SOCRATES: IDENTIFICATION OF GENOMIC REARRANGEMENTS IN TUMOUR GENOMES BY RE-ALIGNING SOFT CLIPPED READS—SUPPLEMENTARY MATERIAL

JAN SCHROEDER*, ARTHUR HSU*, SAMANTHA BOYLE, GEOFF MACINTYRE, MAREK CMERO, RICHARD W. TOTHILL, RICKY W. JOHNSTONE, MARK SHACKLETON, ANTHONY T. PAPENFUSS†

1. Algorithm Implementation Details

The main Socrates algorithm is implemented in java, with a driver and several utility scripts implemented in python. For convenience, Socrates can optionally: annotate whether predicted breakpoints overlap known repeats, output a BAM file with anchor sequences and re-aligned long soft clips that are mated, which is useful for visualising the split read data in IGV.

Socrates is designed to run efficiently on modern day computing resources. The algorithm offers parallelisation scaling to any number of processors on a shared memory machine, and memory mapping input files allow for efficient usage of memory and improved speed We took particular care when implementing Socrates to streamline computation by balancing the use of memory and multi-core CPUs. Firstly, anchor site information is encoded into re-alignment records in SAM format as paired read information. Anchor chromosome, position and orientation (positive if aligned sequence is to the left of breakpoint) are encoded as the paired read entry, with "mate unmapped" flag used as ideal evidence flag (evidence is ideal if soft clip is at 5' end of properly mapped pair or 3' end of anomalously mapped pair) and an extra attribute tag "ZS" containing aligned anchor sequence. Socrates stratifies chromosomes in parallel and then merges all chromosomes when all stratification is done. Clusters are stored in a Red-Black tree structure to allow fast retrieval of clusters for merging and pairing. Socrates implemented multi-threading to perform concurrent building and pairing of clusters for all possible pair of loci. To enable this implementation with minimum disk IO, BAM files are all memory-mapped for fast and parallel access.

2. Algorithmic Complexity

The initial preprocessing step of the algorithm parses the entire input file and scans it for soft clipped reads of any kind – this is dependent on the number of reads in the input, say N: $O(N)$. Note that we do not consider the read length as a separate variable here, but include it in the size of the input N. All the following stages work on the long soft clips, short soft clips, the generated clusters, or any combination of these. All these variables are directly dependent on the initial input file, so we will continue to refer to

them as N. The remapping of long soft clips is not part of the algorithm, but an external tool – its complexity is typically bound by $O(NG)$, G the genome size. The clustering stage generates a cluster for every re-aligned read in the long soft clip file and then tries to merge clusters if they support the same event. The clusters are kept in a sorted data structure, and are therefore accessible in logarithmic time. The complexity is $O(N \log N)$. The cluster pairing stage holds a complexity of $O(N^2)$. First clusters are compared and paired ($O(N \log N)$), then short read support ($O(N)$) is gathered for each cluster pair, and added to the output. Similarly the short soft clip cluster pairing compares unpaired clusters with reads in the vicinity of its re-aligned locus. The complexity is therefore $O(N^2)$ as well. However, the constants on N are small in these later stages (number of clusters and particularly cluster pairs have been reduced to a small amount compared the number of reads in the input data) that the quadratic relationship is not damaging to the algorithm's performance, and the cluster generation stage is the most time consuming part of the algorithm.

3. Cluster Pairing

A special case not discussed in detail in the paper are fusions with both micro-homologies and untemplated sequence. This case is analogous to the methods discussed in the paper, but we provide a little more detail here. If either of the breakpoint loci shows homology to the novel insert, reads are placed within the boundaries of the insert as long as it coincides with the original chromosome. This causes differences in the mapping loci of the realigned soft clips and the anchor locus of the reciprocal cluster. It can however be addressed as a combination of the general cases of homologies and untemplated sequence. The novel insert causes the realigned soft clips to be soft clipped again, and the homology for a difference Δ in breakpoint loci. Socrates identifies this and reports it accordingly in the output. Supplementary Figure 7 illustrates this scenario.

4. Post-processing of Tumour-Normal Paired Sequencing Data

Socrates provides a number of tools to support useful post-processing, annotation and filtering of its predictions. These include subtraction of predicted germline rearrangements and annotation of breakpoints in coding or repeat regions.

Mutant Allele Frequency. An interesting and potentially useful filter is the mutant allele frequency (MAF). This statistic is the ratio of reads supporting the breakpoint to reads supporting the reference allele (ie reads that map cleanly across the breakpoint without soft-clipping). To estimate MAF, we parse the BAM file and count reads that support the reference for each cluster pair. The advantage of this measure is that the magnitude of the MAF is not dependent on static coverage cutoffs, although its accuracy does depend on coverage. MAF close to 1 is a strong indicator that an event is real and homozygous. Small values of MAF can be meaningful in context of low cellularity, poly-clonal, population studies, meta-genomics, etc. An interesting idea that deserves further exploration is using a low MAF threshold on predicted breakpoints. SNP array analysis of the melanoma data described in the paper suggests the tumour is largely tetraploid with multiple copy

number changes. Single copy breakpoints would be expected to have a MAF around 0.25. Supplementary Figure 8 compares the composition of repeat types in the unfiltered Socrates predictions for tumour at different minimum MAF. Breakpoints located in satellite repeats would be removed altogether from the output using a MAF>0.125. The relative composition then remains consistent at higher MAF thresholds, but the size of the output data is reduced drastically. It seems reasonable to think that a combination of absolute support and a MAF filter may be effective in improving predictions, but more data is required to demonstrate this.

