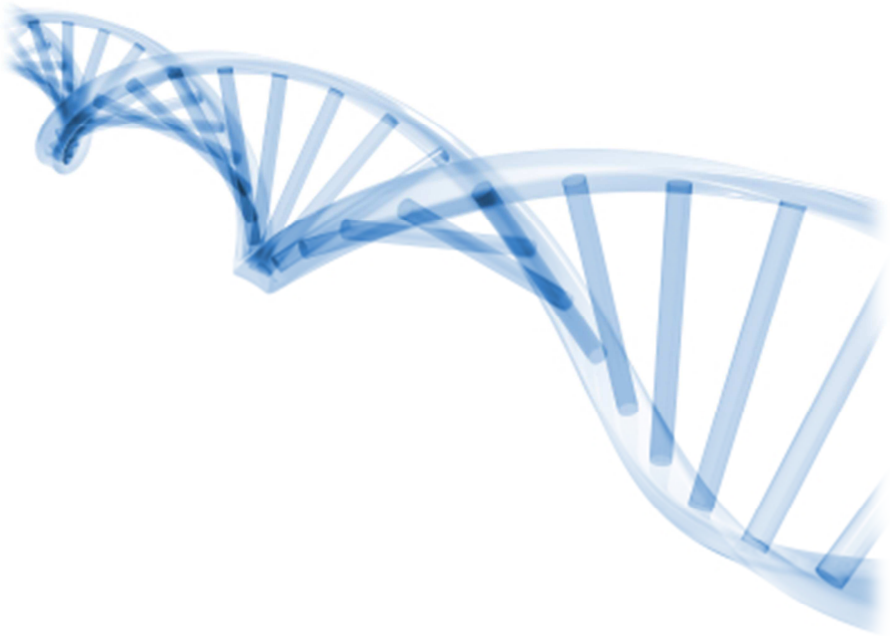
BENCHMARKING STRUCTURAL VARIATION DETECTION SOFTWARE

MINOR THESIS REPORT

Xi Wei



Supervised by: Ke Lin, Guusje Bonnema

April 23 – August 24 2012

Abstract

Genomic structural variations are important in their contribution to morphological variations for a wide range of traits in humans. Structural variants may have obvious phenotypic consequences, being associated with some complex diseases such as cancer. Recently, more interests have been grown on implications of structural variations for crop improvement and plant breeding. To detect high levels of structural variations in plants, robust software (caller) is required. Great advances on the next-generation sequencing techniques have helped dramatically for the discovery of structural variation. Most structural variation callers are freely available to the scientific community, and they are mainly based on three signatures: paired end distance, split-reads and read depth. However, each signature has its limitations. This thesis aims to evaluate available structural variation software and present a guideline to help users to choose callers for their re-sequencing projects. Benchmarking results suggested that the types and sizes of structural variants detected by individual caller were different. Performances of unique indels detection by Breakdancer_max, Clever, Pindel and SVDetect were size dependent. Pindel was the most favourable caller to detect very small indels (1-19bp), but its performance greatly decreased when the indel size increased. Clever was outstanding in predicting intermediate size indels (20-99 bp). Breakdancer_max better performed in detecting longer size indels (50 to 999 bp), while SVDetect was able to predict some long deletions. CnD and CNVnator could be used for downstream validation. To produce a more comprehensive set of calls, it is better to use a complementary method involving a variety of structural variation detection callers with different algorisms.

Keywords: Benchmarking, Structural variation detection, SV, Insertion, Deletion, Indels

Table of Contents

1. Introduction

1.1 Background

Brassica rapa, as one of the major commercial *Brassica* crops, provides vegetables and seed oil contributing to human nutrition. A remarkable morphological diversity has been found in *Brassica rapa*, but further research is still required to determine its underlying genetic basis [1-4]. Detection of genetic variants may provide new insights to explain morphological variation of target traits. Recently, paired end or mate pair methodologies help dramatically for the discovery of structural variants in expanding re-sequencing projects [5]. Generally speaking, Paired end data for the genome of interest (the donor, re-sequenced one) is generated by next-generation sequencing (NGS) platforms and read pairs are mapped to the reference genome. These projects require robust software to detect structural variations.

1.2 Methods to detect structural variations

Structural variants (SV) include insertions/deletions (indels), inversions, imbalance substitutions and other genomic structural rearrangements. They could be detected more accurately and straightforwardly if it was possible to directly assemble the donor's genome from the NGS reads. However, the very short length of reads make *de novo* assembly challenging, with the presence of repetitive sequences in large genomes [6-7]. Instead, current methods have been concentrated on analysing mapped reads compared to the reference. Based on different detection signatures (Figure 1), approaches for SV detection in software can be categorized into three groups: paired end distance/orientation, split reads and read depth [5].

The first group uses discordant pairs to detect SV (Figure 1a, 1b, 1c). Discordant pairs refer to paired ends with incorrect different mapped distance or orientation when it is mapped to the reference genome. Differences of mapping distance or orientation suggest structural variants. Drawbacks of methods based on this signature are that it can only provide approximate breakpoint regions, and it cannot detect very small SV events [8].

