MoMI-G: Modular Multi-scale Integrated Genome Graph Browser

Toshiyuki T. Yokoyama, Yoshitaka Sakamoto, Masahide Seki, Yutaka Suzuki, Masahiro Kasahara*

Affiliations

Department of Computational Biology and Medical Sciences, Graduate School of Frontier Sciences, The University of Tokyo, Chiba, Japan

*Correspondence should be addressed to

Masahiro Kasahara

E-mail: mkasa@k.u-tokyo.ac.jp

Tel/Fax: +81 4 7136 4110

Keywords

Structural Variant; Genome Browser; Visualization; Variation Graphs; Long-read Sequencing; Genome Graphs

ABSTRACT

Long-read sequencing allows more sensitive and accurate discovery of structural variants (SVs). While more and more SVs are being identified, a number of them are difficult to visualize using existing SV visualization tools. Therefore, methods to visualize SVs such as nested or large SVs of over a megabase pair need to be developed. To this end, we developed MOdular Multi-scale Integrated Genome graph browser, MoMI-G, a web-based genome browser to visualize SVs, genes, repeats, and other annotations as a variation graph with paths. This browser allows more intuitive recognition of large, nested, and potentially more complex SVs. MoMI-G has view modules for different scales, which allow users to view the whole genome down to nucleotide-level alignments of long reads. Alignments spanning reference alleles and those spanning alternative alleles are shown in the same view. Users can customize the view, if they are not satisfied with the preset views. In addition, MoMI-G has Interval Card Deck, a feature for rapid manual inspection of hundreds of SVs. Herein, we describe the utility of MoMI-G by using representative examples of large and nested SVs found in two cell lines, LC-2/ad and CHM1. MoMI-G is freely available at https://github.com/MoMI-G/MoMI-G under the MIT license.

INTRODUCTION

Structural Variants (SVs), which are often characterized as 50 bp or larger genomic rearrangements of chromosomal segments, are associated with various human diseases (Stankiewicz and Lupski 2010; Weischenfeldt et al. 2013; Sedlazeck et al. 2018a). For example, some fusion genes caused by SVs are known as oncogenes (Mertens et al. 2015). Identifying SVs and interpreting their potential impacts are a critical step toward cataloguing the variations in the human genome and mechanistic understanding of genetic diseases and cancers.

SV visualization is a very important step in an SV calling process because it enables the manual inspection of SVs for achieving two goals. The first is to better understand the relationships between SVs and other genomic features. The second is to ensure a smaller number of false positives. Previously, most structural genomic rearrangements were categorized into insertion, deletion, inversion, duplication, and translocation, which were referred to by some researchers as canonical SVs (Quinlan and Hall 2012; Collins et al. 2017). SV visualization tools focused on visualizing canonical SVs, because they accounted for a significant portion of the identified SVs at that time.

However, as long-read sequencing technologies revealed an increasing number of SVs, SV visualization with the existing tools became more challenging. For example, a large inversion is often identified as two separate translocations at the two breakpoints of the inversion; one might not be able to immediately recognize that the two translocation events are explained by a single large inversion. Another example is a nested SV. When there is a large inversion that contains several smaller SVs such as insertion of transposons or deletions, the nested SVs often obscure the relationship between genomic regions that are distant in the reference genome, but are actually close in the target genome. Thus, SV visualization tools should be able to simultaneously display multiple intervals along with their relationships, even when the breakpoints are distant or when SVs are nested.

For the second goal, manual inspection of SVs identified using SV calling tools is important

61 because these tools are not yet accurate enough; therefore, human experts are required to accurately
62 and reliably distinguish true positive SVs from false positive ones. False positives often increase under
63 the following conditions: (1) when the sequencing coverage is low, (2) when the sequencing error rate
64 of reads used for SV calling is high, (3) when the target genome has segmental duplications or
65 abundant repetitive sequences, or (4) when many SVs are heterozygous. Therefore, SV candidates
66 need to be manually inspected using read alignments and genomic annotations (Guan and Sung 2016)
67 and tens of thousands of them need to be filtered. However, manual filtering by using existing SV
68 visualization tools occasionally becomes very difficult for certain cases. For example, for nested SVs
69 and long reads spanning over multiple breakpoints, existing tools cannot show the read alignments in
70 multiple intervals at a glance, making it unrealistic to manually judge the authenticity of candidate
71 SVs.

72 To achieve these two goals, we developed MoMI-G (pronounced as mo-me-gee), a genome
73 graph browser that visualizes SVs using variation graphs (Fig. 1, Supplemental Fig. 1). Herein, we
74 describe the use cases and features of MoMI-G using the LC-2/ad human lung adenocarcinoma cell
75 line that carries a CCDC6-RET fusion gene (Matsubara et al. 2012; Suzuki et al. 2014, 2015, 2017),
76 and CHM1, a human hydatidiform mole cell line that originates from a single haploid (Chaisson et al.
77 2015). MoMI-G helps in understanding the entire picture of SVs, even those that are nested or large,
78 regardless of their size. MoMI-G allows researchers to obtain novel biological knowledge by
79 comparing a reference genome with an individual genome by using a variation graph.

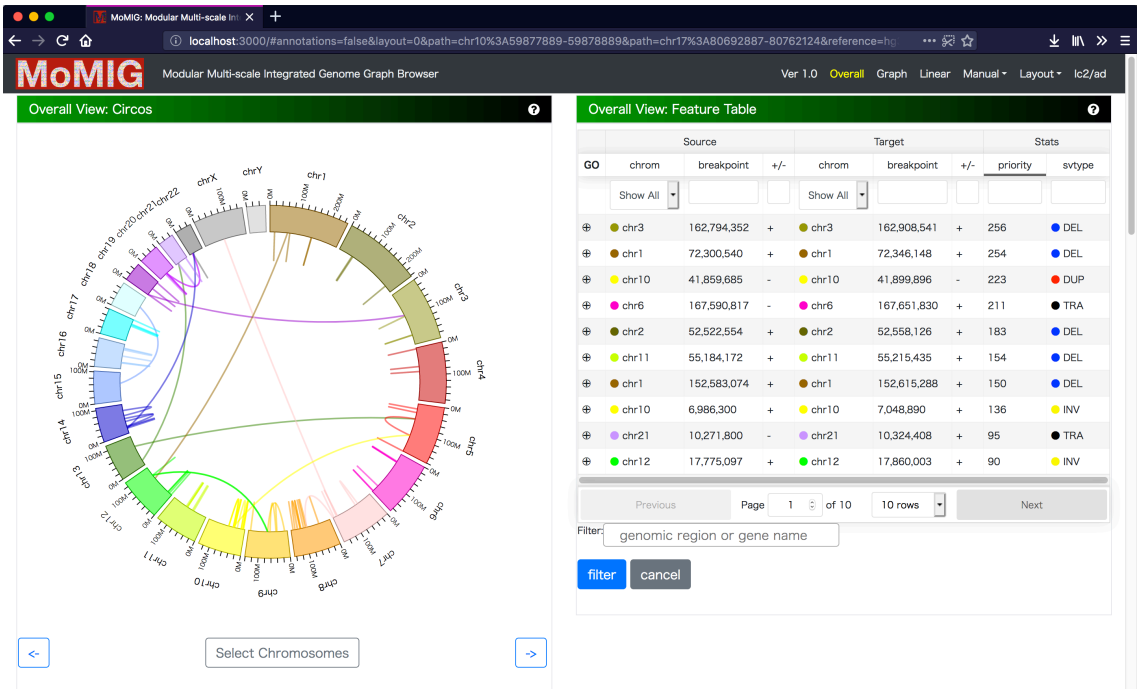
80 The reason for dubbing MoMI-G as a "genome graph" browser is that we employed genome
81 graphs as a theoretical backbone for providing more systematic way of presenting SVs with varying
82 complexities, including nested and large SVs. A genome graph is a new technique to represent multiple
83 genome sequences as a graph (Paten et al. 2017). For example, a cancer genome can be represented as
84 a graph with SVs embedded as alternative edges (Nattestad et al. 2016a). Several variants in the

85 definitions of a sequence graph are available (Paten et al. 2018; Garrison et al. 2018). The definition
86 of a variation graph used herein is almost the same as the one used in SequenceTubeMap
87 (https://github.com/vgteam/sequenceTubeMap). A variation graph is a bi-directed graph composed of
88 nodes and paths. A node represents a part of a DNA sequence. A path represents a contiguous sequence,
89 which can be obtained by concatenating nodes in a way specified by the path (i.e., a list of <node ID,
90 order, orientation>). SequenceTubeMap is a JavaScript library used in MoMI-G for visualizing a
91 variation (sub)graph in a web browser (i.e., client side). In the server side, vg is used for retrieving a
92 subgraph of variation graphs (Garrison et al. 2018). Genome graphs can represent SVs more naturally
93 than those that represent SVs as differences from a reference genome (e.g., VCF).

