Title
novoBreak: local assembly for breakpoint detection in cancer genomes

Journal
Nature Methods, 14(1)

ISSN
1548-7091

Authors
Chong, Zechen
Ruan, Jue
Gao, Min
et al.

Publication Date
2017

DOI
10.1038/nmeth.4084

Peer reviewed

HHS Public Access

Author manuscript

Nat Methods. Author manuscript; available in PMC 2017 May 28.

Published in final edited form as:

Nat Methods. 2017 January ; 14(1): 65–67. doi:10.1038/nmeth.4084.

novoBreak: local assembly for breakpoint detection in cancer genomes

Zechen Chong[1,8], **Jue Ruan**[2,8], **Min Gao**[3,8], **Wanding Zhou**[1], **Tenghui Chen**[1], **Xian Fan**[1], **Li Ding**[4], **Anna Y. Lee**[5], **Paul Boutros**[5,6,7], **Junjie Chen**[3], and **Ken Chen**[1,9]

[1]Department of Bioinformatics and Computational Biology, The University of Texas MD Anderson Cancer Center, Houston, TX, USA

[2]Agricultural Genomics Institute at Shenzhen, Chinese Academy of Agricultural Sciences, Shenzhen, China

[3]Department of Experimental Radiation Oncology, The University of Texas MD Anderson Cancer Center, Houston, TX, USA

[4]McDonnell Genome Institute, Washington University, St. Louis, MO, USA

[5]Informatics and Biocomputing Program, Ontario Institute for Cancer Research, Toronto, Ontario, Canada

[6]Department of Medical Biophysics, University of Toronto, Toronto, Ontario, Canada

[7]Department of Pharmacology & Toxicology, University of Toronto, Toronto, Ontario, Canada

Abstract

We present novoBreak, a novel genome-wide local assembly algorithm that discovers somatic and germline structural variation breakpoints in whole genome sequencing data. In the ICGC-TCGA DREAM 8.5 Somatic Mutation Calling Challenge and real cancer genome data analysis, novoBreak consistently outperformed existing algorithms due largely to more effective utilization of reads spanning breakpoints. NovoBreak also demonstrated great sensitivity in identifying short INDELs. The source code is available at http://sourceforge.net/projects/novobreak/.

[9]Corresponding author: Ken Chen (kchen3@mdanderson.org).
[8]These authors contributed equally

Accession codes.

NCBI Sequence Read Archive: SRP042948, SRP028176.
European Genome-phenome Archive: EGAD00000000055, EGAS00001000255.

AUTHER CONTRIBUTIONS

Z.C., J.R. and K.C. conceived the algorithm. Z.C. developed the software. Z.C. and K.C. designed and analyzed the experiments. M.G. and J.C. designed and performed the validation experiments. W.Z. designed the scoring statistics. M.G., T.C., X.F., L.D., A.Y.L. and P.B. tested the algorithm and performed additional analyses. K.C. supervised the projects. Z.C. and K.C. wrote the manuscript with input from all authors. All authors have read and approved the final manuscript.

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

Somatic structural variations (SVs) are major driving forces of tumor development and progression. Sporadic and recurrent chromosomal aberrations have been observed in most cancer types[1]. Many are desirable therapeutic targets.

The advent of high-throughput next generation sequencing (NGS) technologies has made it possible to perform genome-wide detection of SVs at base pair resolution. As a result, unprecedented landscapes of SVs have been unveiled in various cancer genomes[2]. However, computational approaches[3] for detecting SVs from NGS data are limited in sensitivity and comprehensiveness[4].

One approach is to align short paired-end reads to a reference genome and identify signals in discordant read pairs[5], read depths[6], split reads[7], or their combinations[8]. Another approach is through targeted local assembly of aligned and partially aligned reads in candidate SV regions discovered a priori[9]. These approaches depend heavily on the accuracy of read alignments, which is often limited for reads spanning breakpoints or substantially different from the reference. In comparison, whole genome assembly approaches[10] are less biased. However, assembling a whole genome is computationally intensive[11] and results are often affected by repeats, polyploidy, read length and sequencing coverage.

