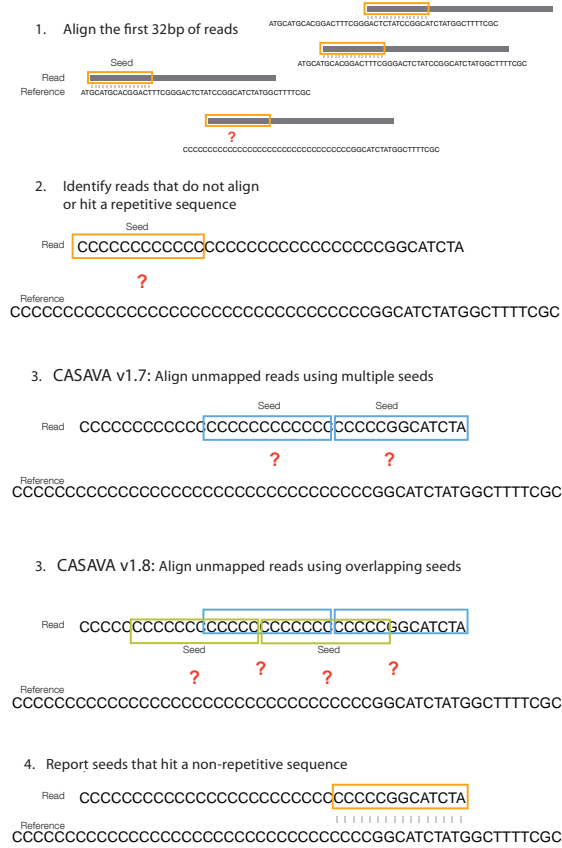


Figure 2: Multiseed ELAND Changes



CASAVA 1.8 multiseed ELAND changes result in more reads aligned in repeat regions. The big difference is in step 3, where CASAVA 1.8 uses overlapping seeds.

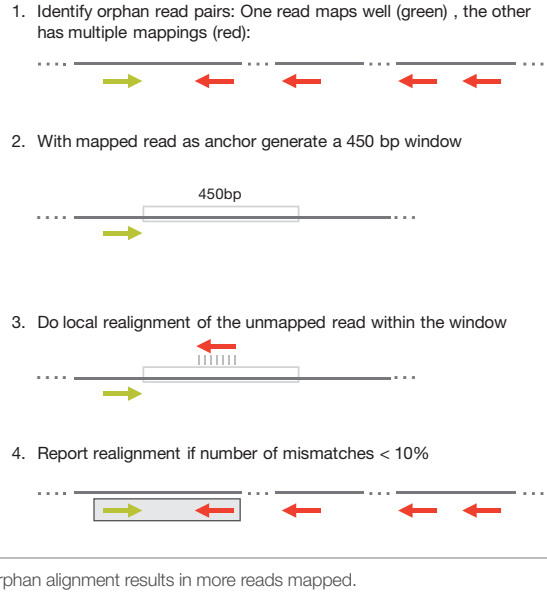
Orphan Alignment

ELANDv2e performs orphan alignment by identifying read pairs for which only one of the reads aligns. ELANDv2e tries to align the other read in a defined window (by default 450 bp). If the number of mismatches is <10% of the read length, ELANDv2e reports the alignment (Figure 3).

Table 1: Alignment and Mismatch Rates

	v1.7		v1.8, semi-repeat		v1.8, full repeat	
	% Align	% Mis-match	% Align	% Mis-match	% Align	% Mis-match
Read 1	84.56	0.70	88.29	0.72	90.17	0.73
Read 2	81.92	1.39	85.81	1.44	87.56	1.44

Figure 3: Orphan Alignment



Alignment Performance Improvements

The multiple component updates in CASAVA were designed to improve overall alignment performance. To assess the performance change, alignment percentage, mismatch rates, and CPU run times were compared for three different configurations: CASAVA 1.7, CASAVA 1.8 with semi-repeat resolution, and CASAVA 1.8 with full repeat resolution. The data set consisted of three lanes of HiSeq® data from a single sample sequenced with TruSeq v3 chemistry; the analysis was performed on an iCompute cluster with j = 32.

