intansv: an R package for integrative analysis of structural variations

Lihua Jia[1,2,#], Na Liu[1,#], Fangfang Huang[1], Zhengfu Zhou[3], Xin He[1], Haoran Li[1], Zhizhan Wang[1], Wen Yao[1,4,*]

[1] National Key Laboratory of Wheat and Maize Crop Science, College of Life Sciences, Henan Agricultural University, Zhengzhou 450002, China

[2] National Key Laboratory of Wheat and Maize Crop Science, College of Agronomy, Henan Agricultural University, Zhengzhou 450002, China

[3] Wheat Research Institute, Henan Academy of Agricultural Sciences, Zhengzhou, Henan, 450002, China

[4] National Key Laboratory of Crop Genetic Improvement, National Center of Plant Gene Research (Wuhan), Huazhong Agricultural University, Wuhan 430070, China

Corresponding Author:

Wen Yao

Agricultural Road No. 63, Zhengzhou, Henan, 450002, China

Email address: yaowen@henau.edu.cn

[#] These authors contributed equally to this work.

Abstract

Identification of structural variations between individuals is very important for the understanding of phenotype variations and diseases. Despite the existence of dozens of programs for prediction of structural variations, none of them is the golden standard in this field and the results of multiple programs were usually integrated to get more reliable predictions. Annotation and visualization of structural variations are important for the understanding of their functions. However, no program provides these functions currently as far as we are concerned. We report an R package, intansv, which can integrate the predictions of multiple programs as well as annotate and visualize structural variations. The source code and the help manual of intansv is freely available at https://github.com/venyao/intansv and http://www.bioconductor.org/packages/devel/bioc/html/intansv.html.

Introduction

Next-generation sequencing (NGS) has greatly enhanced our ability to detect genomic variations between individuals which is the key to the understanding of phenotype variations and diseases. Although the third-generation sequencing technologies have emerged as new options to detect genomic variations, NGS is still the preferred approach concerning the sequencing cost and accuracy (Spealman et al. 2019). Genomic variation is comprised of single nucleotide polymorphisms (SNPs), insertions/deletions (indels) and structural variations (SVs). SNPs have been widely detected and utilized in quantitative trait locus (QTL) mapping and genome wide association study (GWAS) while SVs, which usually cause large effects, are more difficult to be detected (Alkan et al. 2011; Sadowski et al. 2019).

Currently, many programs have been developed to detect SVs based on NGS (Abyzov et al. 2011; Becker et al. 2018; Chen et al. 2009; Layer et al. 2014; Rausch et al. 2012). The algorithms employed by these programs could be categorized into three groups: (1) read-pair methods which analyze the span size and mapping orientation of paired-end reads and their inconsistency from expectation (Chen et al. 2009), (2) split-read algorithms which align the unmappable end of read pairs of which only one end can be mapped to the reference genome using the split-read alignment (Ye et al. 2009) and (3) read depth analysis which utilizes the increase and decrease in sequence coverage (Abyzov et al. 2011). However, the results of different tools have been reported to be discordant with each other and none of the currently existing programs is the golden standard in this field (Lin et al. 2015). As a result, the outputs of different tools were usually integrated to get more reliable predictions (Sudmant et al. 2015).

SVMerge (Wong et al. 2010), HugeSeq (Lam et al. 2012) and iSVP (Mimori et al. 2013) are tools to detect SVs by integrating different SV callers as part of their pipelines. BreakDancer

70   (Chen et al. 2009), Pindel (Ye et al. 2009), SECluster, RDXplorer (Yoon et al. 2009) and

71   RetroSeq (Keane et al. 2012) are integrated in the pipeline of SVMerge. HugeSeq and iSVP only

72   allow the use of supplied default callers. HugeSeq integrates four SV callers including

73   BreakDancer, Pindel, CNVnator (Abyzov et al. 2011) and BreakSeq (Lam et al. 2010) while

74   iSVP incorporates BreakDancer and Pindel as part of its pipeline. The installation and

75   configuration of these tools are challenging for users with no computing experience as different

76   SV callers are implemented with varying programming languages and are glued together by

77   these pipelines utilizing Linux shell, Perl and Python. To use these pipelines, the users must

78   successfully install several dependency software required by these pipelines in advance. In

79   addition, it is important to annotate SVs and visualize the distribution of SVs in the whole

80   genome or a specified genomic region. However, no software provides these functions

81   simultaneously for now.

82   We report intansv, an R package which is easy to install and use, aiming to integrate the

83   output of multiple popular programs, annotate the effects of SVs to genes and visualize the SVs

84   in the whole genome or specified genomic regions (Figure 1). The installation of intansv is

85   independent to the installation of SV callers as it only relies on the output of SV callers. intansv

86   supports the integration of the results of some or all seven prevalent SV callers.

87

Materials and Methods

89   The NGS data of Zhenshan 97 were downloaded from NCBI SRA (accession number

90   SRA012177) and were aligned to the rice Nipponbare reference genome (MSU version 7.0)

91   using BWA (version 0.6.1-r104) with default parameters (Kawahara et al. 2013; Li & Durbin

92   2009). The alignment result was processed by SAMtools and stored in BAM format (Li et al.

93   2009). The alignment result in BAM format and the reference genome sequence are essential

94   inputs for SV callers. Some of the SV callers use only the alignment result in BAM format as the

95   input file while some of the SV callers also need the reference genome sequence file as the input

96   file. All the seven SV callers were run with default parameters.