We developed a novel method, novoBreak, which obtains genome-wide local (glocal) assembly of breakpoints from clusters of reads sharing a set of *k*-mers (contiguous nucleotide sequences of length *k*) uniquely present in a subject genome (*e.g.*, a tumor genome) but not in the reference genome or any control data (*e.g.*, a matched normal genome) (Fig. 1 and **Online Methods**). When applied to somatic breakpoint detection from matched tumor and normal tissue data, novoBreak first constructs a hash table from the tumor reads, containing all the *k*-mers, their host reads and frequencies in the set (**Online Methods** and Supplementary Note 1). Next, it filters out *k*-mers representing reference alleles or sequencing errors, and retains those representing variants or novel sequences not present in the reference genome. It then queries the normal reads and further classifies the *k*-mers into 1) germline *k*-mers, those present in both the tumor and the normal genome, and 2) somatic *k*-mers, those present in the tumor but not the normal genome. Then, novoBreak identifies clusters of read pairs spanning each somatic breakpoint, and assembles each cluster of reads into contigs (**Online Methods** and Supplementary Note 2). By comparing the resulting high-quality contigs with the reference, novoBreak identifies breakpoints and associated SVs. Finally, novoBreak quantifies the amount of the supporting evidences at each breakpoint and outputs a final report.

We examined the performance of novoBreak in the ICGC-TCGA DREAM 8.5 Somatic Mutation Calling Challenge (https://www.synapse.org/#!Synapse:syn312572), which aimed to identify the best algorithms for detecting somatic mutations in NGS data[12]. In each of the synthetic sub-challenges, a high coverage (60–80×) whole genome sequencing (WGS) bam file produced from a cell line or a patient tissue was divided into two parts (30–40× each). One part was treated as the normal data and the other as the tumor data containing mutations spiked-in by BAMSurgeon[13]. A series of *in silico* sub-challenges (IS) were implemented with increasing numbers, types of variants and cellular complexity. In total, 204 submissions were made by 27 teams that developed the most widely used SV detection tools such as

Breakdancer[5], Delly[8], Pindel[7], and Manta (https://github.com/StructuralVariants/manta). NovoBreak consistently achieved the best balanced accuracy (sensitivity and precision) in IS2, IS3 and IS4 (Supplementary Table 1). Almost all the top performing tools achieved high precision (>0.98) after stringent filtering. NovoBreak excelled in higher sensitivity, which was particularly evident when insertions were introduced in IS2 and IS3.

IS3 was probably the most difficult because it not only contained SNVs and four types of SVs: deletions, duplications, inversions and insertions (mobile elements), but also INDELs (insertions and deletions shorter than 100 bp). It also contained subclones at respective cellular fractions of 50%, 33%, and 20%. NovoBreak achieved the highest balanced accuracy of 0.892 (sensitivity: 0.801 and precision: 0.984) due mainly to its higher sensitivity in detecting insertions (Fig. 2a). It discovered 100 (4.3%) and 120 (5.1%) more insertions in the ground truth than DELLY and Manta, respectively. Compared to alignment-based approaches, novoBreak more effectively utilized reads spanning insertion breakpoints (Supplementary Fig. 1). Further analysis of the SVs missed by DELLY and Manta indicates that novoBreak performs better in low coverage regions with few discordantly paired or split reads (Supplementary Note 3). Breakpoints identified by novoBreak also had the highest precision: 98.9% are within −2bp to 2bp relative to the ground truth (Fig. 2b).

Detection of INDELs, particularly insertions, is challenging because of difficulties in achieving accurate short-read alignment. NovoBreak ranked 2nd and 1st in IS3 and IS4, respectively (Supplementary Table 2). IS4 was particularly difficult for INDELs and SNVs due to three times more simulated events than in the previous challenges, including sub-clonal events at relatively low allelic frequencies (15%). Encouragingly, novoBreak achieved a balanced INDEL detection accuracy of 0.857 (sensitivity: 0.788 and precision: 0.926), close to the best SNV detection accuracy on the leaderboard. After comparison with the ground truth, we found that novoBreak discovered a higher fraction of INDELs in almost every size range than GATK-HaplotypeCaller[14] (balanced accuracy: 0.364, sensitivity: 0.499 and precision: 0.229) and Strelka[15] (balanced accuracy: 0.626, sensitivity: 0.601 and precision: 0.650) under the default parameters and filters (Fig. 2c). GATK-HaplotypeCaller had significantly lower sensitivity in detecting 1, 2, and 3 bp INDELs, due likely to limitations of aligning short-reads and stringent filtering. In contrast, Strelka demonstrated reduced sensitivity as INDEL size increased. It did not report any insertion longer than 25 bp.