In split mapping, one end of a pair is mapped uniquely to the genome as anchor, while the other end is split. The second group uses split reads to predict SV, with single base resolution (Figure 1d, 1e). However, it has difficulty to map in the repetitive regions and it is less efficient to map split reads with large gaps [9].

The rationale of last method based on read depth, is that SV events would influence the frequency of mapped reads in a certain genomic region (Figure 1f). Although this signature could provide information about copy number variation (CNV), It needs high coverage, and it is poor at identifying exact SV events and breakpoints [9].

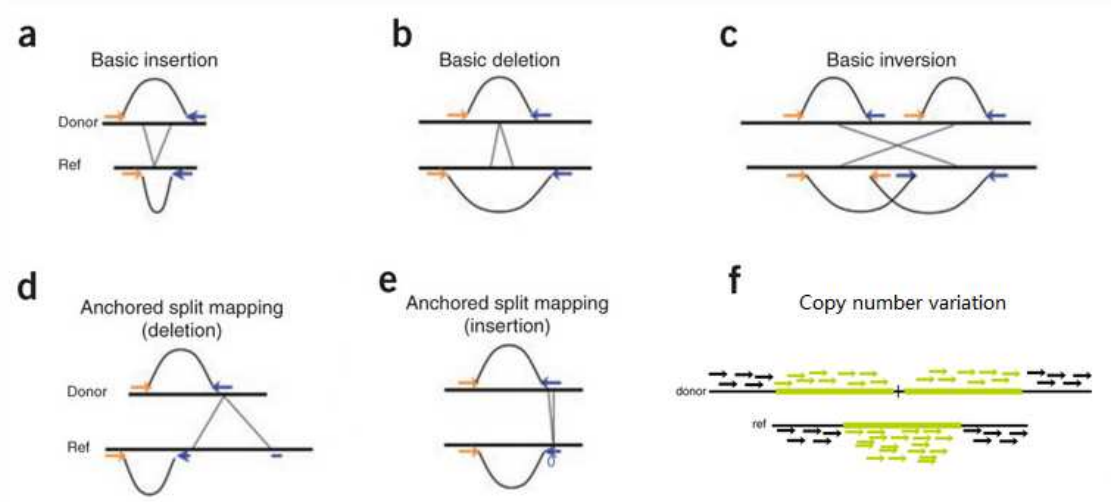


Figure 1 Illustrations of SV detection signatures [5-6]. Paired ends (orange and blue arrows) from the donor are ordered with opposite orientation and mapped to the reference. In the case of an insertion (a), the mapped distance is closer than expected. Conversely for the deletion (b), pairs will map farther. If the donor has an inversion, the order of pairs is preserved but one end of pairs inside the inversion changes its orientation. (d) and (e) are deletion and insertion signatures in split mapping. one end of a pair is mapped uniquely to the genome as anchor, while the other end is split. For a deletion (d), the prefix and suffix of the split read will be separated while for an insertion (e) they are adjacent with the middle part unmapped. (f) shows a CNV region (green) detected by signature of read depth. The donor contains twice of a CNV region in the reference, therefore, there will be twice the expected number of reads mapped to the reference.

To sum up, each category has its limitations. Software based on different signature differs with respect of sensitivity and specificity [10]. The SV detection will certainly benefit from a combination of SV callers from different categories. In order to determine which callers should be used under which criteria, we started to benchmark eleven pre-selected SV callers.

1.3 Minor thesis

The objectives of this thesis were to 1) evaluate available SV software; 2) build a pipeline to run all the software automatically; 3) better select and merge the results. The research was carried out within plant breeding and bioinformatics groups from Wageningen University . This thesis lasts for four months, from 23rd April till 24th August, 2012.

2. Data sets and Methods

2.1 Data sets

The sequence data sets used for software benchmarking were three *Arabidopsis thaliana* accessions: No-0, Po-0 (the donor) and Col-0 (the reference) from 19 genomes project of *Arabidopsis thaliana* (http://mus.well.ox.ac.uk/19genomes/). They were generated by the NGS Illumina platform [11]. They used two libraries for No-0 and Po-0. Phase 1 corresponded to 36bp paired end reads with ~200bp inserts, while Phase 2 comprised 51bp reads with ~400bp inserts. Reads had been aligned to the Col-0 (TAIR10 annotation) using BWA v0.5.9, and information was stored in BAM files. BAM files could also be downloaded from the project website. Outcomes of the benchmarking would be used in the later research for three *Brassica rapa* accessions: Turnip VT117, Rapid Cycling and Chiifu.

2.2 Methods

2.2.1 SV detection

To detect SV, we used eleven structural variants callers: Breakdancer_max, Breakway, Clever, cnD, CNVnator, CREST, GASVpro, Hydra, Pindel, SVDetect, SplazerS [12-22], which were categorized to three groups based on their detection signatures.

Groups A included Breakdancer_max, Breakway, Clever, GASVpro, Hydra and SVDetect, using paired end distance/orientation as the signature; Groups B included CREST, Pindel and SplazerS, using split reads; Groups C includes cnD, CNVnator and SVDetect, based on the read depth.