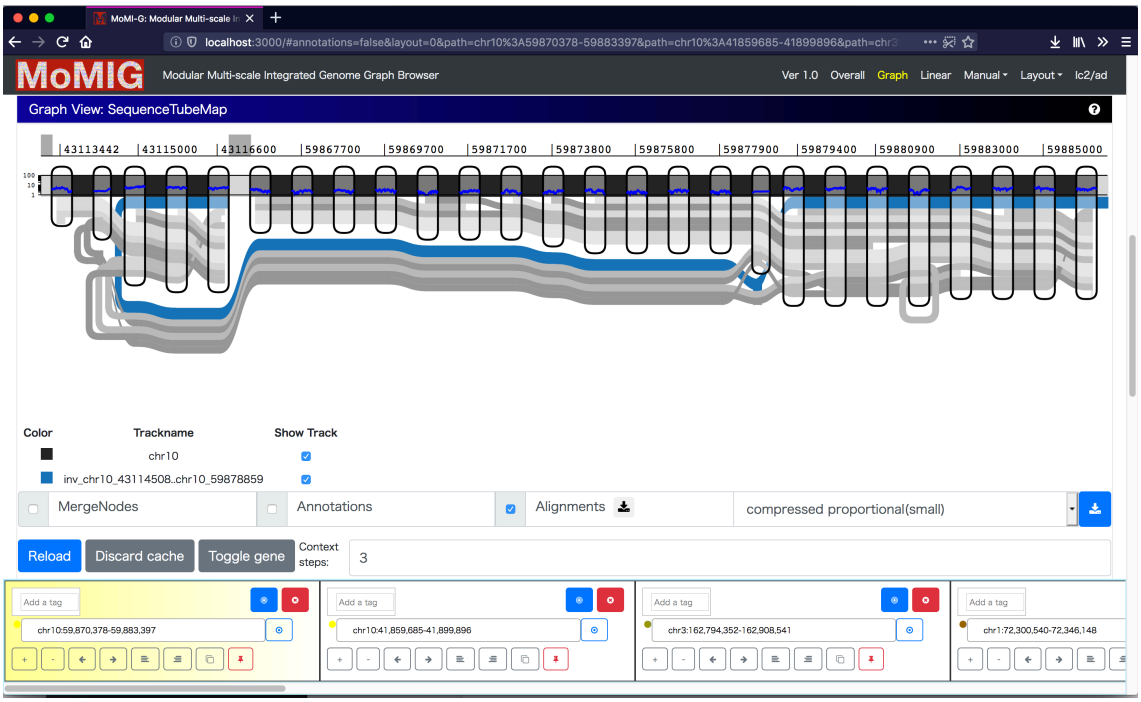
94 To our knowledge, MoMI-G is the only SV visualization tool that satisfies the following
95 conditions: (1) allows visualization of (possibly distant) multiple intervals; (2) displays SVs that span
96 multiple intervals; (3) displays SVs at varying scales, i.e., chromosome, gene, and nucleotide scales;
97 (4a) the chromosome scale view can show the distribution of SVs on one or more chromosomes; (4b)
98 the gene scale view can show annotations such as exon/intron structures and repeats; (4c) the
99 nucleotide scale view can show nucleotide-level alignments, in particular, read alignments that
100 correspond to both alleles of heterozygous SVs are shown simultaneously; and (5) allows users to
101 manually inspect hundreds of SVs. Most genome browsers such as UCSC (Kent et al. 2002), JBrowse
102 (Buels et al. 2016), and IGV (Thorvaldsdóttir et al. 2013) are designed for inspecting single-nucleotide
103 variants or short insertions and deletions by showing annotations and read alignments on a reference
104 genome, but none of them meet conditions (2), (3), (4a), and (5). Tools for visualizing SVs in a human
105 genome, such as 10X Loupe (http://loupe.10xgenomics.com/loupe/), iFUSE (Hiltemann et al. 2013),
106 targetSeqView (Halper-Stromberg et al. 2014), AGFusion (Murphy and Elemento 2016), SVPV
107 (Munro et al. 2017), BreakPoint Surveyor (Wyczalkowski et al. 2017), NGB (Ahdesmäki et al. 2017),
108 VIPER (Wöste and Dugas 2018), SV-plaudit (Belyeu et al. 2018), and MAVIS (Reisle et al. 2018), are

suitable for inspecting an SV between two intervals, but they do not visualize an SV spanning over multiple intervals. Circos (Krzywinski et al. 2009) and Circos-based visualizations such as GenomeRing (Herbig et al. 2012) and CIRCUS (Naquin et al. 2014) are widely used for visualizing the distribution of SVs by using the circular layout at the chromosome scale, but they fail in meeting conditions (3), (4b), and (4c). Svviz (Spies et al. 2015) and BasePlayer (Katainen et al. 2018) can be used to visualize an SV between multiple intervals of a reference genome, although they cannot be used to visualize nested SVs. Fastbreak (Bressler et al. 2012) integrates views at different scales by using a graph representation; however, it has no view for nucleotide-scale information such as read alignment. Gremlin (O'Brien et al. 2010) and ViVar (Sante et al. 2014) allow visualization of inter-chromosomal relationships of SVs, but they cannot visualize read alignments. Ribbon and SplitThreader (Nattestad et al. 2016a, 2016b) are a pair of tools for visualizing SVs with different views such as Circos, SVs between two chromosomes, read alignments, and genomic annotations, but they lack tight integration, and hence an ID to specify the coordinate interval needs to be manually transferred, which could lead to non-negligible time loss when hundreds of SVs need to be inspected. Moreover, they cannot visualize read alignments for heterozygous SVs. Assembly graph visualization tools such as ConPath, ABySS Explorer, TGNet, ContigScape, Contiguity, Bandage, and BARLEX (Kim et al. 2008; Nielsen et al. 2009; Riba-grognuz et al. 2011; Tang et al. 2013; Revanna et al. 2012; Wick et al. 2015; Colmsee et al. 2015) are also useful for visualizing contigs and their connections. However, they are designed for small genomes. When these tools are used to visualize a graph for larger genomes, an uninterpretable large dense hairball is usually obtained. Displaying only a part of the graph is critical for avoiding the hairball problem. For the same reason, graphviz (Gansner and North 1999) is unsuitable for a larger genome graph. No tools described above (other than MoMI-G) meet all conditions (1) to (5).

A.


MoMIG: Modular Multi-scale Integ
localhost:3000/#annotations=false&layout=0&path=chr10%3A59877889-59878889&path=chr17%3A80692887-80762124&reference=hg
MoMIG
Modular Multi-scale Integrated Genome Graph Browser
Ver 1.0 Overall Graph Linear Manual Layout Ic2/ad
Overall View: Circos
Overall View: Feature Table
Source Target Stats
GO chrom breakpoint +/- chrom breakpoint +/- priority svtype
Show All Show All
chr3 162,794,352 + chr3 162,908,541 + 256 DEL
chr1 72,300,540 + chr1 72,346,148 + 254 DEL
chr10 41,859,685 - chr10 41,899,896 - 223 DUP
chr6 167,590,817 - chr6 167,651,830 + 211 TRA
chr2 52,522,554 + chr2 52,558,126 + 183 DEL
chr11 55,184,172 + chr11 55,215,435 + 154 DEL
chr1 152,583,074 + chr1 152,615,288 + 150 DEL
chr10 6,986,300 + chr10 7,048,890 + 136 INV
chr21 10,271,800 - chr21 10,324,408 + 95 TRA
chr12 17,775,097 + chr12 17,860,003 + 90 INV
Previous Page 1 of 10 10 rows Next
Filter: genomic region or gene name
filter cancel
Select Chromosomes

B.


MoMI-G: Modular Multi-scale In
localhost:3000/#annotations=false&layout=0&path=chr10%3A59870378-59883397&path=chr10%3A41859685-41899896&path=chr3
MoMIG
Modular Multi-scale Integrated Genome Graph Browser
Ver 1.0 Overall Graph Linear Manual Layout Ic2/ad
Graph View: SequenceTubeMap
43113442 43115000 43116600 59867700 59869700 59871700 59873800 59875800 59877900 59879400 59880900 59883000 59885000
Color Trackname Show Track
chr10
inv_chr10_43114508..chr10_59878859
MergeNodes Annotations Alignments compressed proportional(small)
Reload Discard cache Toggle gene Context steps: 3
Add a tag
chr10:59,870,378-59,883,397
chr10:41,859,685-41,899,896
chr3:162,794,352-162,908,541
chr1:72,300,540-72,346,148

C.

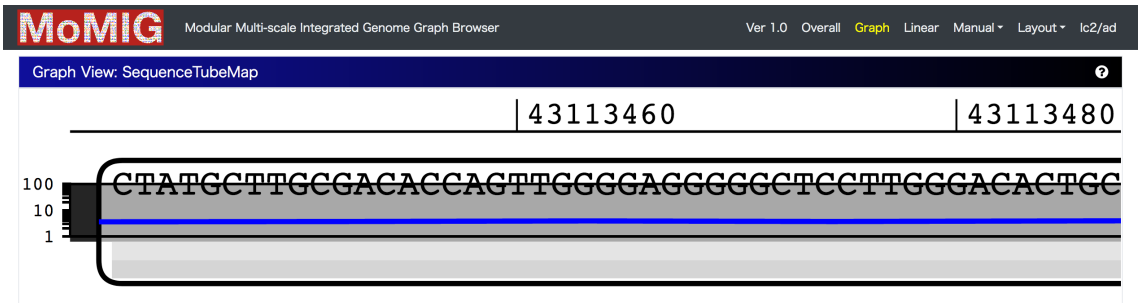


Figure 1. Overview of MoMI-G. A user typically selects one of the preset combinations of view modules. The user can customize the window by adding or removing view modules, if necessary. Three examples showing views of different scales are shown. Comprehensive descriptions for all modules are shown in Supplemental Fig. 1.

(A) Chromosome-scale view: Circos Plot (left) shows the distribution of SVs over all chromosomes. Arcs are chromosomes. Curves represent SVs. Feature Table (right) shows a filtered/sorted list of the SVs in an input VCF file.

(B) Gene-scale view: SequenceTubeMap (top) shows the graphical view of the genomic region selected in Circos Plot, Feature Table, or Interval Card Deck. A rounded rectangle is a node that represents a piece of a genomic sequence. The thick lines spanning over nodes are paths; the horizontal thick black line with light/dark shades is a chromosome of the reference genome, and the blue line indicates one end of an inversion. The color of lines indicating SVs is assigned arbitrarily. Read alignments are shown as gray thin lines, suggesting that the inversion here is likely heterozygous. Interval Card Deck (bottom) queues a list of genomic intervals for candidate SVs selected by using Circos Plot or Feature Table for rapidly screening hundreds of candidate intervals.

(C) Nucleotide-scale view: SequenceTubeMap can show nucleotides.

RESULTS

MoMI-G is a web-based genome browser developed as a single-page application implemented in TypeScript and with React. Because users need different types of views even for the same data, MoMI-G provides three groups of view modules for the analysis of SVs at different scales, namely, chromosome-scale, gene-scale, and nucleotide-scale view groups (Supplemental Table 1). Users can use one or more view modules in a single window.

The input of MoMI-G is a variation graph, read alignment (optional), and annotations (optional). MoMI-G accepts a succinct representation of a vg variation graph, an XG file, as a variation graph. A script that converts a FASTA file of a reference genome and a common variant format (VCF) file into an XG file is included in the MoMI-G package, although the VCF format cannot represent some types of SVs that the XG format can represent, such as nested insertions. Read alignment data on the graph need to be a graph alignment map (GAM) file; alternatively, users can convert a binary alignment map (BAM) file into a GAM file, although this is not recommended due to some limitations.

We show three examples from two samples for demonstrating the utility of MoMI-G. One of the examples is a large inversion, and the other is nested SVs that are difficult to visualize using existing tools. We also show nested SVs from the CHM1 dataset (Chaisson et al. 2015). For all the examples, we used Amazon EC2 instance type t2.large with 8 GB of memory and 2.4 GHz Intel Xeon processor as a MoMI-G server; the server requirement is minimal. MoMI-G supports common browsers, including Chrome, Safari, and Firefox.