We compared novoBreak with BreakDancer[5] (v1.1.2), DELLY[8] (v0.6.3) and Fermi[16] (v1.1-r751-beta) using the WGS data from COLO-829, a melanoma tumor cell line[17]. Fermi is a string-graph-based whole genome assembler that retains contigs containing SNPs, INDELs and SVs. Because Fermi does not come with a ready-to-use tool to call SV breakpoints, we used the SV-calling steps of novoBreak to evaluate its assembly results. These data were previously analyzed by a read pair approach[17] and CREST[18]. In total, 48 SV breakpoints have been previously validated via polymerase chain reaction (PCR) and Sanger sequencing (Supplementary Table 3). We used these 48 breakpoints as ground truth to benchmark these tools. Under default parameters, BreakDancer identified 37 true positives (TPs), with a total of 14,340 predicted; DELLY, 34 TPs, with 1,113 predicted; and Fermi, 40 TPs, with 16,849 predicted. A large fraction of SVs reported by these tools were likely germline, instead of

false SVs. In contrast, novoBreak identified 44 TPs with 78 breakpoints predicted (Fig. 2d and Supplementary Table 4). Of the 4 missing TPs, two were missed by all the tools and 2 could be recovered by novoBreak at less stringent settings. We designed PCR primers around the 34 novel breakpoints and validated 9 (Supplementary Table 5). The remaining ones were not necessarily false calls and could be attributed to deficiency in validation experiments or evolution of the cultured cell line. Indeed, 19 (57.6%) of the 34 calls were also predicted by at least one other tool. These results demonstrated novoBreak's high sensitivity and specificity in analyzing real tumor data under default settings. Users can adjust the filtering parameters to obtain different sensitivity and specificity tradeoff in different applications.

To further evaluate the sensitivity of novoBreak on cancer patient data, we analyzed the WGS data of a patient with low-grade glioma (Supplementary Note 4) and those of 22 invasive breast carcinoma samples in The Cancer Genome Atlas (TCGA). This set of TCGA samples was analyzed previously by INTEGRATE[19], which integrates matched whole genome and whole transcriptome sequencing (WTS) data to discover gene fusions. Overall, novoBreak identified 1,628 deletions, 1,724 duplications, 2,335 inversions and 1,982 translocations, equivalent to 349 SVs per sample (Supplementary Table 6 and 7). It identified 104 (86.7%) of the 120 known high-confidence gene fusions[19] (Supplementary Table 8). The true sensitivity was probably higher because 19% of the known SVs were likely false positives[19]. In addition, they were identified using both WGS and WTS data; whereas novoBreak only examined the WGS data.

We present a new algorithm, novoBreak, for detecting structural variation breakpoints in subject genomes. The most significant improvement of novoBreak compared to other approaches is the *k*-mer identification, filtering and classification strategy, which substantially narrows down the number of putative SV breakpoints and focuses computational power on the most informative portion of the data. By clustering and performing local assembly around breakpoints, novoBreak takes full advantage of unmapped reads and/or partially mapped reads. The scoring and filtering strategy of novoBreak provides high precision in the final results. The *k*-mer targeted assembly framework exemplified by novoBreak will facilitate comprehensive, sensitive, efficient, and accurate identification of novel sequence alterations in genomic, exomic and transcriptomic sequencing data. A caveat of novoBreak is that it misses SV breakpoints in repetitive sequences longer than 2*k*-1 bp. Further versions of novoBreak with increased *k* will alleviate this limitation. The source code of novoBreak (Supplementary Software) is freely available for academic use at http://sourceforge.net/projects/novobreak/.

Online Methods

The novoBreak pipeline

novoBreak is developed to comprehensively discover exact chromosomal breakpoints introduced by structural variations in genomes or transcriptomes. It is based on 1) a genome-wide classification and filtering strategy, which identifies specific nucleotide signatures (novo *k*-mers) and 2) a local assembly approach, which constructs breakpoint sequences from reads containing the novo *k*-mers. The workflow of novoBreak consists of the