5. Tools and Parameters

The following tools and parameters are used in the results section of the paper:

- DELLY (v 0.0.9): default parameters.
- BreakDancer (v 1.3): default parameters.
- Pindel (v 0.2.4t): true insert size and std are supplied; default parameters, except minimum evidence = 5.
- SVseq2 (v 2.2): default parameters.
- PRISM (v 1.1.6): default parameters, except minimum evidence = 5.
- CREST (v 0.0.1): default parameters.
- Socrates (v 0.9): default parameters.

6. Simulated Data Results

Here we add further plots to illustrate the behaviour of the tested algorithms on simulated data. As the simulation are repeated over ten runs in *E. coli* and five times in human chromosome 12, we investigate the distributions over these repetitions. In the main manuscript only the mean of these distributions is plotted, so Supplementary Figure 3 shows box and whisker plots for the *E. coli* results. Supplementary Figure 4 shows the same for chr12. The distributions are reasonably tight, so that we are confident that the observed behaviour is reproducible over any number of repetitions of the experiment. Furthermore, we view the false negatives presented in Figure 2B in the manuscript in more detail. The presented figure shows calls in a binned structure for the size category. Supplementary Figures 5-7 show histograms with more fine grained sizes of the pure counts of false negatives for the different categories of structural variations. There are gaps between the different size categories as to the experimental design.

We also present numbers for the outcome of the experiments in Table 2.

6.1. **Redundant calls.** During the experiments we discovered that all of the algorithms tend to make redundant calls in the output files: SV events are being reported more than once, either identically or marginally different. Such redundancy can be cause by sequencing errors or low quality sequence affecting the soft-clipping starts. Supplementary Table 1 reports the total number of redundant calls for each method. The calls in Table 2 are adjusted for this redundancy and only show unique false positives.

7. Melanoma breakpoint validation

To validate predicted genomic fusions, 10 somatic rearrangements in the melanoma were selected and PCR primers were designed within the flanking sequence 100bp either side of the breakpoint using Primer3 (see Supplementary Table 2). PCR was run using GoTaq DNA polymerase (Promega, USA) as per manufacturers instructions. Products for large deletions were run on a 2% agarose gel for 90 mins at 100V and PCR products for small deletions were run on a 4% agarose gel for 3 hours at 120V. Products of correct size were cut from the gel and purified using QIAquick gel extraction kit (Qiagen, USA) as per manufacturers instructions. Purified DNA was sequenced at Australian Genome Research Facility (AGRF) to confirm predicted breakpoint.

Supplementary Table 1. Average number of redundant TP calls in simulated runs.

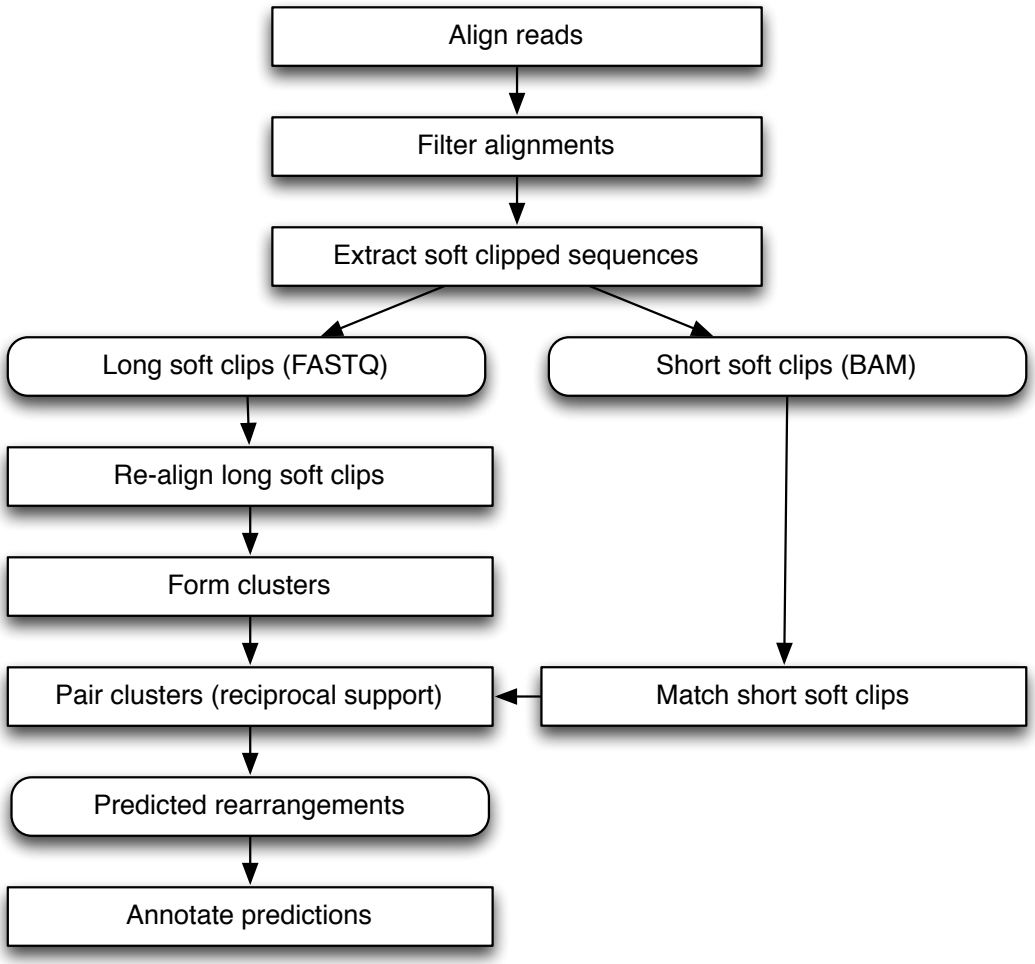
	B	Cl	Cr	D	Pi	Pr	S
E. coli 7.5X	3	43	13	7	0	2	0
E. coli 15X	1	87	20	7	0	4	0
E. coli 30X	1	256	33	7	0	9	0