Data model used in MoMI-G and MoMI-G tools

To our knowledge, no publicly available SV visualization tools are available for large and nested SVs with alignments of long reads. Thus, we aimed at developing a graph genome browser at the earliest so that users can obtain new biological knowledge from real data. We used an existing library,

183 SequenceTubeMap for visualizing a variation subgraph, rather than developing our own library from
184 scratch.

185 SequenceTubeMap is a JavaScript library that visualizes multiple related sequences such as
186 haplotypes. A variation graph used in SequenceTubeMap is a set of nodes and paths, where a node
187 represents a part of a DNA sequence, and a path represents (a part of) a haplotype. Edges are implicitly
188 represented by adjacent nodes in paths.

189 MoMI-G accepts variation graphs in which SVs are represented by paths so that
190 SequenceTubeMap can visualize them. A deletion is represented by a path that skips over a sequence
191 that other paths pass through. Similarly, an insertion is represented by a path that passes through an
192 extra sequence that other paths do not visit; an inversion is represented by a path where a part of the
193 sequence in other paths is reversed; and a duplication is represented by a path that passes through the
194 same sequence twice or more.

195 The MoMI-G package includes a set of scripts (MoMI-G tools) that converts a VCF file into
196 the variation graph format. We used MoMI-G tools for generating the input variation graphs;
197 alternatively, users can generate variation graphs on their own. See the method section and
198 Supplemental Figure 2 for the details of MoMI-G tools. Briefly, MoMI-G tools convert a VCF record
199 into a path in the output variation graph. A deletion is converted into a path that starts at the most 1
200 Mbp before one breakend of the deletion, traverses to the breakend, jumps to the other breakend, and
201 proceeds for a certain length (<1 Mbp). Note that the sequences flanking the deletion are added to
202 indicate the edge representing the deletion because edges are implicitly represented in
203 SequenceTubeMap. Insertions, inversions, and duplications are similarly represented by paths with
204 flanking sequences.

205

206 **Visualization Examples**

207 *Revealing a large SV: A large inversion and a subsequent short deletion*

208 We show an example of revealing a complex SV, where a large inversion and two flanking deletions
209 to the reference genome are identified as two SVs, each of which connects two different points on a
210 chromosome (i.e., INV or BND record in VCF). Previous studies involving the use of whole genome
211 sequencing or RNA-seq by using Illumina HiSeq or Nanopore MinION, identified the CCDC6-RET
212 fusion gene in LC-2/ad (Suzuki et al. 2014, 2015, 2017; Sereewattanawoot et al. 2018). However,
213 those studies focused only on the region around the CCDC6-RET fusion point, and the entire picture,
214 including the other end of the inversion, was unclear. To address this issue, we explored the wider
215 region around CCDC6-RET by using MoMI-G.

216 First, we sequenced the genome of LC-2/ad by using Oxford Nanopore MinION R9.5 pore
217 chemistry and merged reads with those from a previous study (accession No. DRX143541-
218 DRX143544) (Sereewattanawoot et al. 2018). We generated 3.5 M reads to 12.8× coverage in total,
219 and then aligned them with GRCh38. The average length of the aligned reads was 16 kb (Supplemental
220 Table 2). We detected 11,316 SVs in the VCF format, including the previously known CCDC6-RET
221 fusion gene, on the nuclear DNA of LC-2/ad cell line (Supplemental Table 3). See the method section
222 for details.

223 The distance between RET (chr10: 43,075,069–43,132,349) and CCDC6 (chr10:
224 59,786,748–59,908,656) is about 17 Mbp in GRCh38. We confirmed that a CCDC6-RET fusion gene
225 exists in LC-2/ad (Fig. 2A). This fusion gene is presumably caused by an inversion, although only one
226 end of the inversion was found. We found an unknown novel adjacency that well explains the other
227 end of the inversion (Fig. 2B, Supplemental Table 4). MoMI-G was able to display the relationships
228 between the two breakends of the inversion, enabling users to understand large SVs. We explored the
229 read alignments around the fusion and found that the fusion was heterozygous (Fig. 2C). MoMI-G is

230 the first stand-alone graph genome browser that can display long-read alignments over branching
231 sequences that represent a heterozygous SV.

232 Further, we found that the large inversion was flanked by two small deletions. These
233 deletions are explained by a single deletion event following the large inversion event (Fig. 2D). The
234 loss of the RET-CCDC6 fusion gene corresponds to the two small deletions on GRCh38. A simple
235 explanation is that a deletion occurred after the inversion event, but not vice versa, in favor of the
236 smaller number of mutation events.

237 Next, we attempted to estimate the breakpoints of the large inversion before the deletion
238 occurred. There were two possible scenarios for the positions of the two breakends of the large
239 inversion. The first is that RET-CCDC6 and CCDC6-RET were generated by a large inversion, and
240 then RET-CCDC6 was lost. The second is that CCDC6 was first broken by a large inversion, and a
241 subsequent small deletion led to CCDC6-RET. Previous studies support the former scenario. First, the
242 RET gene often tends to be disrupted in thyroid cancer by paracentric inversion of the long arm of
243 chromosome 10, or by chromosomal fusion (Santoro and Carlomagno 2013). Second, in a previous
244 study, two clinical samples had both RET-CCDC6 and CCDC6-RET in the genome (Mizukami et al.
245 2014). Both studies suggested that an inversion disrupted both CCDC6 and RET, and then a small
246 deletion disrupted RET-CCDC6. We could never recognize these two deletions flanking the large
247 inversion without simultaneously observing both the inversion records in VCF.
248

249 *Nested SVs with alignment coverage*

250 Visualizing nested SVs is necessary for evaluating the output of SV callers. However, most existing
251 genome browsers cannot visualize nested SVs as well as the relationships between them. Genome
252 browsers, including IGV, collapse SVs into intervals between breakpoints, and thus the topological

253 relationships between nested SVs are not shown. MoMI-G can visualize nested SVs as a variation

254 graph (Fig. 3).

255

256 *Visualizing nested SVs in a pseudodiploid genome*

257 We show an example of nested SVs in a pseudodiploid genome visualized using MoMI-G. We

258 downloaded a CHM1 genome with an SV list previously generated in a whole-genome resequencing

259 study by using PacBio sequencers from human hydatidiform (Chaisson et al. 2015). The SV list

260 includes insertions, deletions, and inversions for GRCh37/hg19. We converted the BED file of the

261 SV list of CHM1 to a VCF file, and then filtered out deletions of less than 1,000 bp for focusing on

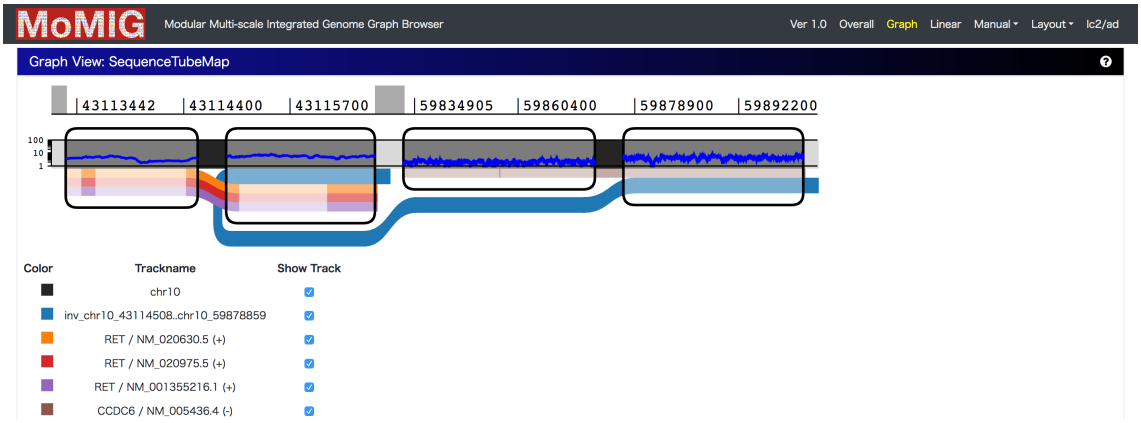
262 medium to large SVs. We found nested SVs for which existing genome browsers do not intuitively

263 show the relationships between them (Fig. 4). This example indicates that four insertions and

264 deletions occur in the large inversion.