All callers were downloaded and installed in advance. They received BAM/SAM files as input, relying on SAMtools to function properly.

2.2.2 Pipeline building

To improve the efficiency for data analysis, we built a pipeline to run all the software. First, scripts were written to make sure each software runs properly. For callers which cannot run properly, errors were sent to the author and a conjoint effort was made to solve the problem. After that, more scripts were added to combine workflows of properly functioning software together. Python language was used to write the scripts. With this pipeline, possible users should be able to generate results from all callers at one time. The value of various parameters were defined in a configuration file before running this pipeline.

2.2.3 Benchmarking SV detection callers

As "reference results", we used lists of sequence variants of No-0 and Po-0 relative to TAIR10, as downloaded from 19 genome project website. Around 40% of indels in the lists were 1bp, and ~40% of indels were between 2-19bp, while the rest 20% were bigger than 19 bp. We regarded them as reliable results, because they were coming from a combination of iterative mapping and *de novo* assembly, which had the most accuracy so far. These published results were only comprising of SNPs and indels (deletion and insertion), therefore, we only focused on benchmarking indels detection ability of different callers.

Raw calls from each structural variant caller were transformed into BED format, and then compared to the "reference results" by BEDTools. If deletions (DEL) and insertions (INS) overlapped with the "reference results", they were regarded as "proved indels".

Uniqueness and size based benchmarking

In normal benchmarking, the caller which calls more SV comparing to the standard would be regarded as a better caller. Since our aim was to benefit from a combination of SV callers from different categories, we proposed here a novel way of benchmarking, based on their uniqueness and SV sizes. We firstly check the performance of the caller to detect unique proved indels, in other words, the ability to detect indels that was private to itself, not sharing by other callers. To check whether the caller performance is size dependent, we divided detected indels into 5 intervals: 1-19bp, 20-49bp, 50-99bp, 100-999bp, 1000-9999bp, >9999bp. The ability of different callers to detect indels of different sizes was studied. With this sized based approach, we wanted to provide better choice of SV callers to detect specific sizes of indels .

Minimum supported reads

In paired end distance/orientation based algorism, each call is detected by a certain number of reads aligning to reference genome. The smallest number of reads the caller requires to detect a certain call, we referred it as, minimum supported reads. To see whether this value would reflect detection sensitivity, we randomly selected several shared SV from different callers and compared this value with each other.

Multiple anchored reads

For short reads, there is high possibility that they will align to multiple regions on the reference genome, especially for repetitive regions. These reads are called multiple anchored reads. Detection of SV on the repetitive regions has been a barrier for all computational methods for a long time. In our research, we tried to summarize the ratio of multiple anchored reads for different algorisms and discuss which callers might be better performing for genomes containing more repetitive regions.

Read depth coverage validation

It is possible that within the same region, the types of SV different callers claimed are contradictory. With read depth based callers, we zoomed in specific region to further validate the result in a visible way. An example was shown in the results part.

2.2.4 Results merging

After benchmarking, we merged results (calls) from all raw callers into a non-redundant result based on SV type and chromosomal coordinates. The rules applied for merging results are outlined as follows:

(1) Calls that were private to each caller were accepted;

(2) If the coordinate of calls from different callers overlapped and the types of SV those callers claimed were consistent, we merged the calls, taking the outer coordinates of the union of the spans;

If the coordinate of calls from different callers overlapped, but the types of SV those callers claimed were contradictory, first checked the location and size of this SV, and then further analysed these calls by validating depth of coverage in that region.

The final call set consisted of four types of SV: insertions, deletions, inversions and imbalance substitutions.

3. Results

3.1 Software, analysis method and called SV types

We have tested a variety of SV detection callers, in total, eleven software. In which, GASVPro and SplazerS did not run properly. GASVPro gave disordered format of intermediate files and successive procedure was interrupted. According to our communication with the authors of GASVPro, errors might be caused by the incompatibility of inconsistent versions between different components. New version of GASVPro has released several days before, but not enough time was left for testing before deadline. SplazerS could not recognize our SAM files. Changing of parameter did not solve this problem and no response was from the author so far. Hydra and Crest run properly, but no results were output. Reasons for this still remained to be discussed. In a word, we got outputs from 7 callers. Table 1 lists the software, analysis signature and SV types called by these algorithms. As expected, 5 algorithms mainly based on paired end distance/orientation or split reads could call exact SV, like DEL and INS with chromosomal coordinates, while the other 2 algorithms using read depth, could only provide information on approximate copy number gain or lost in some regions. It was reasonable that we started benchmarking on callers based on the first two signature and then further validated with read depth based callers.