	Algorithm	Fusion type	Tiny	Small	Medium	Large	Extra large
FPs	BreakDancer	Any	5	53	3	0	2
	Clever	Any	71	1602	10	12	7
	Crest	Any	0	0	1	0	0
	Delly	Any	164	967	34	0	8
	Pindel	Any	5	0	0	1	0
	Socrates	Any	0	1	0	0	6
FNs	BreakDancer	DEL	0	15	2	3	2
		TRA	0	11	6	1	3
		TAN	0	2	0	0	0
		INV	0	12	17	16	22
		INS	49	0	0	0	0
	Clever	DEL	0	1	1	3	16
		TRA	0	36	50	44	46
		TAN	0	29	0	0	17
		INV	0	50	36	38	56
		INS	49	0	0	0	0
	Crest	DEL	0	7	3	8	6
		TRA	0	5	8	5	9
		TAN	0	16	2	1	1
		INV	0	24	6	2	8
		INS	49	0	0	0	0
	Delly	DEL	0	1	6	18	9
		TRA	0	9	11	8	6
		TAN	0	21	0	0	0
		INV	0	0	1	1	0
		INS	49	0	0	0	0
	Pindel	DEL	0	8	0	5	9
		TRA	0	35	42	30	32
		TAN	0	29	22	19	29
		INV	0	21	17	11	20
		INS	3	0	0	0	0
	Prism	DEL	0	1	1	4	1
		TRA	0	35	42	40	38
		TAN	0	29	0	0	0
		INV	0	50	15	14	22
		INS	1	0	0	0	0
	Socrates	DEL	0	3	0	1	3
		TRA	0	0	6	0	2
		TAN	0	3	0	0	0
		INV	0	3	1	1	2
		INS	17	0	0	0	0
All events		DEL	0	29	25	34	31
		TRA	0	36	50	44	46
		TAN	0	29	22	19	29
		INV	0	50	36	38	56
		INS	49	0	0	0	0

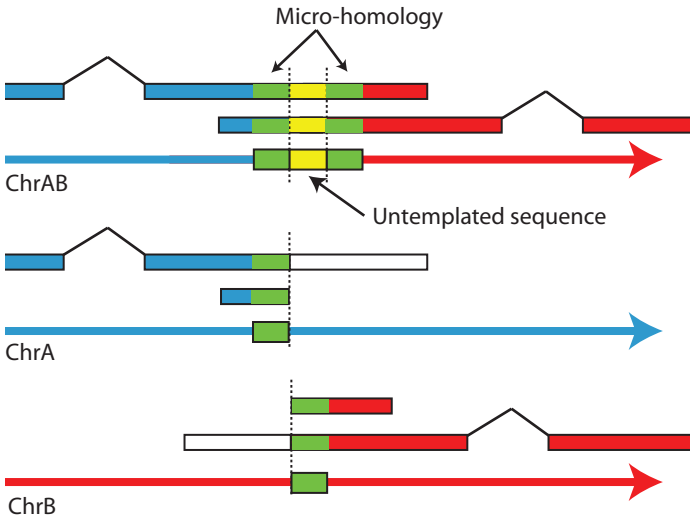
SUPPLEMENTARY TABLE 2. Summary of simulation results on Chr12 with 30X coverage. No SV type is shown for False Positives (FPs), because most algorithms do not distinguish these in a meaningful way. The size categories are extended to cover a continuous scale, since FPs are not bound to the chosen categories. The size categories are: tiny (1-50nt), small (51-250nt), medium (251-1,500nt), large (1,501-15,000nt) and extra large (15,001-140,000,000nt). False Negatives (FNs) are shown for all sizes and event types. The absolute numbers of events present in the data are shown, so statistics such as recall or precision can be computed on specific algorithms/feature types.