following steps (Figure 1): (1) novoBreak begins with an indexing and filtering procedure to obtain "novo *k*-mers" and associated short reads, described in the section "indexing and filtering *k*-mers". (2) Paired-end reads containing the same set of novo *k*-mers are clustered together. Each cluster contains read pairs covering the same breakpoint. An assembly algorithm is then applied to each cluster to construct a breakpoint spanning sequence. The clustering and local assembly step is described later in the section "Clustering and local assembly algorithm". (3) After short reads are assembled in each cluster, the resulting contigs are aligned to the reference using BWA-MEM[20] (Supplementary Note 2) with '-M' option to obtain secondary alignments. The alignment results are parsed to infer breakpoints and the associated SVs. For short SVs, such as INDELs, novoBreak directly parse the Compact Idiosyncratic Gapped Alignment Report (CIGAR) strings of the aligned contigs. For large SVs, novoBreak will consider both the primary and the secondary alignments of each contig. In current implementation, novoBreak predicts deletions, insertions, inversions, duplications and translocations at base pair resolution. (4) To achieve a high precision, novoBreak employs a scoring and filtering module, as described in the section "scoring method".

Indexing and filtering *k*-mers

Given a sequence S of length L, a *k*-mer is a length k ($k < L$) substring of sequence S. We notice that if a read R contains a breakpoint of a structural change with respect to the reference or the normal genome of a cancer patient, there are at most *k*-1 *k*-mers ($k < |R|$) covering the breakpoint. The default k is 31 (Supplementary Note 1) in novoBreak. We define these *k*-mers as "novo *k*-mers" because they contain novel sequence information specific to the subject. In a tumor-normal paired cancer genome sequencing study, the novo *k*-mers contain the somatic breakpoints that specifically exist in the tumor but not in the paired normal sample. The first critical step of novoBreak is to obtain the novo *k*-mers. An effective approach is to implement a hash table that first indexes and loads all the *k*-mers in all the reads in the tumor sample into the memory, and then eliminate *k*-mers that are present in the reference or the normal genome. The remaining high frequency *k*-mers should contain genuine somatic breakpoints, including SNVs, small indels and large SVs. This approach is computationally feasible for whole exome or whole transcriptome analysis. But for high coverage whole genome analysis, the memory cost is extremely high (usually a few hundred gigabytes) due mainly to the presence of sequencing errors. A critical component of novoBreak is to reduce memory consumption. For whole genome sequencing data, instead of indexing the sequenced reads, novoBreak starts from hashing all the *k*-mers in the reference genome. Then, it adopts a two-pass approach to calculate novo *k*-mers in the sequenced genomes. The first pass is to scan every reads and mark the status (presence/absence) of each constituent *k*-mer in the reference genome using the pre-constructed hash table. In the process, novoBreak automatically trims off error prone ends in low quality reads (Supplementary Note 1). novoBreak uses a bit array data structure to mark a read. If a *k*-mer in a read is in the hash table, it will be marked as 1 (otherwise 0) in the corresponding bit in the bit array. When all the reads are processed, the hash table for the reference *k*-mers is released. Next, novoBreak goes through the reads containing at least one 0 bit to obtain the minimal occurrence of the non-reference *k*-mers. novoBreak adopts Bloom filter[21], a probabilistic data structure that tests whether a given element is in a set. A Bloom filter is a

bit array of *m* bits, initialized to be 0. *k* different hash functions are applied to an element and map the element to *k* different positions in the array. To add an element, these *k* positions will be set to 1. To test whether an element is in the set, each of the k positions will be examined. If there is a 0 at any of the *k* positions, the element is definitely not in the set. If all the *k* positions are 1, then either the element is in the set or the positions were coincidently set to 1 by other elements. Such false positive (FP) errors could happen because different elements could be coincidently hashed to same positions in the bit array.

Fortunately, the chance of having an error is very small, less than $(1-e^{-\frac{k(n+0.5)}{m-1}})^k$, where *n* is the total number of elements, *m* is the size of the bit array of the Bloom filter and *k* is the number of hash functions. Note these rare FP errors do not hurt sensitivity and have negligible possibility of introducing false positive breakpoints, due to the subsequent read clustering, assembly, alignment and variant calling steps. We expand the above standard Bloom filter from one bit to two or more (default to 3 bits in novoBreak) to count if a *k*-mer has occurred more than a minimal number of times (default 3 in novoBreak) (Supplementary Note 1) in the dataset. Thus, *k*-mers introduced by sequencing errors will be automatically disregarded and the remaining are novo *k*-mers from the variant alleles. For somatic analysis, novoBreak will further scan the normal control reads using a hash table and counts the occurrence of these *k*-mers in the normal reads. Based on these counts, candidate somatic *k*-mers (i.e., *k*-mers only present in the tumor but not the normal sample) can be identified, with the effect of cross-contamination between the samples being accounted for. Finally, novoBreak loads read pairs containing the candidate somatic *k*-mers and automatically removes duplicated read pairs that have identical sequences in both reads.