97   For each SV caller, various information was output to evaluate the quality of each predicted

98   SV, including read mapping quality, number of discordant read pairs supporting the SV, number

99   of split reads supporting the SV, etc. The information was utilized to filter low quality

100   predictions of these programs. For example, SVs predicted by BreakDancer with fewer than 3

101   discordant read pairs support were filtered by intansv by default. Each SV was assigned a score

102   representing the quality of the SV by BreakDancer and SVs with a score smaller than 60 were

103   filtered by intansv. In addition, SVs shorter than 100 bp or longer than 10 Mb predicted by any

104   SV caller were also filtered by intansv by default. For the results of Pindel, SVs supported by

105 fewer than 3 split reads were filtered. For the results of SoftSearch, SVs supported by fewer than
106 3 split reads or supported by fewer than 3 discordant read pairs were filtered.

107 To cluster SVs predicted by different SV callers, we first combined the same type of SVs of
108 different SV callers and detected the coordinate overlapping between SVs of the same type. The
109 findOverlaps function of the R package GenomicRanges was used to detect the coordinate
110 overlapping between different SVs, which could be represented as genomic ranges (Lawrence et
111 al. 2013). Then the coordinate overlapping percentage between pair-wise overlapped SVs was
112 calculated. We further define the distance between pair-wise SVs as 0 if they have a reciprocal
113 coordinate overlap of more than 80%. The distance between non-overlapped pair-wise SVs or
114 pair-wise SVs with a reciprocal coordinate overlap of no more than 80% was defined as 1. In this
115 way, a distance matrix could be constructed for combined SVs of the same type. We can then use
116 hierarchical clustering to divide the combined SVs of the same type into different clusters. A
117 cluster of SVs supported by two or more methods were then merged as a unified SV by taking
118 the mean of the start coordinates of all SVs in the cluster as the start coordinate of the merged
119 SV and taking the mean of the end coordinates of all SVs in the cluster as the end coordinate of
120 the merged SV. Any cluster of SVs supported by only one method were discarded.

121

122 Results

123 The intansv package takes the predictions of multiple SV callers as input and outputs an
124 integrated list of SVs. To use intansv, the users need to run SV callers and prepare the output
125 files as input to intansv. Here, we use the NGS data of a rice variety Zhenshan 97 to illustrate the
126 versatility of intansv (Materials and Methods) (Xie et al. 2010). Seven different programs,
127 BreakDancer (Chen et al. 2009), CNVnator (Abyzov et al. 2011), DELLY (Rausch et al. 2012),
128 Pindel (Ye et al. 2009), SVseq2 (Zhang et al. 2012), Lumpy (Layer et al. 2014) and SoftSearch
129 (Hart et al. 2013) were used with default parameters to predict SVs based on the alignment of the
130 NGS data of Zhenshan 97 to the Nipponbare reference genome (Table 1). The Nipponbare
131 reference genome is composed of 374.5 million nucleotide base pairs with a GC content of
132 43.5%. The NGS data of Zhenshan 97 provided over 14× coverage of the Nipponbare reference
133 genome.

134 The first step to use intansv is the reading of the output of SV callers. The intansv package
135 provides reusable and efficient tools to deal with the outputs of the seven programs as they are
136 tedious and not user-friendly for downstream analysis (Figure 1). During the reading process,
137 low quality predictions given by these programs were filtered out by intansv (Table 1, Materials
138 and Methods). In addition, overlapped predictions output by a single SV caller would be resolved
139 by intansv.

140 BreakDancer is able to predict deletions and inversions. The prediction of all type of SVs

141   were provided as a single file by BreakDancer. The output of BreakDancer contains 9010

142   deletions and 1202 inversions in the whole genome of Zhenshan 97 (Table 1). The following R

143   script shows the processing of the output of BreakDancer by intansv. The parameter

144   breakdancer.file.path stores the file path of the output file of BreakDancer in the disk. A total of

145   6890 deletions and 153 inversions in the whole genome of Zhenshan 97 were read into R as an R

146   list by intansv after filtering low quality predictions (1034 deletions and 24 inversions for

147   chromosome chr05 and chr10 in this demo). The evaluated score and evidence supporting each

148   SV were also read into R. Each element of the R list is an R data frame containing the results of a

149   type of SV, which can be easily wrote into the disk as text files using the write.table function of

150   R.

151   > breakdancer <- readBreakDancer(breakdancer.file.path)

152   > str(breakdancer)

153   List of 2

154         $ del:'data.frame': 1034 obs. of 5 variables:

155   ..$ chromosome: chr [1:1034] "chr05" "chr05" "chr05" "chr05" ...

156   ..$ pos1 : int [1:1034] 65575 86452 120264 153666 201845 208214 230521 276048 ...

157   ..$ pos2 : int [1:1034] 65949 87419 127693 153829 201959 208577 230654 276182 ...

158   ..$ size : num [1:1034] 353 988 7382 107 114 ...

159   ..$ info : chr [1:1034] "score=88;PE=5" "score=99;PE=20" "score=83;PE=6" ...

160         $ inv:'data.frame': 24 obs. of 5 variables:

161   ..$ chromosome: chr [1:24] "chr05" "chr05" "chr05" "chr05" ...

162   ..$ pos1 : int [1:24] 1291603 6942362 12014477 18770973 20583909 29431819 ...

163   ..$ pos2 : int [1:24] 1291649 6944726 12016019 18771402 20584709 29432511 ...

164   ..$ size : num [1:24] -105 2186 1334 452 320 ...

165   ..$ info : chr [1:24] "score=99;PE=3" "score=99;PE=5" "score=99;PE=6" ...

166   - attr(*, "method")= chr "BreakDancer"

167

168   CNVnator is able to predict deletions and duplications. The final output of CNVnator

169   usually contains several files and each file is the output for a single chromosome. All these files

170   should be put in the same directory and the path of this directory should be input to the function

171   readCnvnator of intansv. Then the output of CNVnator will be read into R. The directory given

172   to readCnvnator should only contain the final output files of CNVnator. The output of CNVnator

173   contains 2806 deletions and 566 duplications in the whole genome of Zhenshan 97 (Table 1).