	Forward	Reverse
1	TGTAATGTCCATCTCTGGCTTA	TTCTTCCCTTTTTGCGTGAC
2	AGTGGCCGGGAGGACTT	TGCTTGACAATTTATTGCGTCT
5	GAGGCTATGATGAGGGCAAA	TGGTTACAGTGCTTTGCTGAA
6	CCCCTCCAAAGGTTGGTA	GCACCAGAATTTTGGGGATA
9	TGGTAAAAGGCTGGGAGAAA	GTCCTGCAAAGAACATGACC
11	ACAGGGCCTTGAGCAAGATA	ATTGTGATTGGTGGTTGAACA
12	GCCAGGAGGACCAAGGTTTA	CCAGCCCCATCACACAATA
13	AGTGCCAGGAAAAAGAAGCA	ATGGCACAGGGCTCATTAC
14	AGTGTTCAACCACCAATCACA	CATTTTGAGAGGATGAGTATC
18	TCTGTGTGAAACCCAGGACA	GATGAAAAACGGGGAGGAAT

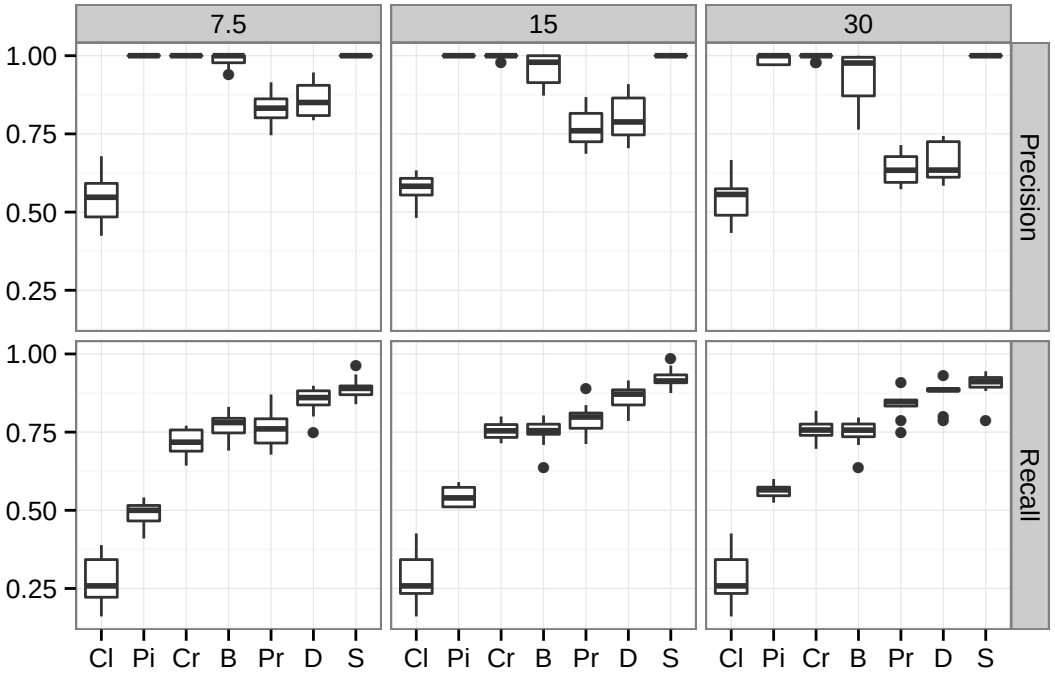
Supplementary Table 3. Table of primers used to amplify fusions in melanoma.



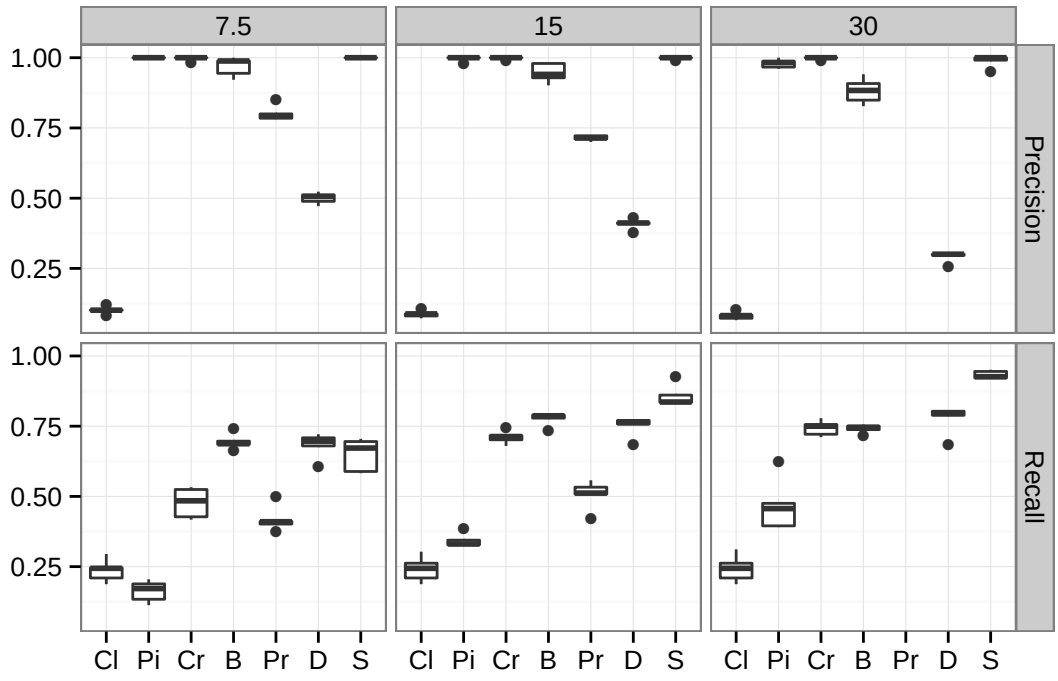
SUPPLEMENTARY FIGURE 1. The Socrates workflow