265

266 **A.**

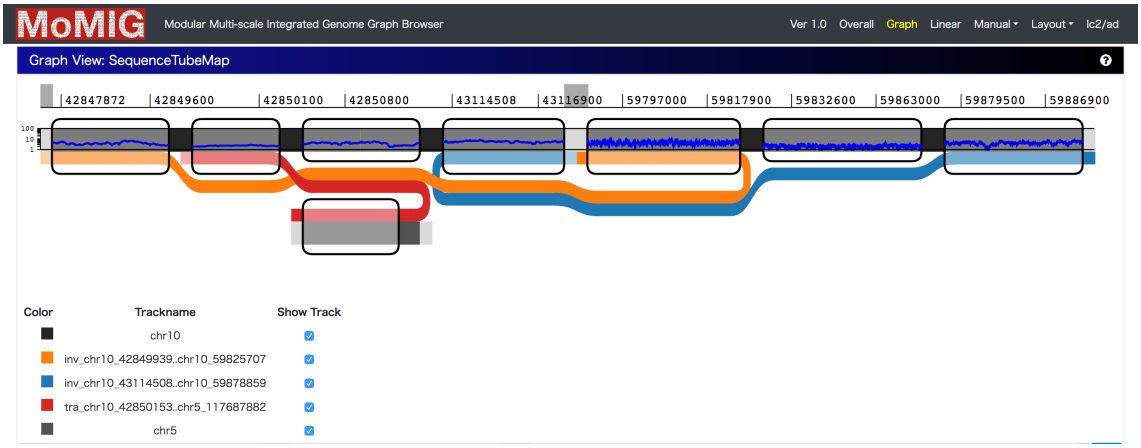


267

268

269 **B.**

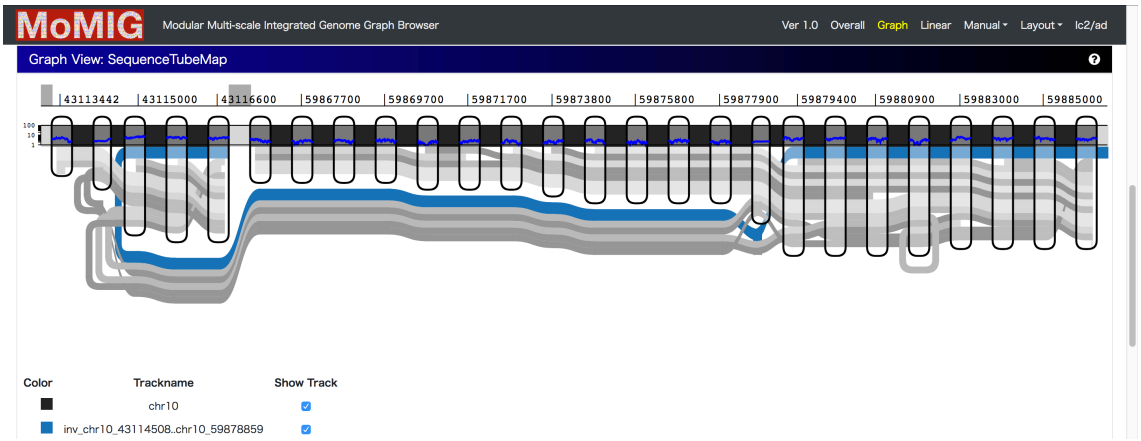
270



271

272

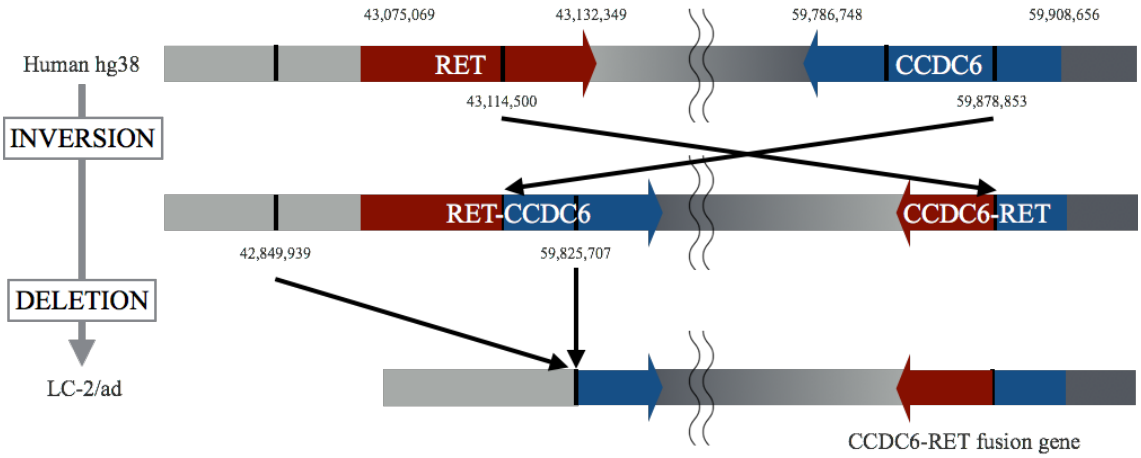
273 **C.**



274

275

276 **D.**



277

Figure 2. Example of CCDC6-RET

(A) CCDC6-RET shown in MoMI-G (compressed view). The thickest black line is chromosome 10 (reference genome). Note that the two distinct intervals of chromosome 10 are shown, which correspond to RET (left interval) and CCDC6 (right interval) genes,. The blue line represents an inversion identified by Sniffles, showing the CCDC6-RET fusion event. The other lines are gene annotations in hg38; the orange, red, purple lines indicate two isoforms of RET, and the brown line is CCDC6. (B) CCDC6-RET with read alignments shown as grey lines. Further, some alignments do not support the inversion, suggesting that CCDC6-RET is heterozygous. (C) The entire picture of the inversion that caused CCDC6-RET. This inversion is too large to span by a single read; thus, it was identified as two independent fusion events at both the ends of the inversion, which would be difficult to understand if the two fusion events are visualized separately. The red line is a translocation that was not analyzed in this study. (D) Putative evolution process of LC-2/ad at the CCDC6-RET site. First, a long inversion generated two fusion genes, CCDC6-RET and RET-CCDC6. Second, a large deletion caused the loss of RET-CCDC6.

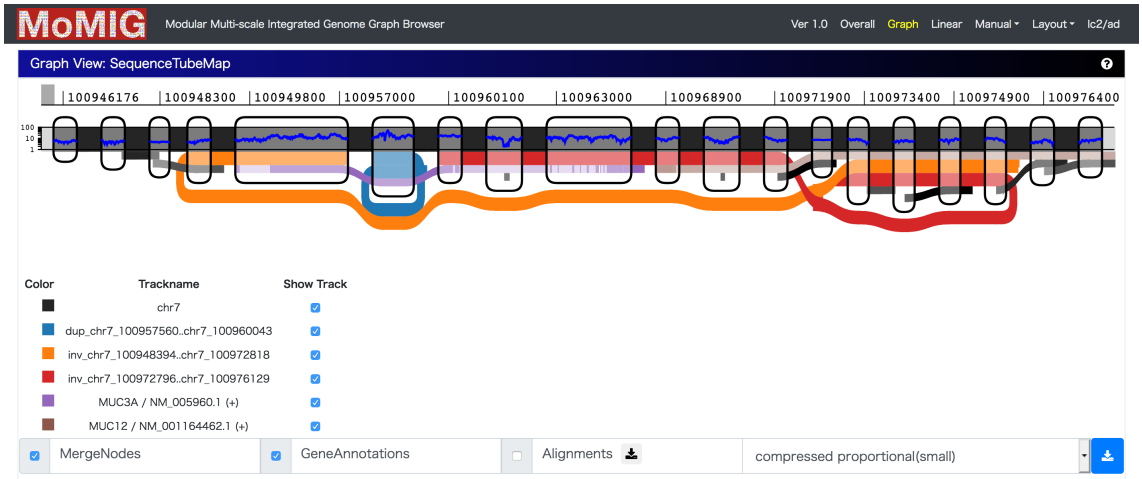
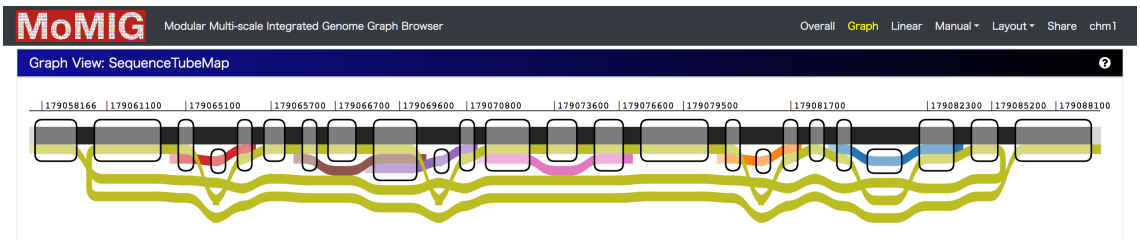


Figure 3. Nested SVs called by Sniffles in LC-2/ad

296 The thin black lines are repeat annotations. The brown and purple lines are gene annotations. The red
297 and orange lines are an end of an inversion called by Sniffles. There are two possibilities for the
298 genome structure: one is that MUC3A and its flanking region are a duplication, and the internal region
299 of MUC12 is an inverted duplication. The other is that MUC3A and its flanking region are an inverted
300 duplication, and the internal region of MUC12 is a duplication. Several read alignments support the
301 former interpretation. Although SVs called from the Illumina reads did not include any of the SVs
302 shown here, the alignment coverage by the Illumina reads is consistent with both duplications. Note
303 that the y-axis of the blue thin line on the chromosome showing the alignment coverage is logarithmic.
304



305

306 **Figure 4. Nested SVs in CHM1**

307 The black line represents a part of chromosome 5, where a large inversion is shown as the brown line.
308 The other lines are smaller SVs included in the large inversion. Because CHM1 is a pseudodiploid
309 genome, all the SVs shown in this figure must be on the same haplotype, although MoMI-G tools
310 assume diploid (polyploid) genomes and show the inner SVs as heterozygous SVs.
311

312 User-interface Design

313 The optimal way of visualizing SVs might vary. To rapidly explore the distribution of SVs in a genome,
314 users might wish to use Circos-like plots. Other users might intend to focus on local graph structures
315 of SVs that might contain a few genes. In another scenario, a user might want to explore individual
316 nucleotides. To address this issue, MoMI-G provides a customizable view in which users can place
317 any combination of view modules. Further, preset view layouts are available for users' convenience.

318

319 *Enabling easy manual inspection of detected SVs*

320 Manual inspection, which includes determining if an SV is heterozygous or homozygous, confirming
321 what part of a gene is affected by that SV, and determining the reason why an SV is called based on
322 read alignments, is an important part for validating called SVs. As variants called by SV callers
323 increase, the burden of manual inspection also increases, underscoring the importance of visualization
324 both to inspect individual SV calls for filtering out false positives and to ensure that a filtered set of
325 SVs is of high confidence (Munro et al. 2017; Wöste and Dugas 2018). MoMI-G help with the efficient
326 inspection of SVs by using (1) Feature Table, which is an SV list; (2) Interval Card Deck, which is
327 genomic coordinate stacks; and (3) shortcut keys. The usage is as follows: (1) one can filter SVs using
328 Feature Table, after which SVs are selected, and then (2) the listed variants are stacked on Interval
329 Card Deck at the bottom of the window. In Interval Card Deck, intervals are displayed as cards, and
330 the interval of the top (leftmost) card of the deck is shown on SequenceTubeMap. Each card can be
331 dragged, and the order of cards can be changed. If one double-clicks on a card, the card moves to the
332 top of the deck. A tag can be added for a card for later reference. Further, a card can be locked to avoid
333 unintended modification or disposal, and the gene name can be input with autocompletion for
334 specifying the interval of a card.

335 When the interval to view is changed, only a part of the view that needs an update is re-
336 rendered, whereas most genome browsers working on web interface require rendering the entire view.
337 Interval Card Deck enables the rapid assessment of hundreds of intervals. Moreover, deciding whether
338 an SV should be discarded or held becomes easier with shortcut keys. After all SVs are inspected, a
339 set of SVs held on the Interval Card Deck is obtained, which might be a set of interesting SVs or a set
340 of manually validated SVs. MoMI-G enables the rapid inspection of hundreds of SVs, providing a tool
341 for validating hundreds of SVs or for selecting interesting SVs.

342

343 *Input requirements*

344 MoMI-G inputs an XG format as a variation graph. Users can specify a GAM file with an index that
345 contains read alignment. They can convert a BAM file into GAM by using MoMI-G tools, or can
346 generate the GAM file on their own. When a BED file of genes is provided, users can specify a
347 genomic interval by gene name. A configuration file is written in YAML. MoMI-G also accepts bigBed
348 and bigWig formats (Kent et al. 2010) for visualizing annotations (e.g., repeats, genes, alignment depth,
349 and GC content) on the reference genome. The bigBed and bigwig need to be extended for graph
350 genomes in the future. The list of formats that are accepted by MoMI-G is shown in Table 1.