174   The following R script shows the processing of the output of CNVnator by intansv. The

175   parameter cnvnator.dir.path stores the directory path of all the output files of CNVnator in the

176   disk. A total of 2805 deletions and 566 duplications in the whole genome of Zhenshan 97 were

177   read into R (369 deletions and 113 duplications for chromosome chr05 and chr10 in this demo).

178   > cnvnator <- readCnvnator(cnvnator.dir.path)

179   > str(cnvnator)

180   List of 2

181    $ del:'data.frame': 369 obs. of 5 variables:

182   ..$ chromosome: chr [1:369] "chr05" "chr05" "chr05" "chr05" ...

183   ..$ pos1 : num [1:369] 104101 230401 348301 1089301 1524301 ...

184   ..$ pos2 : num [1:369] 110700 248400 350100 1103100 1529700 ...

185   ..$ size : num [1:369] 6599 17999 1799 13799 5399 ...

186   ..$ info : chr [1:369]

187 "eval1=0.0100681;eval2=1386200000;eval3=10.1189;eval4=1690690000" …

188    $ dup:'data.frame': 113 obs. of 5 variables:

189   ..$ chromosome: chr [1:113] "chr05" "chr05" "chr05" "chr05" ...

190   ..$ pos1 : num [1:113] 405001 973201 1346401 5036101 6419101 ...

191   ..$ pos2 : num [1:113] 408900 977700 1359000 5055600 6423300 ...

192   ..$ size : num [1:113] 3899 4499 12599 19499 4199 ...

193   ..$ info : chr [1:113] "eval1=0.0169421;eval2=0;eval3=0.00545604;eval4=0" ...

194   - attr(*, "method")= chr "CNVnator"

195

196   As SVs are much more complex to detect than single nucleotide variations, predictions of

197   different programs were usually integrated to get more reliable results. The second step to use

198   intansv is to integrate the predictions of multiple SV callers. The intansv package provides

199   efficient functions to cluster and merge the predictions of different programs (Materials and

200   Methods). Predictions of varying programs were first clustered together if they have reciprocal

201   coordinate overlap of more than a specified percentage (default value 80%) (Figure 2). Then

202   clusters supported by more than a predefined number (default value 2) of methods were merged

203   as one SV respectively while the rest clusters were discarded (Figure 2). The intansv package is

204   also able to integrate the predictions of other SV callers with appropriate file format provided.

205   By integrating the results of different SV callers, intansv provides the most reliable and

206   comprehensive SV predictions (Table 1). The following R script shows the integration of the

207   output of BreakDancer, Pindel, CNVnator, DELLY, SVseq2 by intansv. The integrated results

208   contain 851 deletions, 13 duplications and 16 inversions for chromosome chr05 and chr10 of the

209   Zhenshan 97 genome. A total of 6255 deletions, 139 duplications and 92 inversions in the whole

210   genome of Zhenshan 97 were detected by integrating the results of seven SV callers (Table 1,

211   Table 2). The average length of the 6255 deletions, 139 duplications and 92 inversions were

212   3817 bp (ranging from 103 bp to 312652 bp), 40174 bp (ranging from 109 bp to 898618 bp) and

213   92364 bp (ranging from 143 bp to 904615 bp), respectively (Figure 3A-C). The distribution of

214   SVs in different chromosomes is not even (Figure 3D).

215   > sv_all_methods <- methodsMerge(breakdancer, pindel, cnvnator, delly, svseq)

216   > str(sv_all_methods)

217   List of 3

218       $ del:'data.frame': 851 obs. of 5 variables:

219   ..$ chromosome: chr [1:851] "chr05" "chr05" "chr05" "chr05" ...

220   ..$ pos1 : num [1:851] 65576 86449 120352 208217 280413 ...

221   ..$ pos2 : num [1:851] 65949 87419 127768 208564 281198 ...

222   ..$ methods : chr [1:851] "BreakDancer:Delly" "BreakDancer:Pindel:Delly" ...

223   ..$ info : chr [1:851] "score=88;PE=5;SU=5" "score=99;PE=20:SR=11;score=42:SU=21" ...

224       $ dup:'data.frame': 13 obs. of 5 variables:

225   ..$ chromosome: chr [1:13] "chr05" "chr05" "chr05" "chr05" ...

226   ..$ pos1 : num [1:13] 120449 910600 1615000 3087200 5696556 ...

227   ..$ pos2 : num [1:13] 127870 911326 1616028 3090903 5699328 ...

228   ..$ methods : chr [1:13] "Pindel:Delly" "Pindel:Delly" "Pindel:Delly" "Pindel:Delly" ...

229   ..$ info : chr [1:13] "SR=7;score=14:SU=6" "SR=3;score=4:SU=12" ...

230       $ inv:'data.frame': 16 obs. of 5 variables:

231   ..$ chromosome: chr [1:16] "chr05" "chr05" "chr05" "chr05" ...

232   ..$ pos1 : num [1:16] 6372420 6942364 12014448 15089985 28357689 ...

233   ..$ pos2 : num [1:16] 6453749 6944728 12015959 15158230 28391740 ...

234   ..$ methods : chr [1:16] "Pindel:Delly" "BreakDancer:Delly" ..

235       $ info : chr [1:16] "SR=11;score=12:SU=6" "score=99;PE=5:SU=5" ...

236

237   Analysis of the effect of SVs to the structure of genes is essential for the understanding of

238   their functions. The third step to use intansv is to annotate the effect of SVs to the genome

239   structure. For each identified SV, intansv provides details on the overlapping between this SV

240   and genes or the elements of genes (Figure 4A). Functions of the R package GenomicRanges

241   was utilized in the intansv package to dissect the overlapping between SV and gene elements

242   (Lawrence et al. 2013). Based on the NGS data of Zhenshan 97, we identified 4772 genes

243   affected by deletions, 912 genes affected by duplications and 1338 genes affected by inversions

244   (Supplementary Table 1, Supplementary Table 2, Supplementary Table 3).