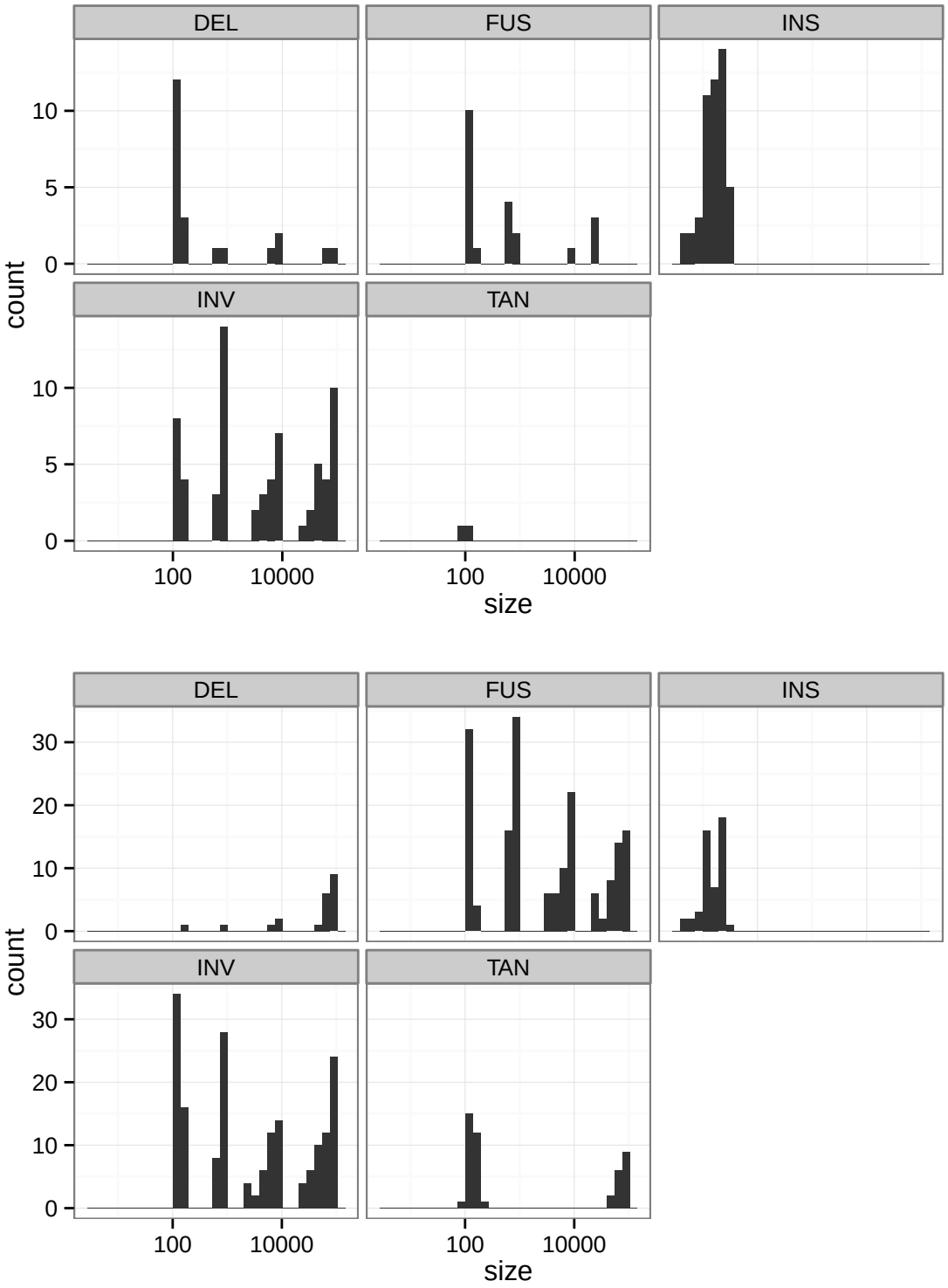
SUPPLEMENTARY FIGURE 2. Cluster pairing in the presence of a nontemplated insert and micro-homologies of the two chromosomes with this inserted sequence.