Clustering and local assembly

With novo *k*-mers and the associated read pairs identified, a straightforward method is to assemble all the read pairs directly. However, the cost of assembly is still very high due to a large number of reads. In addition, presence of alternative alleles, repeats and sequencing errors can easily cause misassemblies. Note that, as shown in (Supplementary Fig. 2), at each breakpoint, there are *k*-1 novo *k*-mers with many reads covering it. Reads covering the same breakpoint share a subset of the *k*-1 novo *k*-mers. Based on this pair-wise relationship between *k*-mers and reads, we can find the set of read pairs covering a breakpoint using a union-find algorithm [22], which identifies all the connected components in an undirected graph consisting of reads and *k*-mers (as nodes) and their connections (as edges). To avoid having large clusters with many reads due to repeats or sequencing errors, novoBreak trims the connected components based on read and *k*-mer statistics. For the purpose of detecting SVs, the computational cost is further reduced by directly reading from bam files and correcting base errors based on high quality aligned reads.

After clustering, it is relatively easy to locally assemble the read pairs in each cluster, since the number of read pairs is small and they originate from the same locus of an allele. Almost every modern assembler can be applied for such a task. novoBreak pipeline uses SSAKE[23] (Supplementary Note 2) to assemble read pairs into contigs. The setting of SSAKE in novoBreak is “-p 1 -k 2 -n 1 -m 16 -x 3 -w 1 -z 30 -o 1”.

SSAKE can generate multiple contigs from each cluster. Each contig is aligned by BWA-MEM and analyzed independently. After all the candidate breakpoints are generated, novoBreak merges them and creates a unique set of SVs.

Somatic SV Scoring methods

novoBreak scores and ranks each predicted breakpoint based on assembly and mapping results. At a given locus, novoBreak calculates a statistical quality score

$$Q=-10log_{10}(\frac{\Pr\{D|G=0,2\}}{\Pr\{D|G=1\}}),$$

where $D=\{D_{I,R}\}$ comprises of the counts of read pairs supporting the reference allele ($R=r$) and those supporting the variant alelle ($R=v$) from the tumor ($I=t$) and the normal ($I=n$) data, respectively; $G=0,1,2$ indicates whether the locus has a reference (no SV in either tumor or normal), somatic (SV only in tumor) or germline (SV in both tumor and normal) status.

We can compute the likelihood of the data, given the status of a locus. For example, likelihood of the somatic status $G=1$ can be estimated as:

$$\Pr\{D|G=1\}=\prod_{I=t,n;R=r,v}\Pr(D_{I,R}|G=1).$$

Because the variant allele fraction in the tumor is unknown, novoBreak uses a beta-binomial distribution to estimate the likelihood of the observed read counts. For example, the number of read pairs supporting a breakpoint in the tumor sample is

$$\Pr(D_{t,v}|G)=\Pr(D_{t,v}|D',\alpha_{t,G},\beta_{t,G})=\begin{pmatrix} D' \\ D_{t,v} \end{pmatrix} B(D_{t,s}+\alpha_{t,G},D'-D_{t,G}+\beta_{t,G})/B(\alpha_{t,G},\beta_{t,G}),$$

where $\alpha_{I,G}$ and $\beta_{I,G}$ indicate the parameters used for the combinations among $I\in\{t,n\}$ and $G\in\{0,1,2\}$. For the somatic status $G=1$, novoBreak initializes $\alpha_{n,1}=1$, $\beta_{n,1}=10$ and $\alpha_{t,1}=1$, $\beta_{t,1}=1$ to reflect the concept that SV signal in the normal sample is largely due to noise. For the germline status $G=2$, novoBreak uses a uniform distribution $\alpha_{n,2}=\beta_{n,2}=\alpha_{t,2}=\beta_{t,2}=1$. For the reference status $G=0$, $\alpha_{n,0}=\alpha_{t,0}=10$ and $\beta_{n,0}=\beta_{t,0}=1$ were used. D' represents the total number of read pairs at the locus with non-zero chances of spanning the breakpoint.

Minor empirical adjustment of these scores and parameters was applied to account for variations introduced by assembly quality, mapping quality, tissue purity, sampling bias, and SV size and type.