245   The final step to use intansv is to visualize the distribution of SVs in the whole genome or a

246   specified genomic region. The chromosomes were split into windows of specific size and the

247   number of SVs within each window were counted and displayed as circular bar plot (Figure 4B).

248   For a specified region, the SVs and genes in it were displayed as circular rectangles to show the

249 effects of SVs to the structures of genes and elements of genes (Figure 4C). Functions of the R

250 package ggbio were utilized to realize this functionality in intansv (Yin et al. 2012).

251

Discussion

253 We provide an R package, intansv, for integrative analysis of structural variations. It can process

254 raw output of seven prevalent programs, merge predictions of different programs, annotate

255 effects caused by SVs to gene and its elements, visualize SVs distribution in the whole genome

256 or a specified genomic region, and output an integrated SV list as a single file. Visualization of

257 SVs in the whole genome is helpful for identifying SV hotspots while visualization of SVs in

258 specific genomic regions is beneficial to uncover the potential effects of SVs through comparing

259 SVs with genomic features or other experimental data. The intansv package is a program with

260 incorporate functions to deal with miscellaneous output styles of different SV callers which has

261 great potential to be widely accepted by the users. The parameter values of all the functions of

262 intansv can be adjusted by the users to meet the criteria of different users. The source code and

263 the help manual of intansv are deposited in GitHub (https://github.com/venyao/intansv). The

264 functionalities of old versions of the intansv package and the example output file format of

265 different SV callers integrated by intansv is provided at https://venyao.github.io/intansv/.

266

Conclusion

268 The discovery and genotyping of SV is very important for understanding its role in phenotype

269 variations. Precise and comprehensive detection of SV is still difficult due to the complexity of

270 SVs. We developed the intansv package which is able to integrate the predictions of multiple SV

271 callers. New features were implemented in the intansv package including the annotation and

272 visualization of SVs. With the development of NGS and long read sequencing technology, more

273 and more programs for the prediction of SVs would be developed. We will further enhance the

274 functionalities of intansv in the future by implementing new functions to deal with the output of

275 more programs to liberate researchers from tedious work of data cleansing.

276

References

278 Abyzov A, Urban AE, Snyder M, and Gerstein M. 2011. CNVnator: An approach to discover, genotype,

279   and characterize typical and atypical CNVs from family and population genome sequencing.

280   *Genome Research* 21:974-984. 10.1101/gr.114876.110

281 Alkan C, Coe BP, and Eichler EE. 2011. Genome structural variation discovery and genotyping. *Nature*

282   *Reviews Genetics* 12:363-376. 10.1038/nrg2958

283 Becker T, Lee W-P, Leone J, Zhu Q, Zhang C, Liu S, Sargent J, Shanker K, Mil-homens A, Cerveira E,

284   Ryan M, Cha J, Navarro FCP, Galeev T, Gerstein M, Mills RE, Shin D-G, Lee C, and Malhotra A.

285   2018. FusorSV: an algorithm for optimally combining data from multiple structural variation

286   detection methods. *Genome Biology* 19:38. 10.1186/s13059-018-1404-6

287 Chen K, Wallis JW, McLellan MD, Larson DE, Kalicki JM, Pohl CS, McGrath SD, Wendl MC, Zhang Q,
288   Locke DP, Shi X, Fulton RS, Ley TJ, Wilson RK, Ding L, and Mardis ER. 2009. Breakdancer: an
289   algorithm for high-resolution mapping of genomic structural variation. *Nature Methods* 6:677-681.
290   10.1038/nmeth.1363

291 Hart SN, Sarangi V, Moore R, Baheti S, Bhavsar JD, Couch FJ, and Kocher J-PA. 2013. SoftSearch:
292   integration of multiple sequence features to identify breakpoints of structural variations. *PLOS*
293   *ONE* 8:e83356. 10.1371/journal.pone.0083356

294 Kawahara Y, de la Bastide M, Hamilton JP, Kanamori H, McCombie WR, Ouyang S, Schwartz DC, Tanaka
295   T, Wu J, Zhou S, Childs KL, Davidson RM, Lin H, Quesada-Ocampo L, Vaillancourt B, Sakai H,
296   Lee SS, Kim J, Numa H, Itoh T, Buell CR, and Matsumoto T. 2013. Improvement of the Oryza
297   sativa Nipponbare reference genome using next generation sequence and optical map data. *Rice*
298   6:4. 10.1186/1939-8433-6-4

299 Keane TM, Wong K, and Adams DJ. 2012. RetroSeq: transposable element discovery from next-generation
300   sequencing data. *Bioinformatics* 29:389-390. 10.1093/bioinformatics/bts697

301 Lam HYK, Mu XJ, Stutz AM, Tanzer A, Cayting PD, Snyder M, Kim PM, Korbel JO, and Gerstein MB.
302   2010. Nucleotide-resolution analysis of structural variants using BreakSeq and a breakpoint library.
303   *Nature Biotechnology* 28:47-55. 10.1038/nbt.1600

304 Lam HYK, Pan C, Clark MJ, Lacroute P, Chen R, Haraksingh R, O'Huallachain M, Gerstein MB, Kidd
305   JM, Bustamante CD, and Snyder M. 2012. Detecting and annotating genetic variations using the
306   HugeSeq pipeline. *Nature Biotechnology* 30:226-229. 10.1038/nbt.2134