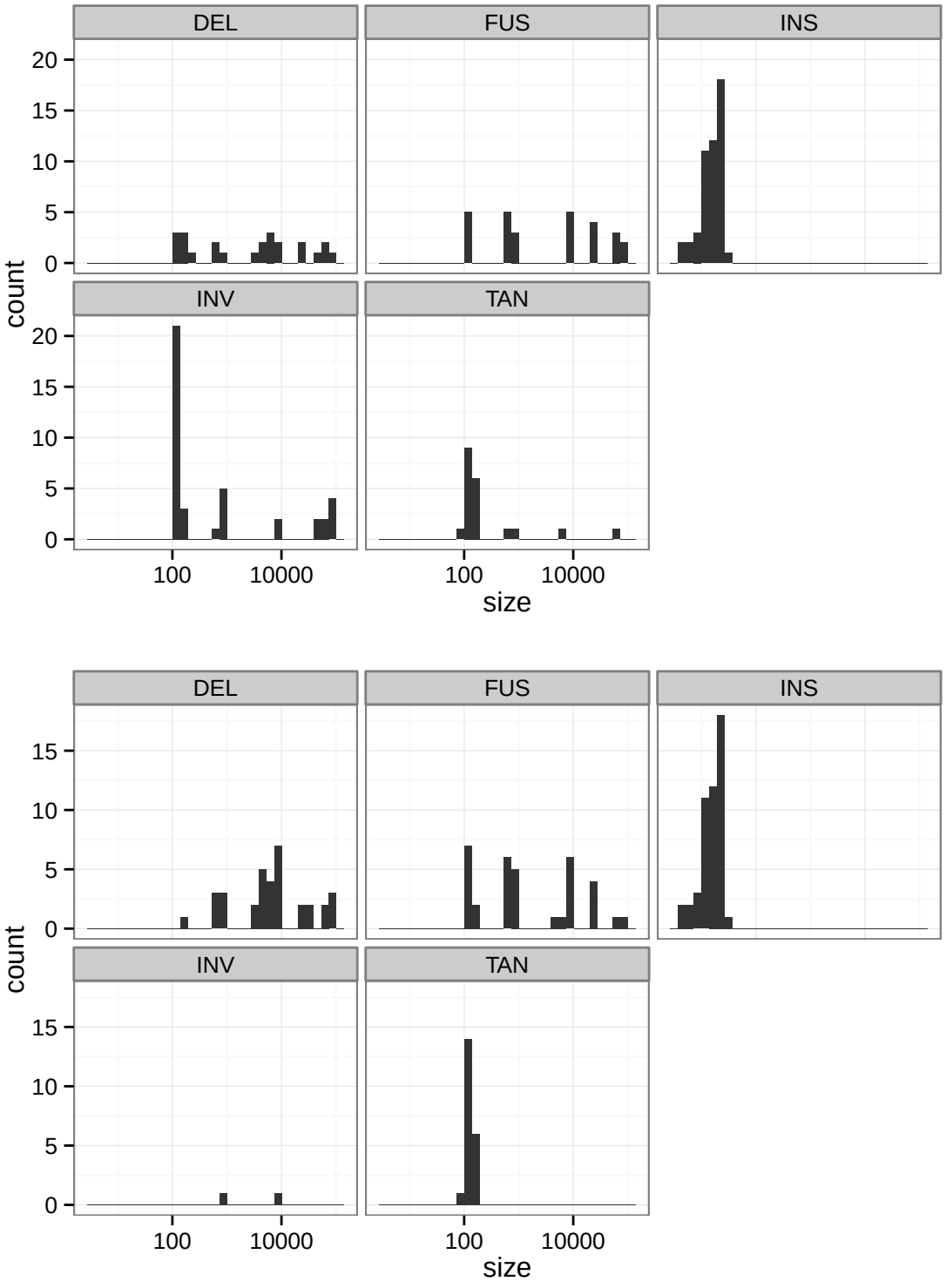
SUPPLEMENTARY FIGURE 3. Distribution of precision and recall values across the 10 repetitions of simulated data in *E. coli*. The horizontal panels show the different coverage levels, and the x-axis distinguishes the tested algorithms: Cl=CLEVER, Pi=Pindel, Cr=CREST, B=BreakDancer, Pr=PRISM, D=DELLY, S=Socrates.