351

352 **Table 1. MoMI-G data files**

353

File type	Extension	Description
A succinct index of variation graphs	.xg	Variation graphs displayed in MoMI-G.
Graphical alignment/map	.gam	Read alignment. (optional)
Comma-separated values	.csv	SV list for chromosome-scale view (optional)
Browser extensible data	.bed	Used for converting gene names to genomic intervals. Also used for autocompletion of gene names. (optional)
Compressed binary indexed BED	.bb	Annotations. (optional)
Compressed binary indexed wiggle	.bw	Annotations. (optional)

354

355

DISCUSSION

We developed a genome graph browser, MoMI-G, that visualizes SVs on a variation graph. Existing visualization tools for SVs show either one SV at a time, or all SVs together; the former does not allow the understanding of the relationships between SVs, whereas the latter is useless when the target genome is very large and the whole variation graphs are too complicated to view in a single screen (i.e., the hairball problem). MoMI-G allows viewing only a part of the genome, which resolves the hairball problem, while providing an intuitive view for multiple SVs, including large and nested SVs. Further, MoMI-G enables the manual inspection of complex SVs by providing integrated multiple view modules; users can filter SVs, validate them by using read alignments, and interpret them by using genomic annotations.

We used vg as a server-side library and SequenceTubeMap as a client-side library for subgraph retrieving and visualization for genome graphs because, to our knowledge, these are the only combinations that are publicly available. We found that significant amount of engineering efforts are required for an even better user experience. For example, vg is a standalone command line application; therefore, it does not allow to store succinct index on memory; every time only a part of the genome is retrieved, the entire index of several gigabytes is loaded, which is unnecessary overhead. SequenceTubeMap displays inversions and duplications as loops; however, we found that new users occasionally find it difficult to recognize the connections between nodes. Visualizing SVs is still an open problem.

The currently available tools and formats for SV analysis have many problems. First, different SV callers output different VCF records even for the same SV. For example, depending on SV callers, an inversion with both boundaries identified at a base pair level is represented by one of the following: (1) a single inversion record, (2) two inversion records at both ends, (3) two breakend records at both ends, or (4) four breakend records at both ends (a variant of (3), but the records are

380 duplicated for both the strands). Thus, developing a universal tool for variant graph construction is
381 difficult. Second, certain types of nested SVs such as an insertion within an insertion are impossible
382 to represent in a VCF file without tricks, although variant graphs can easily handle these SVs.
383 Therefore, generating a variation graph from a VCF file including SVs is not ideal. We need an SV
384 caller that directly outputs variation graphs.

385 Fostering the ecosystem around variation graphs is important for delivering their benefits to
386 end users, as noted in the ecosystem around the SAM/BAM formats that spurred development of
387 production-ready tools for end users. MoMI-G is the first step toward such a goal because the
388 availability of tools ranging from upstream analysis such as read alignment to visualization is critical
389 for the entire ecosystem.

390 This is the one million genome era that requires rapid and memory-sufficient data structure
391 to allow alignments and store haplotype information. Because most parts of the genome are shared
392 between individuals, we need to focus on differences for reducing computation time and resources.
393 Therefore, genome graph is considered promising, especially for human variation analysis. Moreover,
394 new visualization methods as well as genome analysis methods are required. Genome graph browsers
395 should be able to handle even thousands of genomes in the near future. MoMI-G is a step forward for
396 visualizing genome graphs and could allow the development of new algorithms on genome graphs.

397

398 **MATERIAL and METHODS**

399 *Datasets*

400 High-molecular-weight (HMW) gDNA was extracted from lung cancer cell line, LC-2/ad by using
401 Smart DNA prep(a) kit (Analykjena). WGS data were produced from MinION 1D sequencing (SQK-
402 LSK108), MinION 1D^2 sequencing (SQK-LSK308), and MinION Rapid sequencing (SQK-
403 RAD003). For MinION 1D sequencing, 4 μg HMW gDNA was quantified using Tape Station. DNA

404 repair was performed using NEBNext FFPE DNA Repair Mix (M6630, NEB). End-prep was
405 performed using NEBNext Ultra II End Repair/dA-Tailing Module (E7546L, NEB). Adapter ligation
406 was performed using NEBNext Blunt/TA Ligase Master Mix (M0367L, NEB) and Ligation
407 Sequencing Kit 1D (SQK-LSK108, Oxford Nanopore Technologies). Libraries were sequenced for 48
408 h by using MinION (R9.5 chemistry, Oxford Nanopore Technologies). For MinION 1D^2 sequencing,
409 the protocol was the same as that for 1D excluding adapter ligation by using Ligation Sequencing Kit
410 1D^2 (SQK-LSK308, Oxford Nanopore Technologies). The library for MinION Rapid sequencing
411 was prepared according to Sequencing Kit Rapid (SQK-RAD003, Oxford Nanopore Technologies).
412
413 *Nanopore data alignment*
414 Nanopore WGS data were aligned against the GRCh38 human reference genome by using NGM-LR
415 with "-x ont" option (version 0.2.6) for calling SVs by using Sniffles version 1.0.7 (Sedlazeck et al.
416 2018b). MinION 1D^2 sequencing can produce a 1D^2 read, which integrates the information of a
417 read and its complementary read into one read. MinION 1D^2 sequencing produces two types of fastq
418 files, 1D and 1D^2; there was some redundancy between 1D and 1D^2 reads. Therefore, redundant
419 1D reads were removed, and only 1D^2 reads were used. Moreover, some redundant 1D^2 reads were
420 found. These reads were removed from 1D^2 files and used as 1D reads. The percentage of the primary
421 aligned reads was 54.7% (Supplemental Table 2). This is because we did not filter out reads excluding
422 1D-fail reads of Rapid sequencing for alignment.
423
424 *SV calling*
425 Sniffles (version 1.0.7) with a parameter "-s 5" was used to call SVs. The minimum number of
426 supporting reads was determined such that we could detect the known large deletion of CDKN2A
427 (Suzuki et al. 2014), but alignment bias for a reference genome reduces the call rate of insertions

428 compared to that of deletions (Nattestad et al. 2018; Chaisson et al. 2015). We detected 11,316 records
429 as a VCF file of SVs, including CCDC6-RET, on the nuclear DNA of LC-2/ad cell line (Supplemental
430 Table 3). Illumina HiSeq 2000 Paired-end WGS data (DDBJ accession number, DRX015205) aligned
431 against GRCh38 by using bwa (version 7.15) (Li and Durbin 2009) were used for calling SVs by using
432 Lumpy (version 0.2.13) (Layer et al. 2014), delly (version 0.7.7) (Rausch et al. 2012), and manta
433 (version 1.0.3) (Chen et al. 2016). The results of these SV callers are listed in Supplemental Table 5.
434 The four lists of SV candidates detected using Sniffles, Lumpy, delly, and manta were merged using
435 SURVIVOR (version 1.0.0) (Jeffares et al. 2017) with "5 sv_lists 1000 1 1 0 0" option for clustering
436 SV candidates (Supplemental Fig. 3). Further, we filtered the merged candidates based on the
437 following three criteria: (1) Remove SVs of 0 bp to 1 kbp in length to focus on large SVs, (2) Remove
438 all insertions and non-canonical SV types: INVDUP, DEL/INV, and DUP/INS (technically
439 unnecessary, but we aimed at cross-validating SV candidates from Illumina reads with which
440 insertions and non-canonical SV types are hard to call), and (3) Remove SVs overlapping with the
441 intervals of the 10X default blacklist
442 (https://github.com/10XGenomics/supernova/blob/master/tenkit/lib/python/tenkit/sv_data/10X_GRC
443 h38_no_alt_decoy/default_sv_blacklist.bed) for reducing false positives. After filtering, we obtained
444 1,790 SV records.
445
446 *Constructing a variation graph*
447 We constructed a variation graph, including a reference genome (GRCh38) and an individual genome
448 (either LC-2/ad or CHM1). We developed scripts that we call MoMI-G tools. MoMI-G tools follow
449 the procedure in SplitThreader Graph (Nattestad et al. 2016a).
450
451 1. Extract a breakpoint list from a VCF file generated by Sniffles.

452 2. Construct an initial variation graph with each chromosome of the GRCh38 primary assembly as
453 a single node.
454 3. Split nodes every 1 Mbp due to the limitation in the implementation of vg used during the
455 development of MoMI-G; otherwise, vg aborted with an error. The latest version of vg does not
456 have this limitation.
457 4. For each breakpoint in the breakpoint list, split the node of the graph that contains the breakpoint
458 into two nodes at the breakpoint.
459 5. Create paths that represent the chromosomes in the reference genome.
460 6. Create paths that represent SVs, each of which corresponds to one record in the input VCF file.
461 Further, add a node to the graph when the type of an SV is an insertion.
462
463 The representation of SVs varies between SV callers: the same SV can be described in different ways
464 in the current VCF format. Although vg has a "construct" subcommand that constructs a variation
465 graph from a pair of a reference genome and a VCF file, "vg construct" is incompatible with the output
466 of SV callers we know of for the following reason. For example, an INV record in VCF means either
467 a single inversion (both ends included) or only one end of the inversion, depending on the
468 implementation of SV callers. We wrote custom scripts, MoMI-G tools, for Sniffles.
469 In general, to display read alignments, we recommend aligning against a variation graph by
470 directly using "vg map" instead of converting read alignments against the linear reference genomes
471 into the GAM format. Nevertheless, we intended to inspect the results of Sniffles; therefore, we wrote
472 a custom script to convert alignments against the linear reference (BAM file) into a GAM file by using
473 "vg annotate," during which CIGAR is lost; "vg annotate" was originally designed for placing
474 annotations in BED/GFF files on paths on variation graphs, and not for alignments. One possible
475 solution to this issue is making SV callers graph-aware.

476

477 *Inspection of SV candidates*

478 We modified and integrated SequenceTubeMap into MoMI-G so that it can visualize a variation graph
479 converted from SVs for showing the difference between a reference genome and an individual genome.
480 The modifications made to SequenceTubeMap are shown in Supplemental Table 6. One can click on
481 the download button for downloading an SVG image generated by SequenceTubeMap so that a vector
482 image can be used for publishing.