Table 1 Analysis signatures and called SV types of 11 callers

Software	Signature	SV types called	N/A
Breakdancer_max	Paired end distance/orientation	DEL, INS, INV, Translocation	-
Clever	Paired end distance/orientation	DEL, INS	
Pindel	Split reads	DEL, INS, INV, Duplication	-
SVDetect	Paired end distance/orientation & Read depth	DEL, INS, INV, Translocation, Duplication, Copy number gain/lost	-
Breakway	Paired end distance/orientation	DEL, INS, Translocation	-
CNVnator	Read depth	Copy number gain/lost	-
cnD	Read depth	Copy number gain/lost	-
GASVpro	Paired end distance/orientation	-	Disordered format of intermediate files
Hydra	Paired end distance/orientation	-	No outputs in final step
Crest	Split reads	-	No outputs in final step
SplazerS	Split reads	-	Input unrecognized

3.2 Indels detection

As referred in Data sets and Methods, we used lists of sequence variants of No-0 and Po-0 relative to TAIR10 as "reference results". DEL and INS were the two SV types we used for benchmarking. Pindel had the most calls as expected because it was the only caller having 1 base resolution. Within 5 algorithms based on mapping, Breakway did not show overlaps with reference results for both of DEL and INS, it might be due to an inappropriate set up of its parameters. SVDetect gave no proved calls for INS. Here we took results from No-0-1 as an illustration. Table 2 shows the number of proved DEL detected by 4 callers, and INS detected by 3 callers for No-0-1. For DEL detection, each caller had shared SV calls with other 3 callers and unique calls private to itself (Figure 2). For INS detection, Breakdancer_max and Clever had common calls, while Pindel only comprised unique calls (Figure 3).

Table 2 the number of proved deletions detected by 4 callers and insertions detected by 3 callers on No-0-1

Software	DEL	INS
Breakdancer_max	1537	29
Clever	3783	1694
Pindel	21809	11144
SVDetect	742	-

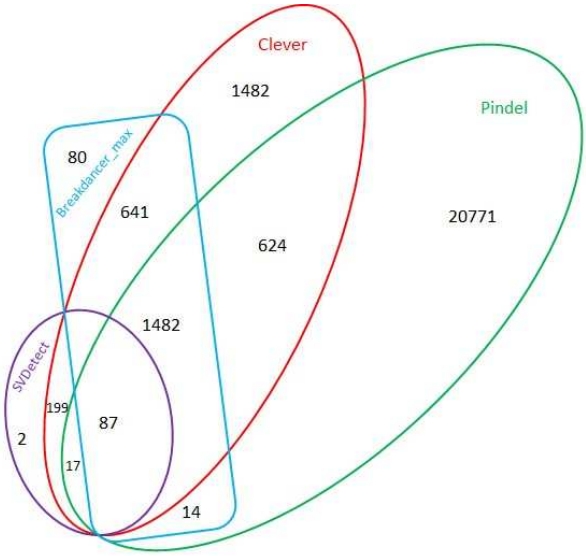


Figure 2 Unweighted Venn diagrams of 4 DEL call sets from 4 callers on No-0-1. Green for Pindel, red for Clever, blue for Breakdancer_max and purple for SVDetect. Each call set comprises of shared DEL and unique DEL. Numbers in overlapping regions indicate shared DELs.

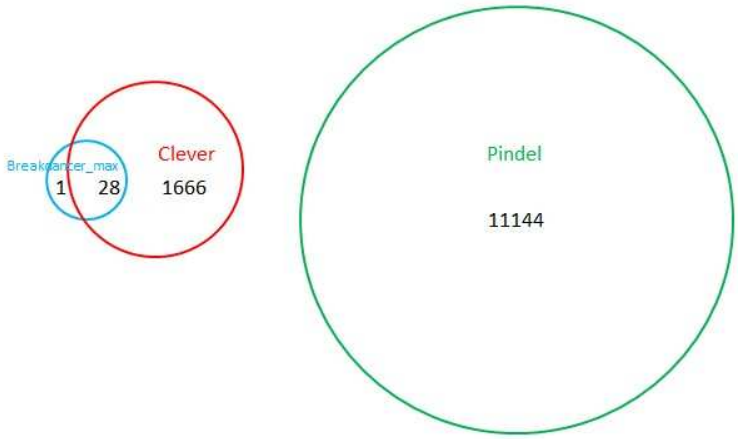


Figure 3 Unweighted Venn diagrams of 3 INS call set from Breakdancer_max, Clever and Pindel on No-0-1. Breakdancer_max and Clever shared 28 INS. All proved INS detected by Pindel were unique.

3.3 Uniqueness and size based benchmarking

We calculated proved unique indels detected by 4 callers for No-0 and Po-0. To check whether their performance was size dependent, we divided detected indels into 6 intervals: 1-19bp, 20-49bp, 50-99bp, 100-999bp, 1000-9999bp, >9999bp.

3.3.1 Unique DEL detection

Table 3 shows the proved DEL calls for two Phases of No-0 and Po-0 within each size intervals. Average number of proved DEL calls for each caller in each interval had been calculated and plotted to a line chart. From Figure 4, we could see caller performance greatly depended on the size of DELs.

1-19 bp: Pindel and Clever were the only callers that made unique predictions, where the split reads based algorism of Pindel yielded the most favourable results.

20-49 bp: Within this size range, Clever started performing better than Pindel. Breakdancer_max could also achieved some unique calls, but the number was much less than the other two.

50-99 bp: Clever clearly outperformed Breakdancer_max and Pindel. The ability of Pindel to detect unique DEL here had been decreased quickly.