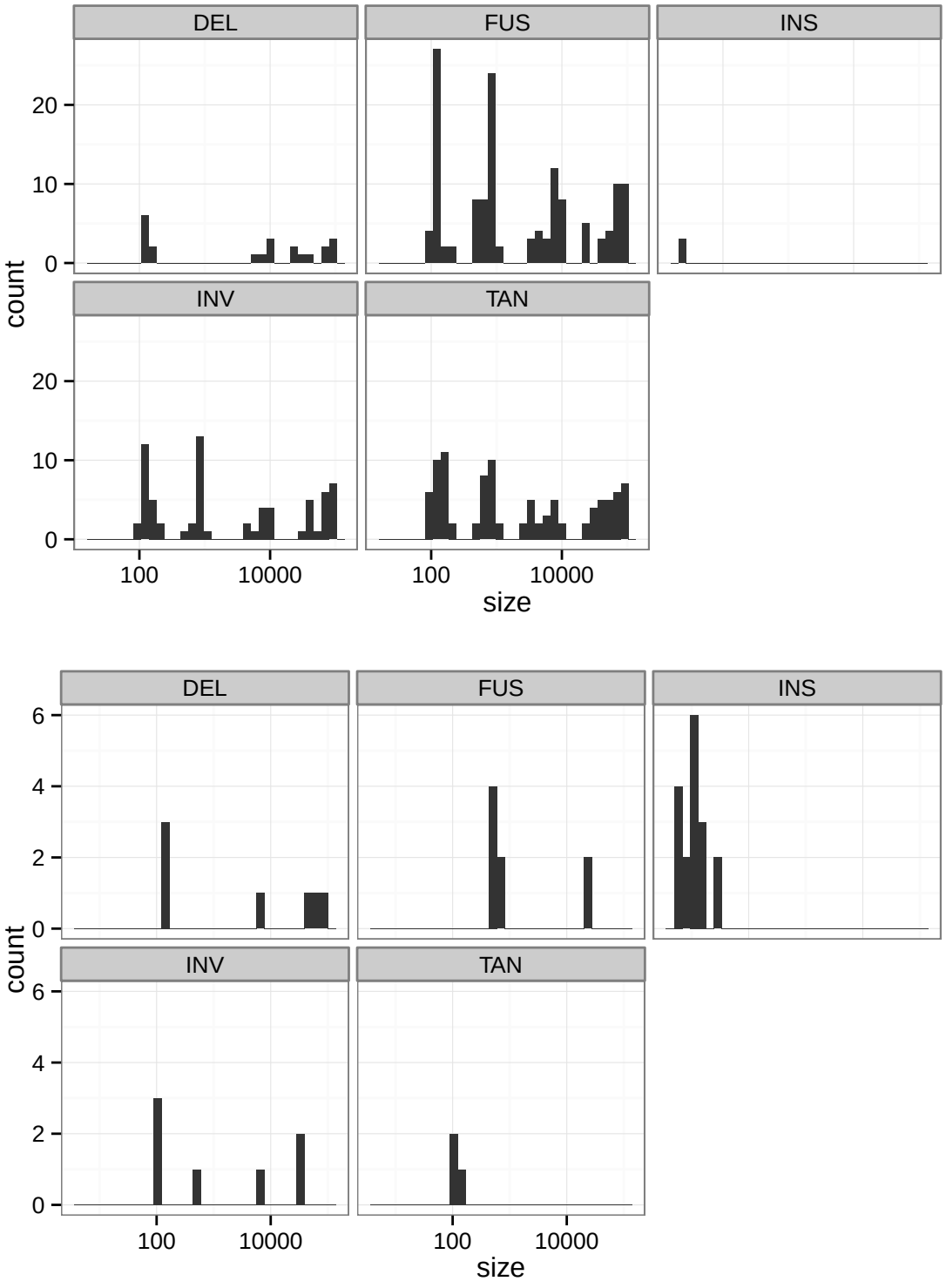
Supplementary Figure 4. Distribution of precision and recall values across the 5 repetitions of simulated data in chr12. The horizontal panels show the different coverage levels, and the x-axis distinguishes the tested algorithms: Cl=CLEVER, Pi=Pindel, Cr=CREST, B=BreakDancer, Pr=PRISM, D=DELLY, S=Socrates.



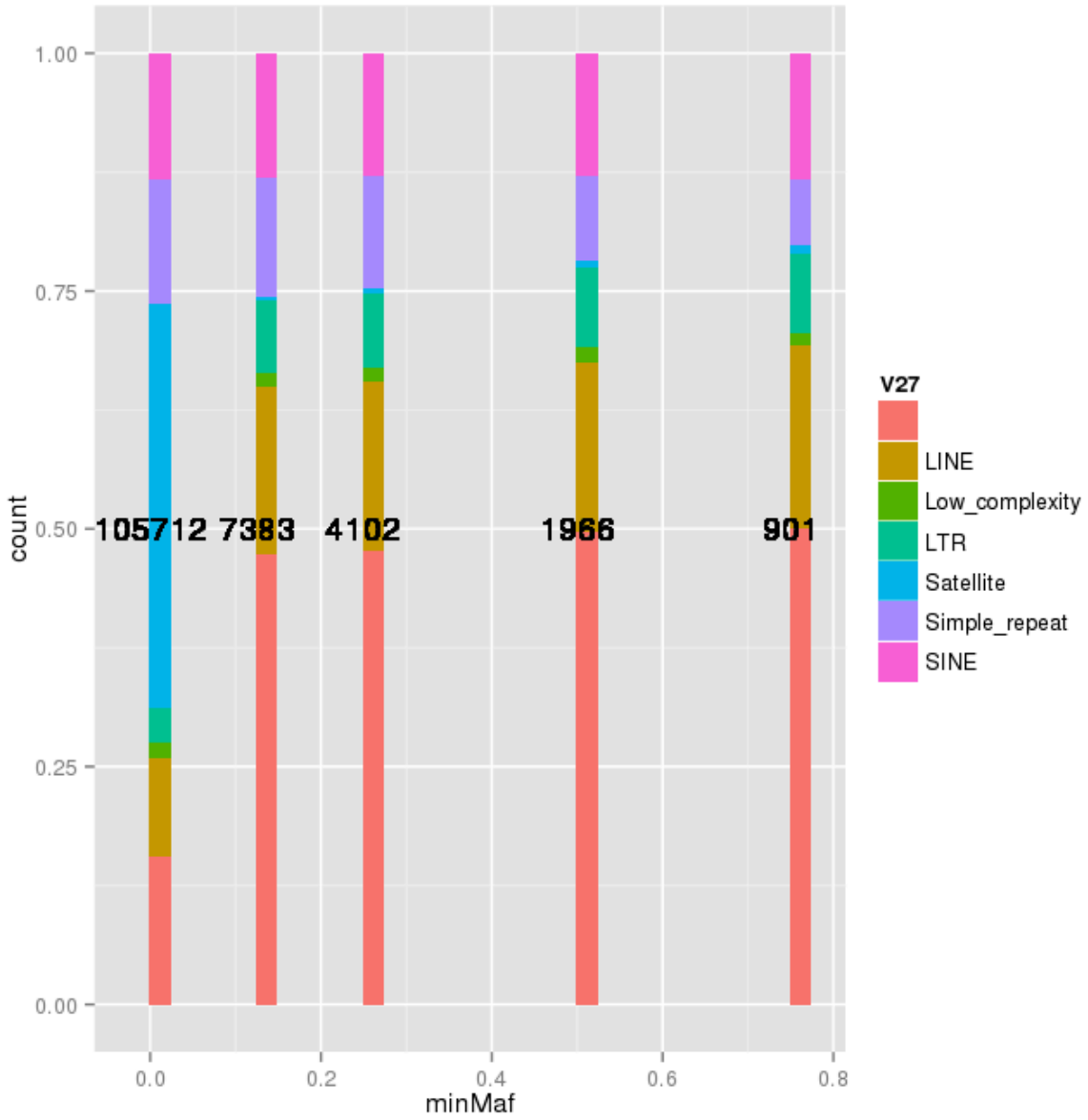
Supplementary Figure 5. Distribution of false negatives with feature size and type for BreakDancer (top) and CLEVER (bottom)