483

484 *Visualizing annotations*

485 Ideally, MoMI-G provides annotations on variation graphs. However, annotations available in public
486 databases are for the linear reference genome. MoMI-G can display annotations in bigWig/bigBed
487 formats. In particular, for human reference genomes, GRCh19/hg37 and GRCh38/hg38, MoMI-G
488 provides an interface for retrieving Ensembl gene annotations from the Ensembl SPARQL endpoint
489 (Jupp et al. 2014) via SPARQList REST API (https://github.com/dbcls/sparqlist). The orientation of
490 genes is shown in the legend of SequenceTubeMap. Further, if one clicks on a gene name, the website
491 of the gene information in TogoGenome (Katayama et al. 2019) opens.

492

493 *Miscellaneous modules*

494 Threshold Filter: Threshold Filter has two use cases. First, one can toggle checkboxes to select whether
495 to show inter-chromosomal SVs and/or intra-chromosomal SVs. Second, one can filter SVs by using
496 a slider based on the custom priority (possibly given by SV callers) of each SV.

497 Annotation Table: Annotation table shows all annotations that are displayed on the SequenceTubeMap.
498 Moreover, annotations can be downloaded as a BED file.

499 Linear Genome Browser: To provide a compatible view of a selected genomic region, we integrated

Pileup.js (Vanderkam et al. 2016) into MoMI-G.

Backend server

The MoMI-G backend server caches subgraphs once a client requests a genomic interval on a path. It then retrieves annotations from bigWig and bigBed with a range query and provides JSON API with which the client can make queries.

DATA ACCESS

Newly obtained long-read sequencing data of LC-2/ad were deposited in the DDBJ with accession numbers DRA007941 (DRX156303-DRX156310). Datasets included in this article are also provided in the database, DBTSS/DBKERO (Suzuki et al. 2018). The source code of MoMI-G is available at https://github.com/MoMI-G/MoMI-G/ under the MIT license. Further, we have included an example dataset and annotations as a Docker image.

ACKNOWLEDGMENTS

We thank all testers who provided testing and feedbacks for MoMI-G. We also thank Kazuyuki Shudo at the Tokyo Institute of Technology and Toshiaki Katayama at the Database Center for Life Science (DBCLS) of Research Organization of Information and Systems (ROIS), Japan, for their useful and insightful discussions, Sarun Sereewattanawoot at the University of Tokyo for initial contributions to Illumina read alignment. This work was supported in part by Information-technology Promotion Agency, Japan (IPA) and Exploratory IT Human Resources Project (The MITOU Program) in the fiscal year 2017 and in part by JSPS KAKENHI (Grant Number, 16H06279).

523 ## AUTHOR CONTRIBUTIONS

524 All authors were involved in study design; TY wrote all codes of MoMI-G; Y. Sakamoto preformed
525 MinION sequencing and SV analysis on LC-2/ad; TY, Y. Sakamoto, and MK drafted the manuscript;
526 MS and Y. Suzuki supervised the sequencing analysis; and MK supervised the study. All authors read
527 and approved the final manuscript.

528

529 ## DISCLOSURE DECLARATION

530 The authors declare that there is no competing financial interest.

531

532 ## REFERENCES

533

534 Ahdesmäki MJ, Chapman BA, Cingolani P, Hofmann O, Sidoruk A, Lai Z, Zakharov G,
535 Rodichenko M, Alperovich M, Jenkins D, et al. 2017. Prioritisation of structural
536 variant calls in cancer genomes. *PeerJ* **5**: e3166.

537 Belyeu JR, Nicholas TJ, Pedersen BS, Sasani TA, Havrilla JM, Kravitz SN, Conway ME,
538 Lohman BK, Quinlan AR, Layer RM. 2018. SV-plaudit: A cloud-based framework for
539 manually curating thousands of structural variants. *Gigascience* **7**: 265058.

540 Bressler R, Lin J, Eakin A, Robinson T, Kreisberg R, Rovira H, Knijnenburg T, Boyle J,
541 Shmulevich I. 2012. Fastbreak: A tool for analysis and visualization of structural
542 variations in genomic data. *EURASIP J Bioinform Syst Biol* **2012**: 15.

543 Buels R, Yao E, Diesh CM, Hayes RD, Munoz-Torres M, Helt G, Goodstein DM, Elsik CG,
544 Lewis SE, Stein L, et al. 2016. JBrowse: A dynamic web platform for genome
545 visualization and analysis. *Genome Biol* **17**: 1–12.

546 Chaisson MJP, Huddleston J, Dennis MY, Sudmant PH, Malig M, Hormozdiari F,

547 Antonacci F, Surti U, Sandstrom R, Boitano M, et al. 2015. Resolving the complexity
548 of the human genome using single-molecule sequencing. *Nature* **517**: 608–611.
549 Chen X, Schulz-Trieglaff O, Shaw R, Barnes B, Schlesinger F, Källberg M, Cox AJ,
550 Kruglyak S, Saunders CT. 2016. Manta: Rapid detection of structural variants and
551 indels for germline and cancer sequencing applications. *Bioinformatics* **32**: 1220–1222.
552 Collins RL, Brand H, Redin CE, Hanscom C, Antolik C, Stone MR, Glessner JT, Mason T,
553 Pregno G, Dorrani N, et al. 2017. Defining the diverse spectrum of inversions, complex
554 structural variation, and chromothripsis in the morbid human genome. *Genome Biol*
555 **18**: 36.
556 Colmsee C, Beier S, Himmelbach A, Schmutzer T, Stein N, Scholz U, Mascher M. 2015.
557 BARLEX—the Barley Draft Genome Explorer. *Mol Plant* **8**: 964–966.
558 Gansner ER, North SC. 1999. An open graph visualization system and its applications.
559 *Softw—Pract Exp* **30**: 1203–1233.
560 Garrison E, Sirén J, Novak AM, Hickey G, Eizenga JM, Dawson ET, Jones W, Garg S,
561 Markello C, Lin MF, et al. 2018. Variation graph toolkit improves read mapping by
562 representing genetic variation in the reference. *Nat Biotechnol* **36**: 875–879.
563 Guan P, Sung W-K. 2016. Structural variation detection using next-generation sequencing
564 data. *Methods* **102**: 36–49.
565 Halper-Stromberg E, Steranka J, Burns KH, Sabunciyan S, Irizarry RA. 2014.
566 Visualization and probability-based scoring of structural variants within repetitive
567 sequences. *Bioinformatics* **30**: 1514–1521.
568 Herbig A, Jäger G, Battke F, Nieselt K. 2012. GenomeRing: Alignment visualization based
569 on SuperGenome coordinates. *Bioinformatics* **28**: 7–15.
570 Hiltemann S, McClellan EA, Van Nijnatten J, Horsman S, Palli I, Alves IT, Hartjes T,

571 Trapman J, Van Der Spek P, Jenster G, et al. 2013. iFUSE: Integrated fusion gene
572 explorer. *Bioinformatics* **29**: 1700–1701.
573 Jeffares DC, Jolly C, Hoti M, Speed D, Shaw L, Rallis C, Balloux F, Dessimoz C, Bähler J,
574 Sedlazeck FJ. 2017. Transient structural variations have strong effects on
575 quantitative traits and reproductive isolation in fission yeast. *Nat Commun* **8**: 1–11.
576 Jupp S, Malone J, Bolleman J, Brandizi M, Davies M, Garcia L, Gaulton A, Gehant S,
577 Laibe C, Redaschi N, et al. 2014. The EBI RDF platform: Linked open data for the life
578 sciences. *Bioinformatics* **30**: 1338–1339.
579 Katainen R, Donner I, Cajuso T, Kaasinen E, Palin K, Mäkinen V, Aaltonen LA, Pitkänen
580 E. 2018. Discovery of potential causative mutations in human coding and noncoding
581 genome with the interactive software BasePlayer. *Nat Protoc* **13**: 2580–2600.
582 Katayama T, Kawashima S, Okamoto S, Moriya Y, Chiba H, Naito Y, Fujisawa T, Mori H,
583 Takagi T. 2019. TogoGenome/TogoStanza: modularized Semantic Web genome
584 database. *Database* **2019**: 1–11.
585 Kent WJ, Sugnet CW, Furey TS, Roskin KM, Pringle TH, Zahler AM, Haussler D. 2002.
586 The Human Genome Browser at UCSC. *Genome Res* **12**: 996–1006.
587 Kent WJ, Zweig AS, Barber G, Hinrichs AS, Karolchik D. 2010. BigWig and BigBed:
588 Enabling browsing of large distributed datasets. *Bioinformatics* **26**: 2204–2207.
589 Kim P-G, Cho H-G, Park K. 2008. A scaffold analysis tool using mate-pair information in
590 genome sequencing. *J Biomed Biotechnol* **2008**: 1–7.
591 Krzywinski M, Schein J, Birol I, Connors J, Gascoyne R, Horsman D, Jones SJ, Marra MA.
592 2009. Circos: An information aesthetic for comparative genomics. *Genome Res* **19**:
593 1639–1645.
594 Layer RM, Chiang C, Quinlan AR, Hall IM. 2014. LUMPY: a probabilistic framework for

structural variant discovery. *Genome Biol* **15**: R84.

Li H, Durbin R. 2009. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25**: 1754–1760.

Matsubara D, Kanai Y, Ishikawa S, Ohara S, Yoshimoto T, Sakatani T, Oguni S, Tamura T, Kataoka H, Endo S, et al. 2012. Identification of CCDC6-RET fusion in the human lung adenocarcinoma cell line, LC-2/ad. *J Thorac Oncol* **7**: 1872–1876.

Mertens F, Johansson B, Fioretos T, Mitelman F. 2015. The emerging complexity of gene fusions in cancer. *Nat Rev Cancer* **15**: 371–381.

Mizukami T, Shiraishi K, Shimada Y, Ogiwara H, Tsuta K, Ichikawa H, Sakamoto H, Kato M, Shibata T, Nakano T, et al. 2014. Molecular mechanisms underlying oncogenic RET fusion in lung adenocarcinoma. *J Thorac Oncol* **9**: 622–630.

Munro JE, Dunwoodie SL, Giannoulatou E. 2017. SVPV: A structural variant prediction viewer for paired-end sequencing datasets. *Bioinformatics* **33**: 2032–2033.

Murphy C, Elemento O. 2016. AGFusion: Annotate and visualize gene fusions. *bioRxiv* 1–4.

Naquin D, D'Aubenton-Carafa Y, Thermes C, Silvain M. 2014. CIRCUS: A package for Circos display of structural genome variations from paired-end and mate-pair sequencing data. *BMC Bioinformatics* **15**: 198.

Nattestad M, Alford MC, Sedlazeck FJ, Schatz MC. 2016a. SplitThreader : Exploration and analysis of rearrangements in cancer genomes. *bioRxiv* 1–8.