100-999 bp: Clever delivered the largest amount of unique DEL, followed by Breakdancer_max, while Pindel and SVDetect predicted much less unique DEL.

1000-9999 bp: Here, Clever still achieved best performance than other callers.

>9999 bp: Rare calls could be detected only by Clever and SVDetect.

Table 3 The number of unique proved DEL calls for two phases of No-0 and Po-0 in 6 size intervals

Caller	Accession	Length of unique DEL (bp)						Total
		1-19	20-49	50-99	100-999	1000-9999	>9999	
Breakdancer_max	No-0-1	-	14	54	12	-	-	80
	No-0-2	-	-	31	65	-	-	96
	Po-0-1	-	65	57	15	-	-	137
	Po-0-2	-	-	27	124	1	-	152
	Avg	-	47	42	54	1	-	
Clever	No-0-1	162	876	148	251	40	5	1482
	No-0-2	19	386	130	109	19	1	664
	Po-0-1	556	672	80	127	21	-	1459
	Po-0-2	71	345	103	41	10	-	572
	Avg	202	570	115	132	23	3	
Pindel	No-0-1	20552	232	16	1	-	-	20771
	No-0-2	30032	735	56	16	2	-	30841
	Po-0-1	24823	196	7	2	-	-	25028
	Po-0-2	32064	726	46	4	2	-	32842
	Avg	26868	472	31	6	2	-	
SVDetect	No-0-1	-	-	-	2	-	-	2
	No-0-2	-	-	-	1	5	1	7
	Po-0-1	-	-	-	2	3	-	5
	Po-0-2	-	-	-	-	4	-	4
	Avg	-	-	-	2	4	1	

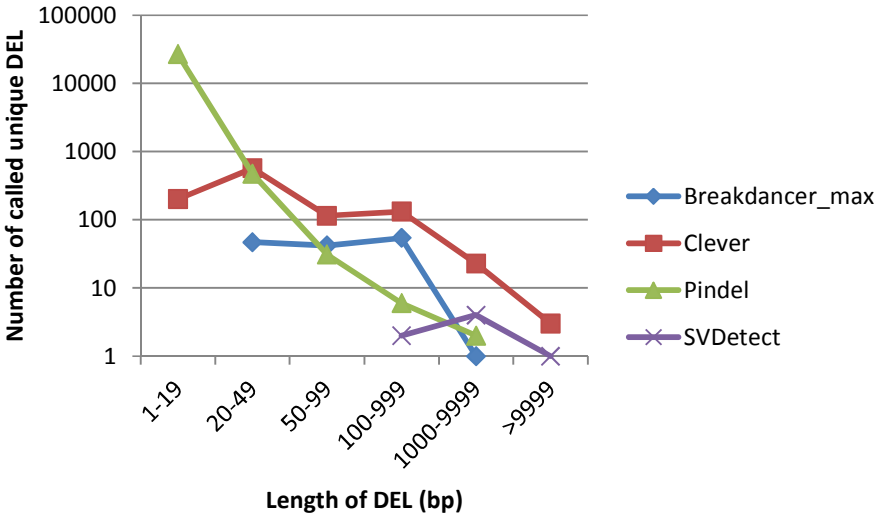


Figure 4 Performances of 4 callers to detect different sizes of unique DEL

3.3.2 Unique INS detection

Table 4 shows the proved INS calls from 3 callers for two Phases of No-0 and Po-0 within each size intervals. For unique INS detection, caller performance also greatly depended on the size (Figure 5).

1-19 bp: Similarly as in DEL detection, Pindel and Clever made unique predictions in this range, and Pindel performed much better.

20-49 bp: Clever had the best performance and it was the only caller to detect unique INS.

50-99 bp: Both of Clever and Breakdancer_max could delivered unique INS calls, but Clever performed better than Breakdancer_max.

100-999 bp: As in 50-99 bp, Clever provided more predictions than Breakdancer_max, although the performance of Breakdancer_max increased slightly.

>999 bp: None of the callers could predict unique INS that were larger than 999bp.

Table 4 The number of proved unique INS calls detected by 3 callers for two phases of No-0 and Po-0 within 5 size intervals.

Caller	Accession	Length of unique INS (bp)					Total
		1-19	20-49	50-99	100-999	>999	
Breakdancer_max	No-0-1	-	-	1	-	-	1
	No-0-2	-	-	-	10	-	10
	Po-0-1	-	-	-	-	-	-
	Po-0-2	-	-	-	-	-	-
	Avg	-	-	1	10	-	
Clever	No-0-1	128	1293	219	26	-	1666
	No-0-2	37	1124	277	100	-	1538
	Po-0-1	1136	1380	175	4	-	2695
	Po-0-2	85	1002	193	124	-	1404
	Avg	347	1200	216	64	-	
Pindel	No-0-1	11144	-	-	-	-	11144
	No-0-2	17508	-	-	-	-	17508
	Po-0-1	13785	-	-	-	-	13785
	Po-0-2	18245	-	-	-	-	18245
	Avg	15171	-	-	-	-	