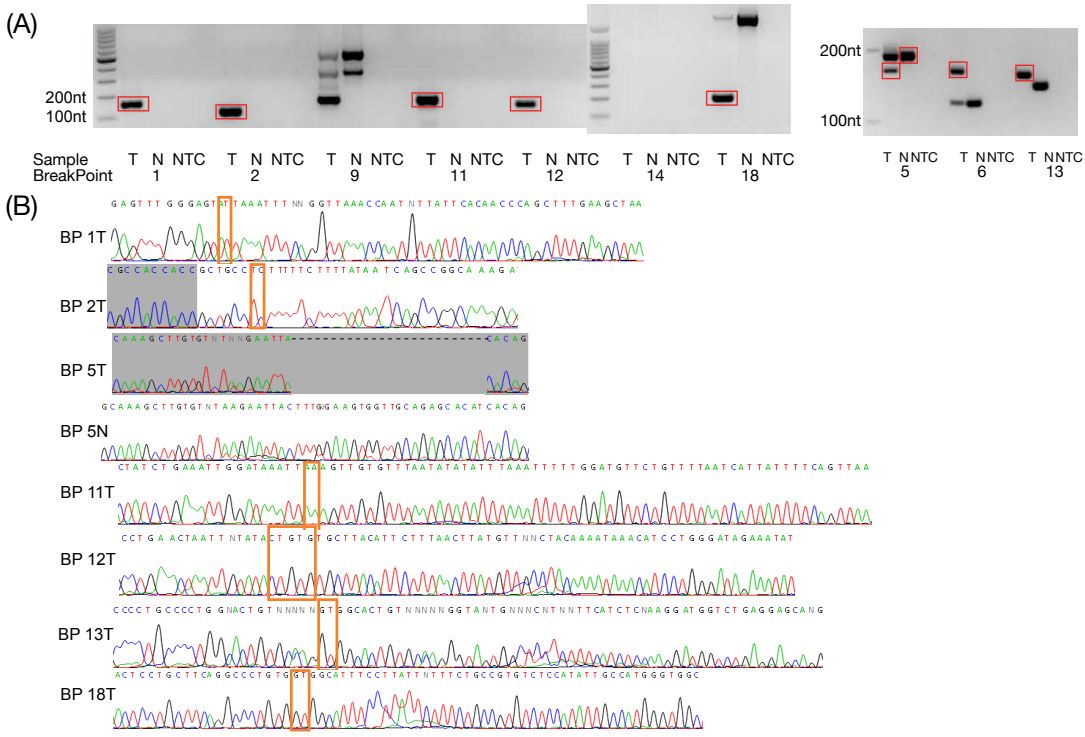
Supplementary Figure 6. Distribution of false negatives with feature size and type for CREST (top) and DELLY (bottom)



Supplementary Figure 7. Distribution of false negatives with feature size and type for Pindel (top) and Socrates (bottom)



Supplementary Figure 8. Repeat type composition of Socrates output data. The x-axis indicates the minimum MAF of a breakpoint to be included in the output. The numbers on the bars indicate the size of the filtered output set. Red bars indicate non-repetitive sequence around both clusters of a breakpoint.



SUPPLEMENTARY FIGURE 9. Validation of selected Socrates predictions in melanoma. (A) Agarose gels of PCR products, red boxes indicate band confirmed by Sanger sequencing. (B) Sequences from PCR products for all confirmed brakepoints, orange boxes indicate breakpoint.