Nattestad M, Chin C-S, Schatz MC. 2016b. Ribbon: Visualizing complex genome alignments and structural variation. *bioRxiv* **0344**: 82123.

Nattestad M, Goodwin S, Ng K, Baslan T, Sedlazeck FJ, Rescheneder P, Garvin T, Fang H, Gurtowski J, Hutton E, et al. 2018. Complex rearrangements and oncogene amplifications revealed by long-read DNA and RNA sequencing of a breast cancer cell

line. *Genome Res* **28**: 1126–1135.

Nielsen CB, Jackman SD, Birol I, Jones SJM. 2009. ABySS-explorer: Visualizing genome sequence assemblies. *IEEE Trans Vis Comput Graph* **15**: 881–888.

O'Brien T, Ritz A, Raphael B, Laidlaw D. 2010. Gremlin: An interactive visualization model for analyzing genomic rearrangements. *IEEE Trans Vis Comput Graph* **16**: 918–926.

Paten B, Eizenga JM, Rosen YM, Novak AM, Garrison E, Hickey G. 2018. Superbubbles, Ultrabubbles, and Cacti. *J Comput Biol* **25**: 649–663.

Paten B, Novak AM, Eizenga JM, Garrison E. 2017. Genome graphs and the evolution of genome inference. *Genome Res* **27**: 665–676.

Quinlan AR, Hall IM. 2012. Characterizing complex structural variation in germline and somatic genomes. *Trends Genet* **28**: 43–53.

Rausch T, Zichner T, Schlattl A, Stütz AM, Benes V, Korbel JO. 2012. DELLY: Structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics* **28**: 333–339.

Reisle C, Mungall KL, Choo C, Paulino D, Bleile DW, Muhammadzadeh A, Mungall AJ, Moore RA, Shlafman I, Coope R, et al. 2018. MAVIS: Merging, Annotation, Validation, and Illustration of Structural variants. *Bioinformatics* 1–3.

Revanna K V, Munro D, Gao A, Chiu C-C, Pathak A, Dong Q, Frazer K, Elnitski L, Church D, Dubchak I, et al. 2012. A web-based multi-genome synteny viewer for customized data. *BMC Bioinformatics* **13**: 190.

Riba-grognuz O, Keller L, Falquet L, Xenarios I, Wurm Y. 2011. Visualization and quality assessment of de novo genome assemblies. *Bioinformatics* **27**: 3425–3426.

Sante T, Vergult S, Volders PJ, Kloosterman WP, Trooskens G, De Preter K, Dheedene A, Speleman F, De Meyer T, Menten B. 2014. ViVar: A comprehensive platform for the

643 analysis and visualization of structural genomic variation. *PLoS One* **9**: 1–12.

644 Santoro M, Carlomagno F. 2013. Central role of RET in thyroid cancer. *Cold Spring Harb*
645 *Perspect Biol* **5**: 1–17.

646 Sedlazeck FJ, Lee H, Darby CA, Schatz MC. 2018a. Piercing the dark matter:
647 Bioinformatics of long-range sequencing and mapping. *Nat Rev Genet* **19**: 329–346.

648 Sedlazeck FJ, Rescheneder P, Smolka M, Fang H, Nattestad M, von Haeseler A, Schatz
649 MC. 2018b. Accurate detection of complex structural variations using single-molecule
650 sequencing. *Nat Methods* **15**: 461–468.

651 Sereewattanawoot S, Suzuki A, Seki M, Sakamoto Y, Kohno T, Sugano S, Tsuchihara K,
652 Suzuki Y. 2018. Identification of potential regulatory mutations using multi-omics
653 analysis and haplotyping of lung adenocarcinoma cell lines. *Sci Rep* **8**: 4926.

654 Spies N, Zook JM, Salit M, Sidow A. 2015. Svviz: A read viewer for validating structural
655 variants. *Bioinformatics* **31**: 3994–3996.

656 Stankiewicz P, Lupski JR. 2010. Structural variation in the human genome and its role in
657 disease. *Annu Rev Med* **61**: 437–455.

658 Suzuki A, Kawano S, Mitsuyama T, Suyama M, Kanai Y, Shirahige K, Sasaki H, Tokunaga
659 K, Tsuchihara K, Sugano S, et al. 2018. DBTSS/DBKERO for integrated analysis of
660 transcriptional regulation. *Nucleic Acids Res* **46**: D229–D238.

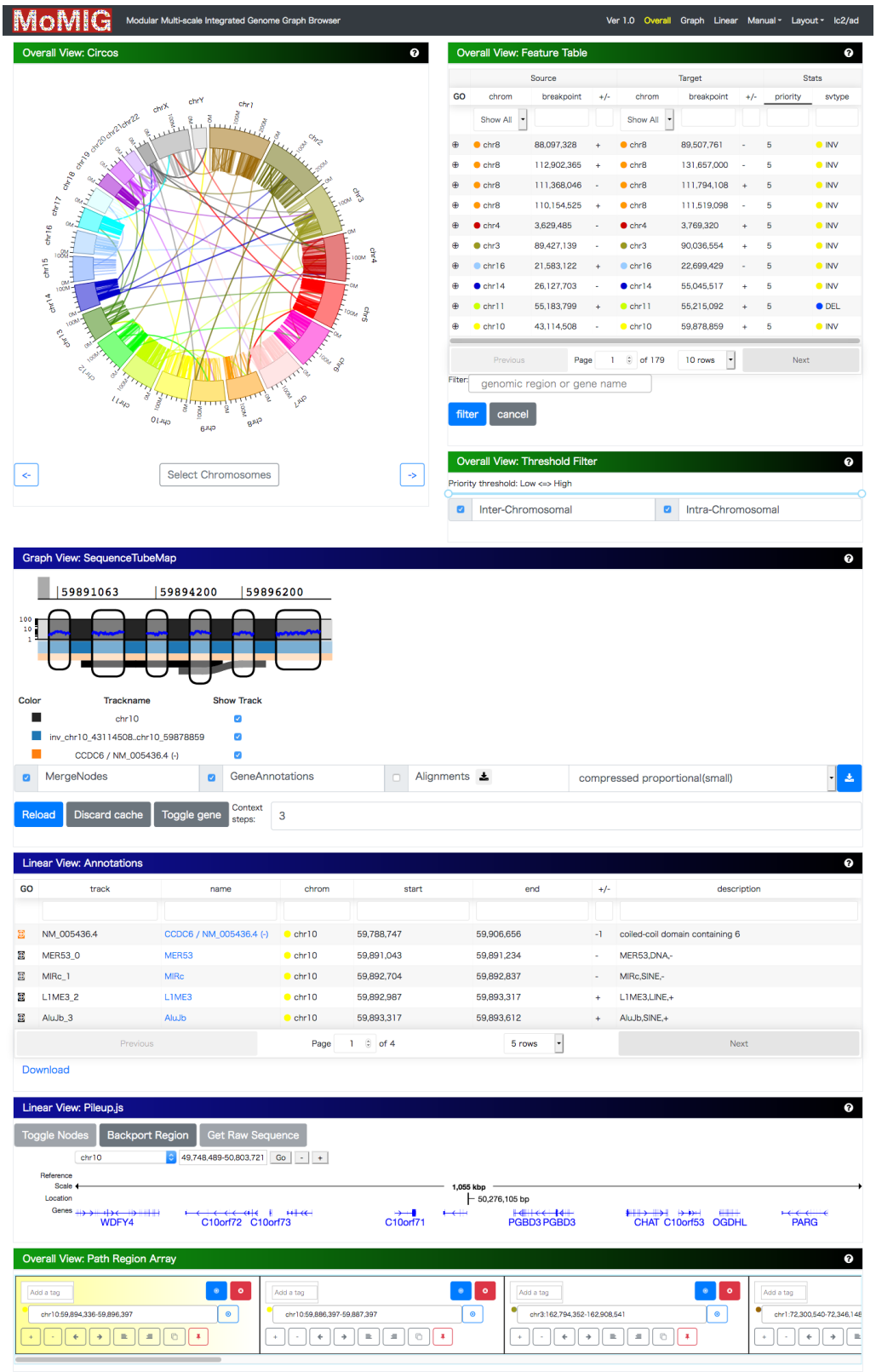
661 Suzuki A, Makinoshima H, Wakaguri H, Esumi H, Sugano S, Kohno T, Tsuchihara K,
662 Suzuki Y. 2014. Aberrant transcriptional regulations in cancers: Genome,
663 transcriptome and epigenome analysis of lung adenocarcinoma cell lines. *Nucleic*
664 *Acids Res* **42**: 13557–13572.

665 Suzuki A, Matsushima K, Makinoshima H, Sugano S, Kohno T, Tsuchihara K, Suzuki Y.
666 2015. Single-cell analysis of lung adenocarcinoma cell lines reveals diverse expression

667 patterns of individual cells invoked by a molecular target drug treatment. *Genome*
668 *Biol* **16**: 66.
669 Suzuki A, Suzuki M, Mizushima-Sugano J, Frith MC, Makałowski W, Kohno T, Sugano S,
670 Tsuchihara K, Suzuki Y. 2017. Sequencing and phasing cancer mutations in lung
671 cancers using a long-read portable sequencer. *DNA Res* **24**: 585–596.
672 Tang B, Wang Q, Yang M, Xie F, Zhu Y, Zhuo Y, Wang S, Gao H, Ding X, Zhang L, et al.
673 2013. ContigScape: A Cytoscape plugin facilitating microbial genome gap closing.
674 *BMC Genomics* **14**: 1–12.
675 Thorvaldsdóttir H, Robinson JT, Mesirov JP. 2013. Integrative Genomics Viewer (IGV):
676 High-performance genomics data visualization and exploration. *Brief Bioinform* **14**:
677 178–192.
678 Vanderkam D, Aksoy BA, Hodes I, Perrone J, Hammerbacher J. 2016. Pileup.js: A
679 JavaScript library for interactive and in-browser visualization of genomic data.
680 *Bioinformatics* **32**: 2378–2379.
681 Weischenfeldt J, Symmons O, Spitz F, Korbel JO. 2013. Phenotypic impact of genomic
682 structural variation: Insights from and for human disease. *Nat Rev Genet* **14**: 125–
683 138.
684 Wick RR, Schultz MB, Zobel J, Holt KE. 2015. Bandage: Interactive visualization of de novo
685 genome assemblies. *Bioinformatics* **31**: 3350–3352.
686 Wöste M, Dugas M. 2018. VIPER: A web application for rapid expert review of variant calls.
687 *Bioinformatics* **34**: 1928–1929.
688 Wyczalkowski MA, Wylie KM, Cao S, Mclellan MD, Flynn J, Huang M, Ye K, Fan X, Chen
689 K, Wendl MC, et al. 2017. BreakPoint Surveyor: A pipeline for structural variant
690 visualization. *Bioinformatics* **33**: 3121–3122.