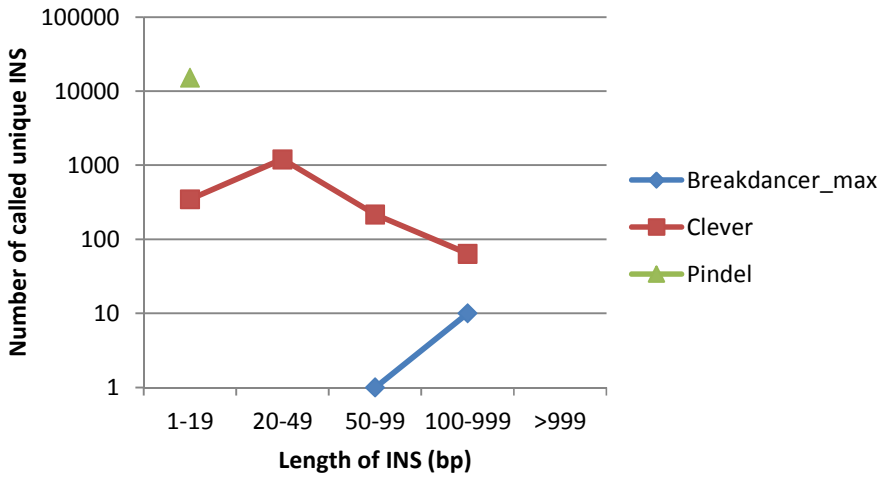


Figure 5 Performances of 3 callers to detect different sizes of unique INS

3.4 Comparison of minimum supported reads

For the comparison of minimum supported reads, we randomly selected 6 shared DEL detected by 4 callers (Table 5). Results showed that Pindel required least reads to detect the same DEL, followed by Clever and Breakdancer_max, while SVDetect needed to aligned more reads to the same region to detect this DEL. It was possible that this value could reflected caller detection sensitivity. More contents about this would be discussed in the discussion part.

Table 5 Minimum supported reads of 6 shared DELs detected by 4 callers.

Chromosome	Coordinate1	Coordinate2	Pindel	Clever	Breakdancer_max	SVDetect
Chr1	3342445	3343052	3	4	4	5
Chr1	3728971	3730422	5	8	10	13
Chr1	3935854	3936316	4	6	9	16
Chr1	6722873	6723422	4	15	19	24
Chr1	7714701	7715145	3	13	21	23
Chr1	9475997	9480635	5	15	17	24

3.5 Comparison of the ratio of multiple anchored reads

According to the manual, Clever already considered multiple anchored reads in its algorism, although it did not output intermediate files with used reads. After calculation, we found Breakdancer_max did not consider any multiple anchored reads in its algorism. For Pindel, around 6-7% of reads it used for calling SV were multiple anchored reads, while SVDetect had 11.6-13.1% reads that could map to multiple regions on genomes (Table 6).

Table 6 Ratios of multiple anchored reads used by 4 callers in their algorisms.

Caller	Ratio of multiple anchored reads (%)			
	No-0-1	No-0-2	Po-0-1	Po-0-2
Breakdancer_max	0	0	0	0
Pindel	7.0	7.0	6.2	6.0
SVDetect	14.4	11.6	13.3	13.1

3.6 Read depth coverage validation

For unproved contradictory results claimed by different callers in the same region, we validated them with read depth coverage based callers. An example was shown as follows. For an indel on Chr5 of No-0, Clever predicted it as an 106bp DEL, while Breakdancer_max claimed it as an 131bp INS. Using CNVnator, the result of Clever was more likely to be true in this case. Between chromosomal coordinates 10936768 to 10896884 (blue line, Figure6), there was an obvious

copy number lost shown by red lines within yellow oval. It was also possible to validate results in this way using cnD and SVDetect, but we should be aware that they could only provide numbers for relative copy number change without graphs.