307 Lawrence M, Huber W, Pagès H, Aboyoun P, Carlson M, Gentleman R, Morgan MT, and Carey VJ. 2013.
308   Software for Computing and Annotating Genomic Ranges. *PLOS Computational Biology*
309   9:e1003118. 10.1371/journal.pcbi.1003118

310 Layer RM, Chiang C, Quinlan AR, and Hall IM. 2014. LUMPY: a probabilistic framework for structural
311   variant discovery. *Genome Biology* 15:R84. 10.1186/gb-2014-15-6-r84

312 Li H, and Durbin R. 2009. Fast and accurate short read alignment with Burrows–Wheeler transform.
313   *Bioinformatics* 25:1754-1760. 10.1093/bioinformatics/btp324

314 Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R, and
315   Subgroup GPDP. 2009. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*
316   25:2078-2079. 10.1093/bioinformatics/btp352

317 Lin K, Smit S, Bonnema G, Sanchez-Perez G, and de Ridder D. 2015. Making the difference: integrating
318   structural variation detection tools. *Briefings in Bioinformatics* 16:852-864. 10.1093/bib/bbu047

319 Mimori T, Nariai N, Kojima K, Takahashi M, Ono A, Sato Y, Yamaguchi-Kabata Y, and Nagasaki M.
320   2013. iSVP: an integrated structural variant calling pipeline from high-throughput sequencing data.
321   *BMC Systems Biology* 7:S8. 10.1186/1752-0509-7-s6-s8

322 Rausch T, Zichner T, Schlattl A, Stütz AM, Benes V, and Korbel JO. 2012. DELLY: structural variant
323   discovery by integrated paired-end and split-read analysis. *Bioinformatics* 28:i333-i339.
324   10.1093/bioinformatics/bts378

325 Sadowski M, Kraft A, Szalaj P, Wlasnowolski M, Tang Z, Ruan Y, and Plewczynski D. 2019. Spatial
326   chromatin architecture alteration by structural variations in human genomes at the population scale.

327   *Genome Biology* 20:148. 10.1186/s13059-019-1728-x

328 Spealman P, Burrell J, and Gresham D. 2019. Nanopore sequencing undergoes catastrophic sequence

329   failure at inverted duplicated DNA sequences. *bioRxiv*:852665. 10.1101/852665

330 Sudmant PH, Rausch T, Gardner EJ, Handsaker RE, Abyzov A, Huddleston J, Zhang Y, Ye K, Jun G, Hsi-

331   Yang Fritz M, Konkel MK, Malhotra A, Stutz AM, Shi X, Paolo Casale F, Chen J, Hormozdiari F,

332   Dayama G, Chen K, Malig M, Chaisson MJP, Walter K, Meiers S, Kashin S, Garrison E, Auton A,

333   Lam HYK, Jasmine Mu X, Alkan C, Antaki D, Bae T, Cerveira E, Chines P, Chong Z, Clarke L,

334   Dal E, Ding L, Emery S, Fan X, Gujral M, Kahveci F, Kidd JM, Kong Y, Lameijer E-W, McCarthy

335   S, Flicek P, Gibbs RA, Marth G, Mason CE, Menelaou A, Muzny DM, Nelson BJ, Noor A, Parrish

336   NF, Pendleton M, Quitadamo A, Raeder B, Schadt EE, Romanovitch M, Schlattl A, Sebra R,

337   Shabalin AA, Untergasser A, Walker JA, Wang M, Yu F, Zhang C, Zhang J, Zheng-Bradley X,

338   Zhou W, Zichner T, Sebat J, Batzer MA, McCarroll SA, The Genomes Project C, Mills RE,

339   Gerstein MB, Bashir A, Stegle O, Devine SE, Lee C, Eichler EE, and Korbel JO. 2015. An

340   integrated map of structural variation in 2,504 human genomes. *Nature* 526:75-81.

341   10.1038/nature15394

342 Wong K, Keane TM, Stalker J, and Adams DJ. 2010. Enhanced structural variant and breakpoint detection

343   using SVMerge by integration of multiple detection methods and local assembly. *Genome Biology*

344   11:R128-R128. 10.1186/gb-2010-11-12-r128

345 Xie W, Feng Q, Yu H, Huang X, Zhao Q, Xing Y, Yu S, Han B, and Zhang Q. 2010. Parent-independent

346   genotyping for constructing an ultrahigh-density linkage map based on population sequencing.

347   *Proceedings of the National Academy of Sciences of the United States of America* 107:10578-

348   10583. 10.1073/pnas.1005931107

349 Ye K, Schulz MH, Long Q, Apweiler R, and Ning Z. 2009. Pindel: a pattern growth approach to detect

350   break points of large deletions and medium sized insertions from paired-end short reads.

351   *Bioinformatics* 25:2865-2871. 10.1093/bioinformatics/btp394

352 Yin T, Cook D, and Lawrence M. 2012. ggbio: an R package for extending the grammar of graphics for

353   genomic data. *Genome Biology* 13:R77. 10.1186/gb-2012-13-8-r77

354 Yoon S, Xuan Z, Makarov V, Ye K, and Sebat J. 2009. Sensitive and accurate detection of copy number

355   variants using read depth of coverage. *Genome Research* 19:1586-1592. 10.1101/gr.092981.109

356 Zhang J, Wang J, and Wu Y. 2012. An improved approach for accurate and efficient calling of structural

357   variations with low-coverage sequence data. *BMC Bioinformatics* 13:S6. 10.1186/1471-2105-13-

358   S6-S6

359

Figure 1

Figure 1. The pipeline of intansv for integrative analysis of structural variations (SV).