Indel analysis

Indels detection on the IS4 data was performed by novoBreak, GATK-HaplotypeCaller and Strelka as follows.

novoBreak—novoBreak (v1.03) was run under the parameters '-k31 -m2'. All the assembled contigs and unassembled short read pairs containing the novo *k*-mers were mapped to the reference using BWA-MEM[20]. The alignment results were sorted and the coordinates of indels were adjusted using SortSam of Picard (v1.107) (http://broadinstitute.github.io/picard/) and LeftAlignIndels of GATK[14, 24, 25] (v2.8-1), respectively. The CIGAR strings of the alignment results were parsed to generate an indel list (in VCFv4.1 format). Indels were further filtered using Database of Single Nucleotide Polymorphisms dbSNP (Build ID: 138, Available from: http://www.ncbi.nlm.nih.gov/SNP/) and low complexity regions identified with the mdust program (http://compbio.dfci.harvard.edu/tgi/). Finally, only indels with allele fraction greater than 1% were selected.

GATK-HaplotypeCaller—GATK v2.8-1 was run on the same data. First, tumor and normal bam files were realigned using IndelRealigner and left-aligned using LeftAlignIndels. HaplotypeCaller was run on tumor and normal bam files with the parameters '--genotyping_mode DISCOVERY-stand_emit_conf 10-stand_call_conf 30′, respectively. Then, SelectVariants of GATK was run with parameter '-selectType INDEL' to generate an indel VCF file for the tumor and the normal. Indels from the tumor and the normal samples were further filtered using VariantFiltration with parameters '--filterExpression "QD < 2.0 || FS > 200.0 || ReadPosRankSum < −20.0"'. Finally, only indels in the tumor VCF file but not in the normal VCF file with "PASS" labels were evaluated as somatic indels.

Strelka—Strelka[15] (v1.0.14) was also tested on the dataset. Files and directories were generated and configured according to the documentation (https://sites.google.com/site/strelkasomaticvariantcaller/). Strelka was run with the default parameters. Since the number of somatic indels with the FILTER field "PASS" was too few, we also selected "QSI_ref" field for evaluation.

Experimental Validation

The COLO-829 and COLO-829BL cell lines were purchased from the American Type Culture Collection (ATCC). Primers for genomic PCR were designed using Primer3 (http://biotools.umassmed.edu/bioapps/primer3_www.cgi). The COLO-829 and COLO-829BL cells were cultured in RPMI 1640(Sigma) supplemented with 10% FBS and 1% penicillin and streptomycin. Genomic DNA was extracted from COLO-829 and COLO-829BL cells using genome DNA kit (Invitrogen) and the PCR was performed using GoTaq DNA Polymerase (Promega). Thermal cycling conditions were one cycles of 95 °C for 2 min, followed by 30 cycles of 95 °C for 30 s, 65°C for 30 s and 72 °C 1 min, followed by a final extension step of 72 °C for 10 min. PCR products were electrophoresed on 1% agarose gels with ethidium bromide, visualized using UV light illumination.

Data

ICGC-TCGA DREAM Challenge data[13] [SRA:SRP042948] was downloaded from https://www.synapse.org/#!Synapse:syn2280639 with public token (*in silico* 1, 2 and 3) or approval access with private token from ICGC (*in silico* 4).

The whole genome sequencing data[17] [EGAD00000000055] of the immortal melanoma cancer cell line COLO-829 and lymphoblastoid cell line derived from the same patient COLO-829BL was requested from the European Genome-phenome Archive.

The TCGA breast cancer WGS data were obtained through dbGAP [accession number phs000178.v7.p6].

The low grade glioma sample (SJLGG039) is available at European Genome-phenome Archive under accession EGAS00001000255.

System requirements and software availability

novoBreak is written in C and Perl. The source code is freely available at https://sourceforge.net/projects/novobreak/?source=updater. For a 40X 2×101bp whole genome tumor and normal pairs, novoBreak needs a main memory less than 40GB and a running time less than ~6 hours with 10 CPU cores.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

We thank the ICGC-TCGA DREAM SMC Challenge organizers and participants for providing data and evaluations, Agda K. Eterovic and Gordon B. Mills for assistance with the experiment and manuscript. This study was supported in part by the National Institutes of Health [grant number R01 CA172652 and U41 HG007497 to K.C.], the National Cancer Institute Cancer Center Support Grant [P30 CA016672], Andrew Sabin Family Foundation to K.C. and a training fellowship from the Computational Cancer Biology Training Program of the Gulf Coast Consortia [CPRIT Grant No. RP140113] to Z.C. The results published here are in part based upon data generated by TCGA established by the NCI and NHGRI. Information about TCGA and the investigators and institutions who constitute the TCGA research network can be found at http://cancergenome.nih.gov/.