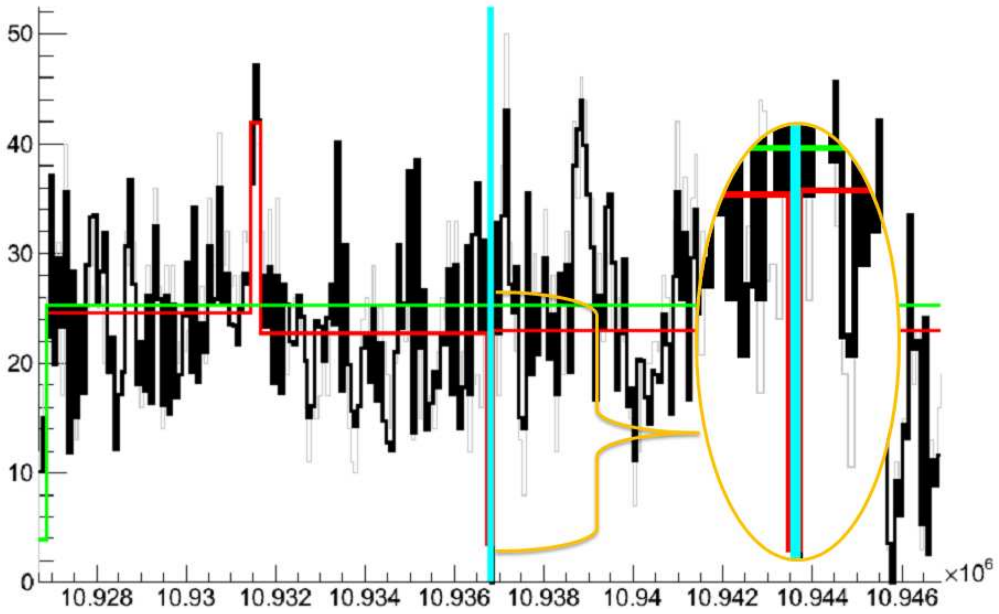


Figure 6 A DEL validated by CNVnator on chromosome 5 of No-0. The horizontal axis represents the chromosome coordinates, while vertical axis illustrates the level of read depth coverage. Green line shows the read depth of the reference. Black line shows the read depth coverage of each bin in the donor, and red line shows the read depth in the donor. The difference of red line compared to green line suggests copy number gain/lost. The blue line in this picture represents where a big copy number lost happened, corresponding to the DEL claimed by Clever.

4. Discussion

According to the results above, types and sizes of SV detected by individual SV callers were different. To detect exact SV, especially indels, users should choose callers based on pair end distance/orientation or split reads. Although 4 callers (Breakdancer_max, Clever, Pindel, SVDetect) had shared DELs with each other, each caller was able to predict unique DELs that were private to itself, using their own algorisms. The presence of INS were more difficult to detect. 3 callers (Breakdancer_max, Clever, Pindel) gave much lower numbers of results than DEL detection. Unique indels detection by callers were size dependent, which means users should take SV size into account to choose a suitable combination of SV callers instead of consisting to one favourite caller for indels of all size.

We recommend to use Pindel to detect very small indels (1-19bp), using split reads signature, which guarantee its accuracy with one based resolution. However, users should notice that its performance greatly decreased when the indel size is increasing. It is sensible to consider other software for longer indels. Breakdancer_max better performed in detecting longer size indels (50 to 999 bp). SVDetect could also add some value to detect long DELs.

Clever in our benchmarking has seem to be an “all round” caller on average. It was possible to detect indels in all sizes. It complemented predictions for split reads caller Pindel for very small indels, and other pair end mapping callers (Breakdancer_max and SVDetect) for longer indels. Most of all, it clearly outperformed other callers for mediate size indels (20-99 bp). The strength of Clever to detect indels of this size has also been emphasised in its paper [14].

Using 6 shared DEL calls, a similar pattern on minimum supported reads was found between Pindel, Clever, Breakdancer_max and SVDetect. Pindel always required least reads to detect the same DEL, followed by Clever and Breakdancer_max, while SVDetect needed most. One of the explanation for this might be that, this value reflects caller detection sensitivity. The less reads a caller requires to detect an SV, the lower coverage of libraries that caller needs as a start input. This idea by far was only a hypothesis and it was tested above with random sampling. More statistically analysis might help to confirm this.

Detection of SV in repetitive regions has remained an challenge for all computational methods. Callers that did not consider any multiple anchored reads in its algorism, like Breakdancer_max, may have problems to detect SV in repetitive regions. By contrast, callers using multiple

anchored reads have already taken repetitive regions in to account. In our case, SVDetect used around 6-7% more multiple anchored reads than Pindel, but it did not show great advantage of detecting more indels. Possible reason might be that, more multiple anchored reads involved also increased false positives.

Read depth based callers (CnD and CNVnator) could only provide approximate copy number gain/lost without determining breakpoints, therefore, they might be more suitable for downstream SV validation.

Despite limitations of single callers, we were able to produce a more comprehensive set of SV calls by incorporating more algorisms from a wider variety of SV callers. Based on the idea of a complementary method, we are ready for creating an SV detection optimizer with parameters related to SV types, sizes and genome characteristics.

5. Conclusion

Benchmarking results suggested that the types and sizes of structural variants detected by individual caller were different. Performances of unique indels detection by Breakdancer_max, Clever, Pindel and SVDetect were size dependent. Pindel was the most favourable caller to detect very small indels (1-19bp), but its performance greatly decreased when the indel size increased. Clever was outstanding in predicting intermediate size indels (20-99 bp). Breakdancer_max better performed in detecting longer size indels (50 to 999 bp), while SVDetect was able to predict some long deletions. CnD and CNVnator could be used for downstream validation. To produce a more comprehensive set of calls, it is better to use a complementary method involving a variety of structural variation detection callers with different algorisms.

6. Acknowledgements

I would like to express my gratitude to Guusje Bonnema and Ke Lin for trusting me with this project. I am grateful for Guusje Bonnema for always directing me to the right way. Ke Lin has been a tremendous support and help for me during the intensive work. Many thanks to other members in plant breeding and bioinformatics groups. This project was not just a technical task for me, but also an enjoyable experience before graduation of my MSc in the Netherlands.