The raw outputs of SV callers are first processed by intansv to get user-friendly results. During this process, the low-quality predictions were filtered out. The processed outputs of different SV callers are then merged for downstream analysis, including annotation and visualization.
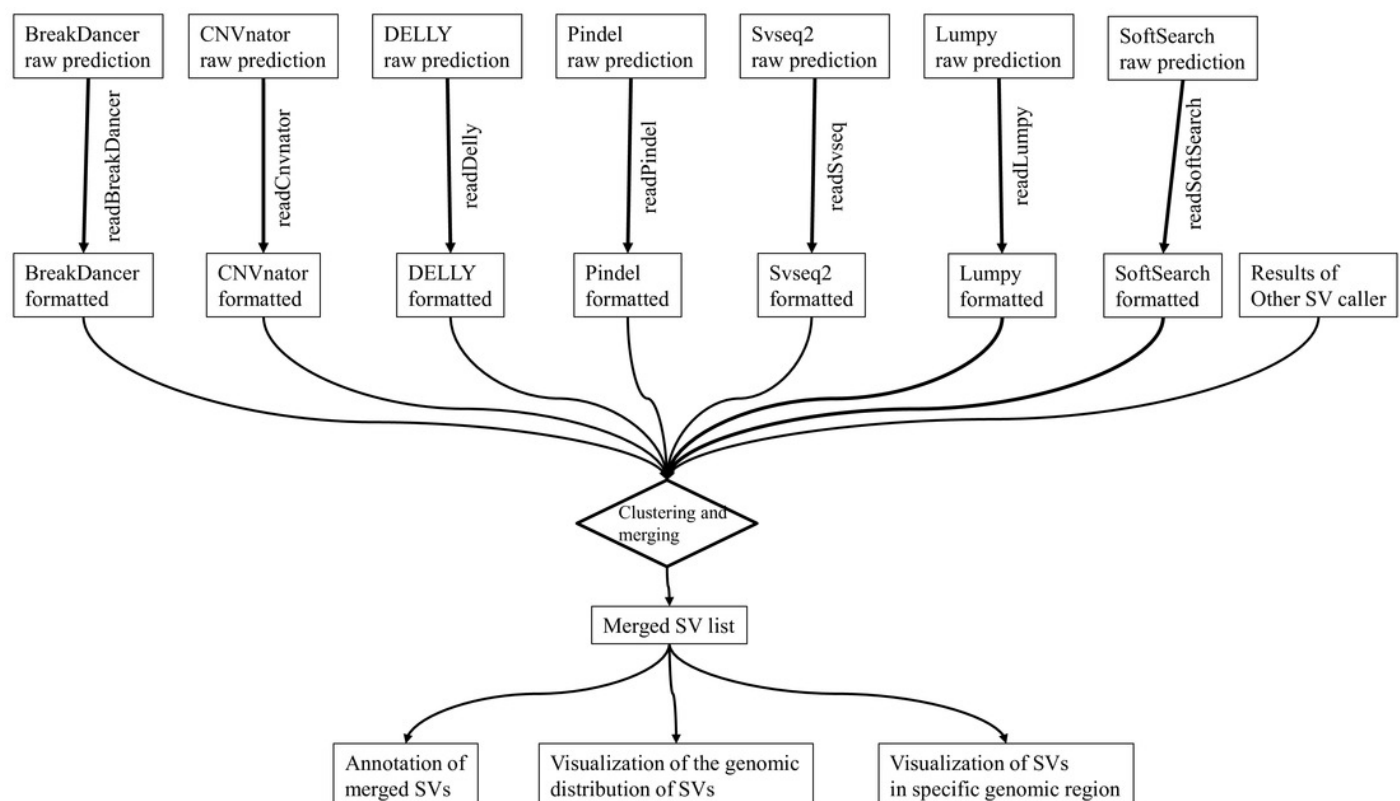
Figure 2

Figure 2. Merging the predictions of different SV callers.

Predictions of different programs were first clustered together if they have reciprocal coordinate overlap of more than a defined threshold (default 80%). Then clusters supported by more than two methods were merged as one SV (A and Cluster 1 in B) respectively while the rest clusters supported by no more than two methods were discarded (Cluster 2 in B).
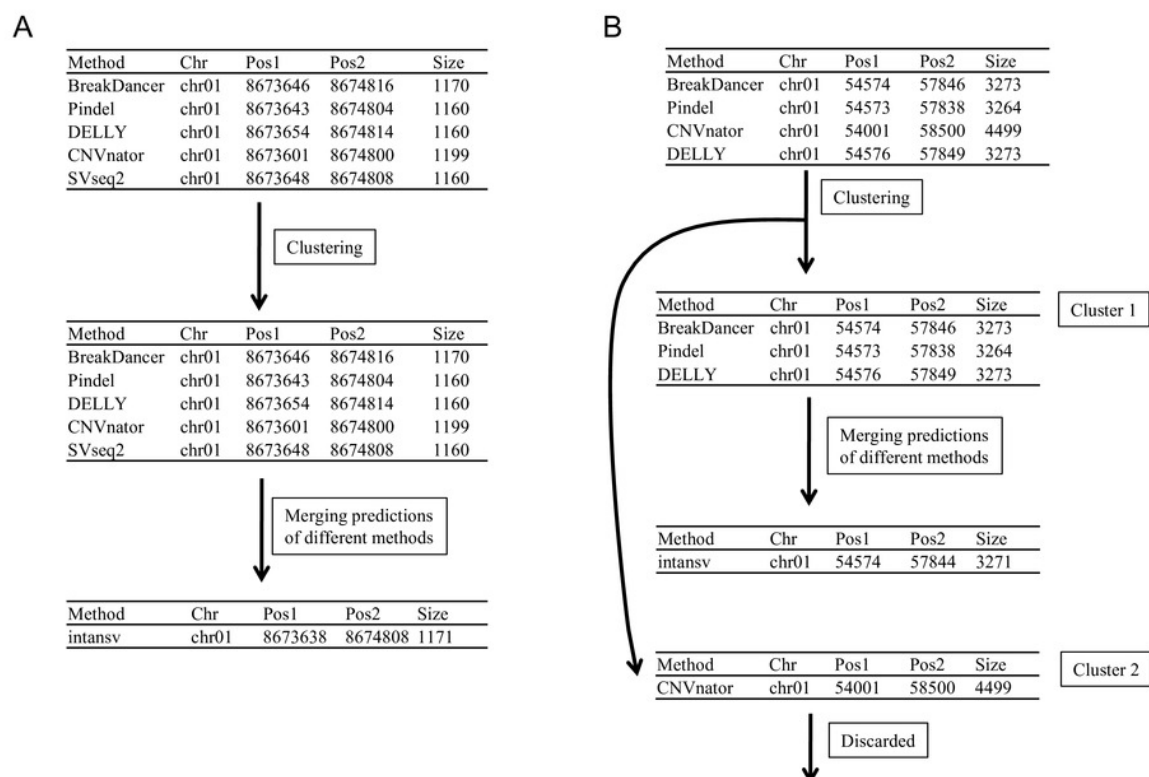
Figure 3