References

1. Mitelman F, Johansson B, Mertens F. The impact of translocations and gene fusions on cancer causation. Nat Rev Cancer. 2007; 7:233–245. [PubMed: 17361217]
2. Stephens PJ, et al. Complex landscapes of somatic rearrangement in human breast cancer genomes. Nature. 2009; 462:1005–1010. [PubMed: 20033038]
3. Alkan C, Coe BP, Eichler EE. Genome structural variation discovery and genotyping. Nature reviews Genetics. 2011; 12:363–376.
4. Alkan C, Sajjadian S, Eichler EE. Limitations of next-generation genome sequence assembly. Nat Methods. 2011; 8:61–65. [PubMed: 21102452]
5. Chen K, et al. BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. Nat Methods. 2009; 6:677–681. [PubMed: 19668202]
6. Abyzov A, Urban AE, Snyder M, Gerstein M. CNVnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. Genome Res. 2011; 21:974–984. [PubMed: 21324876]

7. Ye K, Schulz MH, Long Q, Apweiler R, Ning Z. Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. Bioinformatics. 2009; 25:2865–2871. [PubMed: 19561018]
8. Rausch T, et al. DELLY: structural variant discovery by integrated paired-end and split-read analysis. Bioinformatics. 2012; 28:i333–i339. [PubMed: 22962449]
9. Chen K, et al. TIGRA: A targeted iterative graph routing assembler for breakpoint assembly. Genome Res. 2014; 24:310–317. [PubMed: 24307552]
10. Li Y, et al. Structural variation in two human genomes mapped at single-nucleotide resolution by whole genome de novo assembly. Nat Biotechnol. 2011; 29:723–730. [PubMed: 21785424]
11. Earl D, et al. Assemblathon 1: A competitive assessment of de novo short read assembly methods. Genome research. 2011; 21:2224–2241. [PubMed: 21926179]
12. Boutros PC, et al. Global optimization of somatic variant identification in cancer genomes with a global community challenge. Nat Genet. 2014; 46:318–319. [PubMed: 24675517]
13. Ewing AD, et al. Combining tumor genome simulation with crowdsourcing to benchmark somatic single-nucleotide-variant detection. Nature methods. 2015; 12:623–630. [PubMed: 25984700]
14. McKenna A, et al. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome research. 2010; 20:1297–1303. [PubMed: 20644199]
15. Saunders CT, et al. Strelka: accurate somatic small-variant calling from sequenced tumor-normal sample pairs. Bioinformatics. 2012; 28:1811–1817. [PubMed: 22581179]
16. Li H. Exploring single-sample SNP and INDEL calling with whole-genome de novo assembly. Bioinformatics. 2012; 28:1838–1844. [PubMed: 22569178]
17. Pleasance ED, et al. A comprehensive catalogue of somatic mutations from a human cancer genome. Nature. 2010; 463:191–196. [PubMed: 20016485]
18. Wang J, et al. CREST maps somatic structural variation in cancer genomes with base-pair resolution. Nat Methods. 2011; 8:652–654. [PubMed: 21666668]
19. Zhang J, et al. INTEGRATE: gene fusion discovery using whole genome and transcriptome data. Genome Res. 2016; 26:108–118. [PubMed: 26556708]
20. Li H. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. 2013 arXiv preprint arXiv:1303.3997.
21. Bloom BH. Space/Time Trade/Offs in Hash Coding with Allowable Errors. Communications of the Acm. 1970; 13:422.
22. Sedgewick, R.; Wayne, K. Algorithms. 4. Addison-Wesley; Upper Saddle River, NJ: 2011.
23. Warren RL, Sutton GG, Jones SJM, Holt RA. Assembling millions of short DNA sequences using SSAKE. Bioinformatics. 2007; 23:500–501. [PubMed: 17158514]
24. DePristo MA, et al. A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nat Genet. 2011; 43:491-+. [PubMed: 21478889]
25. Van der Auwera GA, et al. From FastQ data to high confidence variant calls: the Genome Analysis Toolkit best practices pipeline. Curr Protoc Bioinformatics. 2013; 11:11 10 11–11 10 33.