7. References

1. Paterson, A. H., Lan, T., Amasino, R., Osborn, T. C., & Quiros, C. (2001). *Brassica* genomics: a complement to, and early beneficiary of the *Arabidopsis* sequence. *Genome Biology*, 2, 1339-1347.
2. Lou, P., Zhao, J., Kim, J. S., Shen, S., Del Carpio, D. P., Song, X., Koornneef, M. (2007). Quantitative trait loci for flowering time and morphological traits in multiple populations of *Brassica rapa*. *Journal of experimental botany*, 58, 4005-4016.
3. Lou, P., Xie, Q., Xu, X., Edwards, C., Brock, M., Weinig, C., & McClung, C. (2011). Genetic architecture of the circadian clock and flowering time in *Brassica rapa*. *Theoretical and Applied Genetics*, 123, 397-409.
4. Pino Del Carpio, D., Basnet, R. K., De Vos, R. C. H., Maliepaard, C., Visser, R., & Bonnema, G. (2011). The patterns of population differentiation in a *Brassica rapa* core collection. *Theoretical and Applied Genetics*, 122, 1105-1118.
5. Medvedev, P., Stanciu, M., Brudno, M. (2009). Computational methods for discovering structural variation with next-generation sequencing. *Nature Methods*, 6, S13-20.
6. Dalca, A. V., & Brudno, M. (2010). Genome variation discovery with high-throughput sequencing data. *Briefings in bioinformatics*, 11, 3-14.
7. Li, R., Zhu, H., Ruan, J., Qian, W., Fang, X., Shi, Z. & Kristiansen, K. (2010). *De novo* assembly of human genomes with massively parallel short read sequencing. Genome research, 20, 265-272.
8. Suzuki, S., Yasuda, T., Shiraishi, Y., Miyano, S., & Nagasaki, M. (2011). ClipCrop: a tool for detecting structural variations with single-base resolution using soft-clipping information. *BMC Bioinformatics,* 12, S7.
9. Zhang, J., Wang, J., & Wu, Y. (2012). An improved approach for accurate and efficient calling of structural variations with low-coverage sequence data. *BMC Bioinformatics*, 13, S6.
10. Wong, K., Keane, T. M., Stalker, J. & Adams D. J. (2010). Enhanced structural variant and breakpoint detection using SVMerge by integration of multiple detection methods and local assembly. *Genome Biology*, 11, R128.
11. Gan, X., Stegle, O., Behr, J., Steffen, J. G., Drewe, P., Hildebrand, K. L., Sreedharan, V. T. (2011). Multiple reference genomes and transcriptomes for *Arabidopsis thaliana*. *Nature*, 477, 419-423.

12. Chen, K., Wallis, J. W., McLellan, M. D., Larson, D. E., Kalicki, J. M., Pohl, C. S., Locke, D. P. (2009). BreakDancer: an algorithm for high-resolution mapping of genomic structural variation. *Nature Methods*, 6, 677-681.
13. Clark, M. J., Homer, N., O'Connor, B. D., Chen, Z., Eskin, A., Lee, H., Nelson, S. F. (2010). U87MG decoded: the genomic sequence of a cytogenetically aberrant human cancer cell line. *PLoS genetics*, 6, e1000832.
14. Marschall, T., Costa, I, Canzar, S., Bauer, M., Klau, G., Schliep, A., & Schonhuth, A. (2012). CLEVER: Clique-Enumerating Variant Finder. Quantitative biology, under revision.
15. Emde, A. K., Schulz, M. H., Weese, D., Sun, R., Vingron, M., Kalscheuer, V. M., Reinert, K. (2012). Detecting genomic indel variants with exact breakpoints in single-and paired-end sequencing data using SplazerS. *Bioinformatic*s, 28, 619-627.
16. Abyzov, A., Urban, A. E., Snyder, M., & Gerstein, M. (2011). CNVnator: An approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome research*, 21, 974-984.
17. Wang, J., Mullighan, C. G., Easton, J., Roberts, S., Heatley, S. L., Ma, J., Ding, L. (2011). CREST maps somatic structural variation in cancer genomes with base-pair resolution. *Nature Methods*, 8, 652-654.
18. Sindi, S. S., Onal, S., Peng, L., Wu, H. T., & Raphael, B. J. (2012). An integrative probabilistic model for identification of structural variation in sequencing data. *Genome Biology*, 13, R22.
19. Quinlan, A. R., Clark, R. A., Sokolova, S., Leibowitz, M. L., Zhang, Y., Hurles, M. E., Hall, I. M. (2010). Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome. *Genome research*, 20, 623-635.
20. Zeitouni, B., Boeva, V., Janoueix-Lerosey, I., Loeillet, S., Legoix-Né, P., Nicolas, A., Barillot, E. (2010). SVDetect: a tool to identify genomic structural variations from paired-end and mate-pair sequencing data. *Bioinformatics*, 26, 1895-1896.
21. Ye, K., Schulz, M. H., Long, Q., Apweiler, R., & Ning, Z. (2009). Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics*, 25, 2865-2871.
22. Zeitouni, B., Boeva, V., Janoueix-Lerosey, I., Loeillet, S., Legoix-Né, P., Nicolas, A., Barillot, E. (2010). SVDetect: a tool to identify genomic structural variations from paired-end and mate-pair sequencing data. *Bioinformatics*, 26, 1895-1896.