection. The Simons Simplex Collection[35] (SSC) is a permanent repository of genetic samples from 2,700 families operated by SFARI (http://sfari.org/) in collaboration with 12 university-affiliated research clinics. Each simplex family has one child affected with autism spectrum disorder and unaffected siblings. Each genetic sample also has an associated collection of phenotype measurements and assays. The results presented in this work are based on a subset of the SSC composed of 593 families (2,372 individuals). Specifically, this subset of the SSC collection corresponds to families that have been examined in three recent studies: 343 families from Iossifov *et al.*[6] (CSHL), 200 families from Sanders *et al.*[23] (Yale) and 50 families from O'Roak *et al.*[24] (University of Washington). We selected only family units of four individuals (father, mother, proband, one unaffected sibling), referred to as "quads," for all analyses in this study. All available SSC quad samples were evaluated using the criterion described in Iossifov *et al.*[6]. Analysis was supervised under the CSHL IRB review committee. SFARI maintains the consent of all individuals in the SSC.

Analysis of *de novo* indels related to autism. After eliminating all candidate positions that are common in the population, and thus unlikely to be related to the disorder, we reassembled each region associated with the candidates indels across the family members using a more sensitive parameter setting for Scalpel. Specifically, we reduced the starting *k*-mer value to 10 and turned off the removal of low-coverage nodes. This step was important to adjust for possible allele imbalance favoring the reference allele over the mutation in the parents but was impractical to do initially for the whole collection: lowering the *k*-mer and keeping all the nodes in the graph substantially increase the computation complexity of the algorithm. Then we selected *de novo* indels with a χ^2 *k*-mer score ≤10.84. The χ^2 *k*-mer score is computed using the standard formula for the chi-square test statistics (χ^2) but applied to the *k*-mer coverage of the reference and alternative alleles for the mutation according to the following formula:

$$\chi^2 = \frac{(C_o^R - C_e^R)^2}{C_e^R} + \frac{(C_o^A - C_e^A)^2}{C_e^A}$$

where C_o^R and C_o^A are the observed *k*-mer coverage for the reference and alternative alleles, respectively, and C_e^R and C_e^A are the expected coverage such that $C_e^R = C_e^A = \text{totCov}/2$. Finally, we enforced parents to have at least a *k*-mer coverage of 15 over the assembled region. The final list of mutations obtained after applying these filters was then analyzed for enrichment tests between autistic and non-autistic individuals and FMRP enrichment tests using the methods established in Iossifov *et al.*[6].

System requirements and software availability. Scalpel is written in C++ and Perl. The source code is freely available as an open-source software project on the SourceForge website at http://scalpel.sourceforge.net/. It usually takes 2–3 h to process one exome-capture data set (80% of target at ≥20×) using ten cores and requiring a minimum of 3 GB of RAM.

27. Nagarajan, N. & Pop, M. *Nat. Rev. Genet.* **14**, 157–167 (2013).
28. Li, H. & Durbin, R. *Bioinformatics* **26**, 589–595 (2010).
29. Li, H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. Preprint at http://arxiv.org/abs/1303.3997v1 (2013).
30. Smith, T.F. & Waterman, M.S. *J. Mol. Biol.* **147**, 195–197 (1981).
31. Medvedev, P., Georgiou, K., Myers, G. & Brudno, M. *Lect. Notes Comput. Sci.* **4645**, 289–301 (2007).
32. Jackson, B.G. & Aluru, S. in *37th Int. Conf. Parallel Process.* 346–353 (ICPP, 2008).
33. Narzisi, G. & Mishra, B. *Bioinformatics* **27**, 153–160 (2011).
34. Langmead, B. & Salzberg, S. *Nat. Methods* **9**, 357–359 (2012).
35. Fischbach, G.D. & Lord, C. *Neuron* **68**, 192–195 (2010).