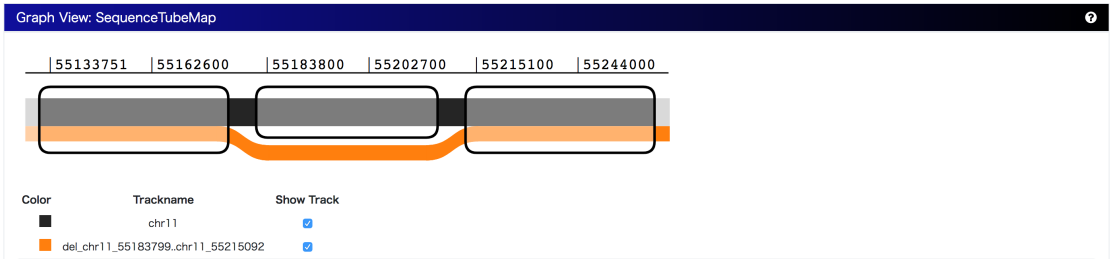
691

SUPPLEMENTAL FIGURES

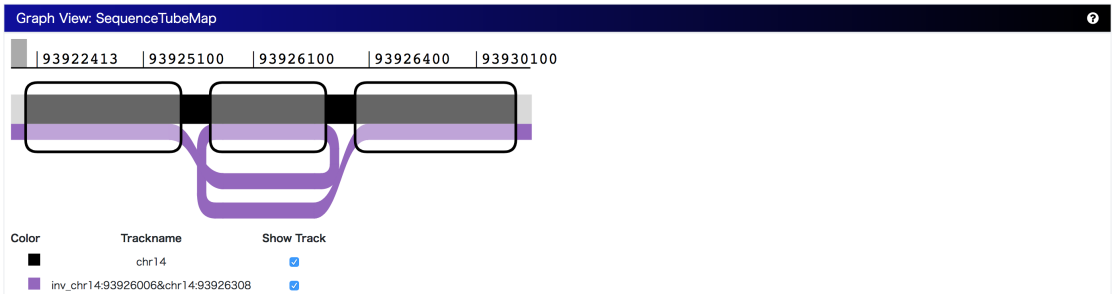


Supplemental Figure 1. A representative screenshot of MoMI-G with all view modules.

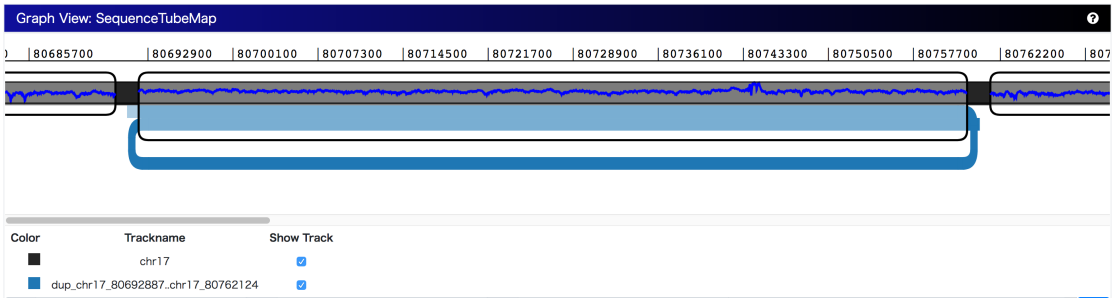
A



B



C

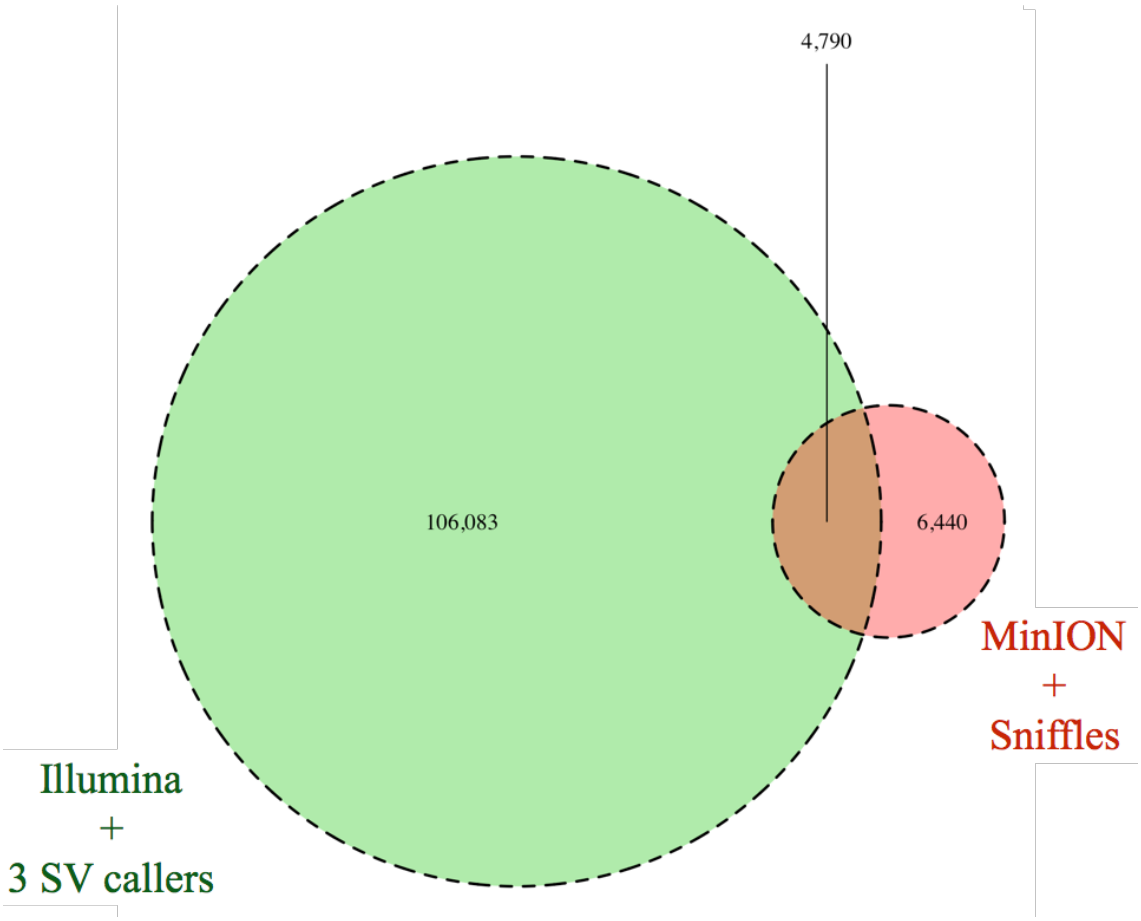


Supplemental Figure 2. Examples of a deletion, balanced inversion, and duplication

(A) Deletion: The orange line indicating a deletion starts before one breakend of the deletion, passes through the middle node that indicates the deleted sequence, and then proceeds for the sequence flanking the deletion.

(B) Balanced Inversion: The purple line indicating a balanced inversion includes flanking sequences on both breakpoints of the inversion.

(C) Duplication: The blue line indicating a duplication passes through the node twice, suggesting that the sequence of the node is duplicated. The line in the node might terminate if the node is interrupted by other SVs.



Supplemental Figure 3. Merged SVs from four SV callers by using SURVIVOR

The green circle describes the count of SVs called by Illumina, whereas the red circle presents the count called by MinION.

717 ## SUPPLEMENTAL TABLES

718 **Supplemental Table 1. List of MoMI-G features**

719

Module group	View module	Description
Chromosome-scale View	Circos Plot	Observe the distribution of SVs
	Feature Table	Select and filter SVs
	Threshold Filter	Filter SVs
	Interval Card Deck	Accumulate genomic intervals for inspection
Gene-scale View	SequenceTubeMap	Visualize a subgraph of a variation graph
	Interval Card Deck	Switch between the intervals
Nucleotide-scale View	Linear Browser	Visualize a linear genome with annotations
	Annotation Table	Show annotations
	SequenceTubeMap	Visualize nucleotides of a variation graph
	Interval Card Deck	Switch between the intervals

720

721 **Supplemental Table 2. Summary of the Oxford Nanopore sequencing data of LC-2/ad**

Number of reads	3,310,982
Coverage	12.8×
Percentage of aligned reads	57.9%
Average length of aligned reads	16,032.2 bp
Average identity of aligned reads	83.0%

722

723 **Supplemental Table 3. Called SVs in LC-2/ad nuclear DNA from Nanopore reads**

SV Types	Called SVs
DEL	6,366
DUP	323
INS	4,449
INV	99
INVDUP	1
TRA	79
DEL/INV	12
DUP/INS	3

724

725 **Supplemental Table 4. Breakends in the opposite end of the CCDC6-RET inversion**

Sequencer, aligner, and SV caller	Start	Stop
ONT MinION, NGM-LR + Sniffles	chr10: 42,849,939	chr10: 59,825,707
Illumina HiSeq 2000, BWA + manta	chr10: 42,849,938	chr10: 59,825,707
Illumina HiSeq 2000, BWA + delly	chr10:42,849,938	chr10:59,825,707
Illumina HiSeq 2000, BWA + lumpy	Not called	Not called

726

727

728

729

730

731 **Supplemental Table 5. Called SVs in LC-2/ad nuclear DNA from Illumina reads**

SV types	manta	delly	lumpy
BND	3,926	9,787	4,214
DEL	43,886	33,429	2,051
DUP	768	3,213	1,197
INS	29,216	2,212	-
INV	576	2,055	62

732

733 **Supplemental Table 6. The modifications made to the original SequenceTubeMap**

Original implementation	**Modified implementation**
Paths are displayed in the same order as they appear in the input.	A selected chromosome path is by default drawn as a straight horizontal line regardless of the path order in the input.
It was impossible to display genes and paths in different colors.	Genes, annotations, chromosomes, and variants can have different colors for easily distinguishing them.
The constant for the linear node width mode was too large for visualizing large SVs.	We implemented another mode suitable for visualizing large SVs.
No bigWig/bigBed support.	We added bigWig/bigBed support.
Non-consecutive genomic intervals are difficult to recognize.	We inserted a shaded gap between distant genomic intervals.

734

735