Figure 3. Lengths and genome distributions of SVs identified by intansv using the dataset of Zhenshan 97.

(A) Histogram of the lengths of deletions identified by intansv. (B) Histogram of the lengths of duplications identified by intansv. (C) Histogram of the lengths of inversions identified by intansv. (D) Comparison of the size of each chromosome with the number of SVs identified in each chromosome. Each chromosome is represented as a black point with the identifier labeled beside the corresponding point.

A

B

C

D
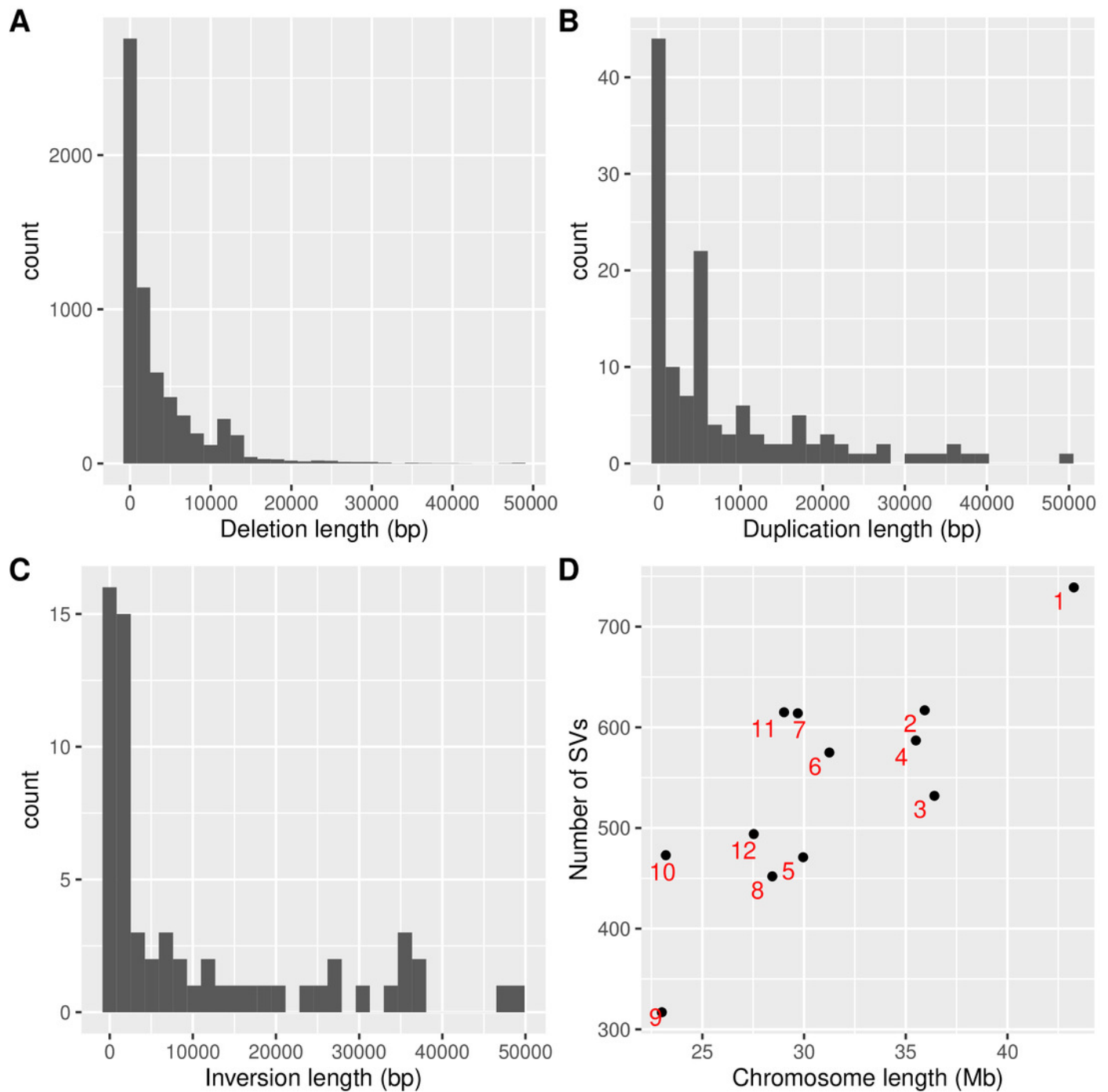
Figure 4

Figure 4. The demonstration of functions of intansv using the dataset of Zhenshan 97.