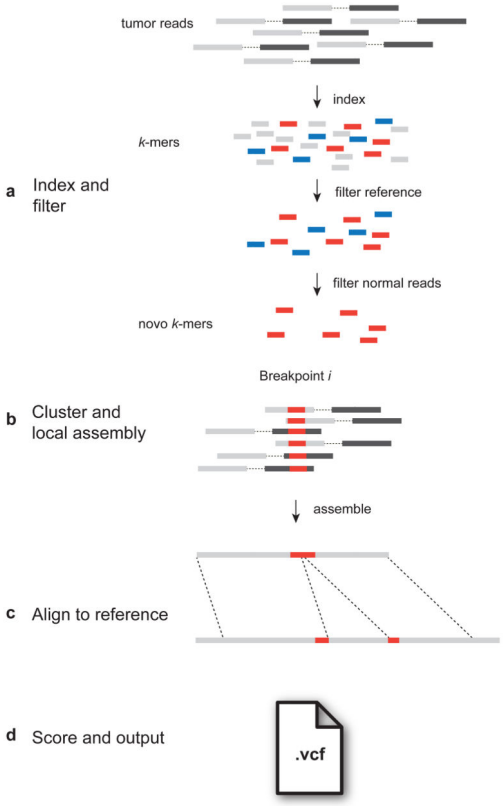

Figure 1. The workflow of novoBreak algorithm

(**a**) Short paired-end tumor reads (pairs of grey and black bars connected by dashed lines) are dissected into constituent *k*-mers. Indexed k-mers are compared against the reference sequence and normal reads. Only somatic novo *k*-mers (red bars) unique in the tumor genome are kept, while germline (green bar) and reference *k*-mers (grey bars) are filtered out. (**b**) The cluster of reads spanning a breakpoint *i* are found in conjunction with a set of shared novo *k*-mers. A long contig (grey bar) containing a unique breakpoint sequence in the middle (highlighted in red) is assembled from the cluster of reads. (**c**) This assembled contig is aligned (dashed line) against the reference sequence to infer exact breakpoint and associated SV. (**d**) Each SV breakpoint is scored, ranked and output in a standard variant call format (VCF) file.

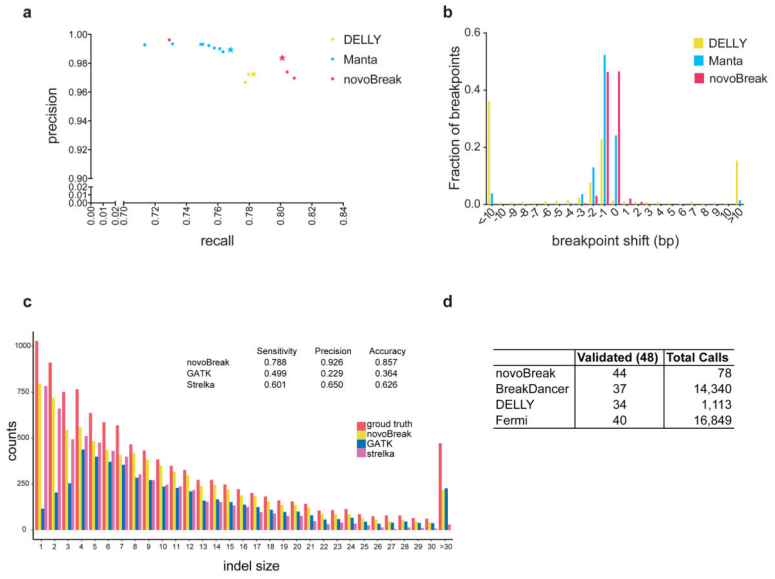

Figure 2. novoBreak performance

in the IS3 data. (**a**) Precision and recall comparison among 3 top-performing tools: novoBreak (green), DELLY (blue) and Manta (red). Star indicates the best scoring results of each tool. (**b**) Comparison of breakpoint precision among the 3 tools. X-axis is the offset (in base pair) between the true and predicted breakpoint coordinates. Y-axis is the fraction of predicted breakpoints at each of the offset values. (**c**) INDEL detection sensitivity of GATK-Haplotypecaller, Strelka and novoBreak in the IS4 data. (**d**) Summary of SV breakpoints detected in COLO-829 data by novoBreak, BreakDancer, DELLY and Fermi.