(A) The integrated output of intansv based on the predictions of seven programs including BreakDancer, Pindel, CNVnator, DELLY, Svseq2, Lumpy and SoftSearch. Chromosome, the chromosome ID of an SV. Pos1, the start coordinate of an SV. Pos2, the end coordinate of an SV. info, Information on an SV reported by different SV callers. (B) The distribution of SVs in the whole genome (blue, deletions; red, duplications; green, inversions). (C) The demonstration of SVs and genes in a specified genomic region (blue, genes; red, deletions; green, duplications).
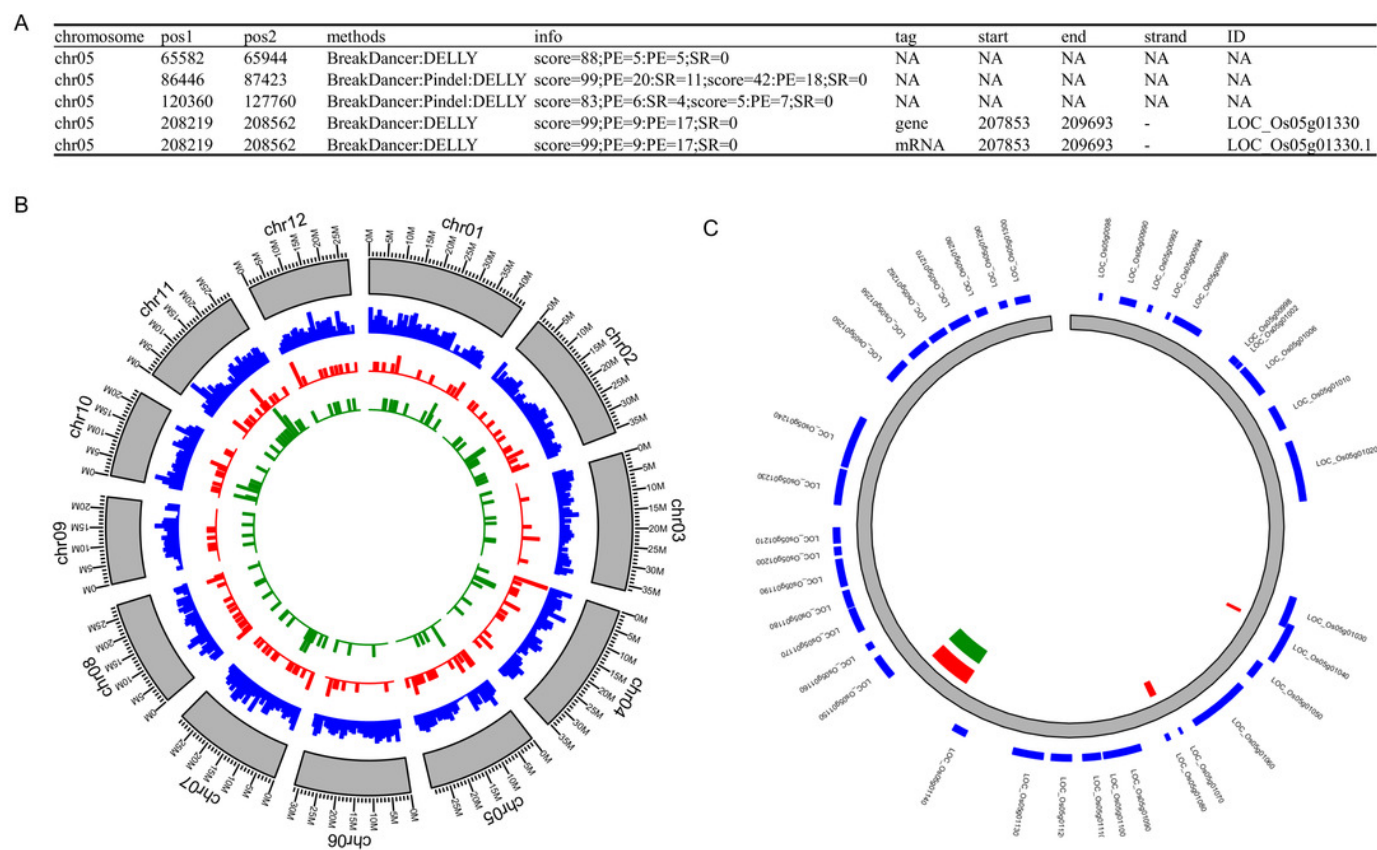
Table 1: Structural variations reported by different programs using NGS data of a rice variety Zhenshan 97.

Table 1: Structural variations reported by different programs using NGS data of a rice variety Zhenshan 97.

Methods	Deletion	Duplication	Inversion
BreakDancer raw predictions	9010	NA[a]	1202
BreakDancer formatted predictions	6890	NA	153
Pindel raw predictions	7641	3672	1724
Pindel formatted predictions	4322	561	1180
CNVnator raw predictions	2806	566	NA
CNVnator formatted predictions	2805	566	NA
Svseq raw predictions	7694	NA	NA
Svseq formatted predictions	1556	NA	NA
DELLY raw predictions	9034	497	1746
DELLY formatted predictions	5881	214	219
Lumpy raw predictions	7439	289	5
Lumpy formatted predictions	4684	109	3
SoftSearch raw predictions	580	14	86
SoftSearch formatted predictions	263	4	16
intansv	6255	139	92

[a] NA, this program doesn't support prediction of this type of SV.

Table 2: Integrated prediction of SVs by intansv based on the NGS data of a rice variety Zhenshan 97.

Table 2: Integrated prediction of SVs by intansv based on the NGS data of a rice variety Zhenshan 97.

Chromosome	Deletion	Duplication	Inversion
chr01	717	12	10
chr02	591	14	12
chr03	521	5	6
chr04	562	17	8
chr05	448	18	5
chr06	565	7	3
chr07	590	13	11
chr08	434	14	4
chr09	307	6	4
chr10	455	9	9
chr11	588	13	14
chr12	477	11	6