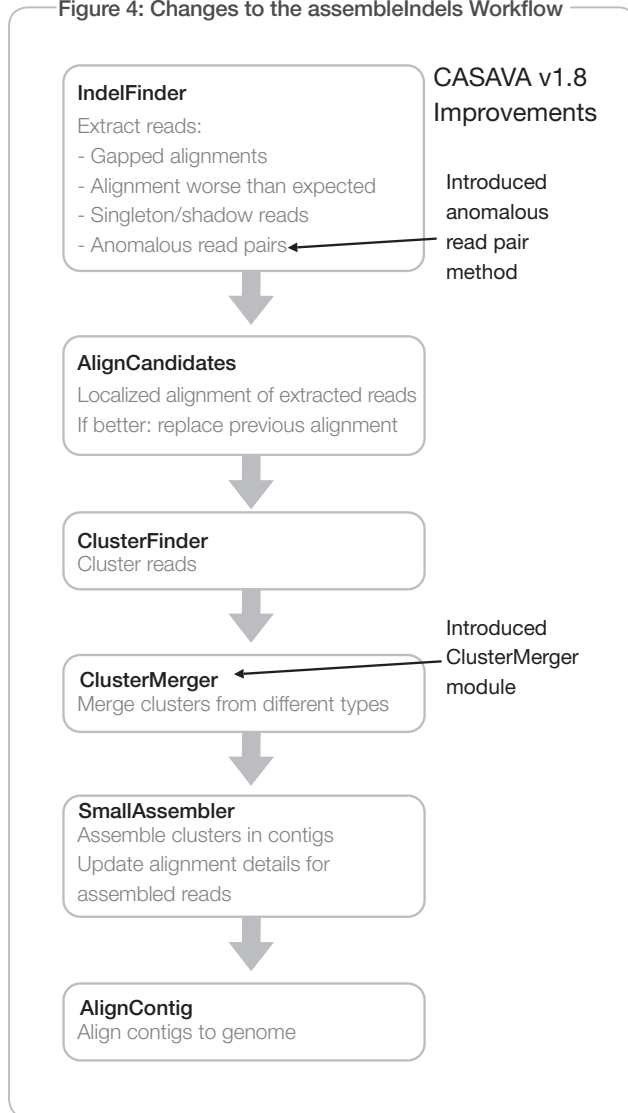
As demonstrated in Table 1, CASAVA 1.8 aligns a higher percentage of reads, with full repeat alignment performing best. This higher alignment rate results from the improved ability to align in more challenging repeat regions. Remarkably, even with more reads aligned in repeat regions, mismatch rates are still very similar.

Table 2: CPU Run Time Comparison

	v1.7 (CPU hours)	v1.8 sr, (CPU hours)	v1.8 fr, (CPU hours)
ELAND	523.28	518.40	855.40
orphanAligner	N/A	54.17	31.20
PickBestPair/alignmentResolver	200.77	14.67	14.97
produceAlignStats	21.65	12.43	14.55
Other Processes	25.99	0.17	0.20
Total	771.72	599.85	916.33

sr: semi-repeat, fr: full repeat.

Figure 4: Changes to the assembleIndels Workflow



While CASAVA 1.8 provides the highest percentage of aligned reads, this level of performance does require additional computational time (Table 2). For the ELAND step, 1.8 full repeat resolution takes quite a bit longer to run than semi-repeat resolution (520 hours versus 855 hours). Therefore, researchers should consider the trade-off between higher performance and slower run time to select the type of analysis best suited for their project.

Other algorithms have been updated in CASAVA 1.8 to improve run times. The module alignmentResolver (previously called PickBestPair) has been rewritten, which has resulted in much faster run times for this step (200 hours for 1.7, versus 15 for 1.8).

The best analysis type therefore depends on the project: is a shorter run time more important, or the highest number of aligned reads.

Indel Finding Improvements

Indels are identified using gapped alignments (callSmallVariants module) and local contig assembly (assembleIndels algorithm). Typically gapped alignments can be used to efficiently identify relatively small indels (roughly 1–10 bases in length), whereas local contig assembly (assembleIndels) can efficiently identify much larger indels. The greatest indel sensitivity can be achieved by generating candidate indels from both of the sources.

In CASAVA 1.8, assembleIndels indel finding has been expanded to use more read pair information.

assembleIndels Module Improvements

The major changes for the assembleIndels module (Grouper) are (Figure 4):

- assembleIndels uses an additional method to identify indels. It finds read pairs that map anomalously (for example, with unexpected insert size), and identifies potential indels.
- assembleIndels merges indel calls detected through anomalous read pairs with those identified through singleton/orphan reads, and combines clusters that appear to correspond to the same event.

assembleIndels Algorithm

The assembleIndels module (Grouper) runs only during paired-read DNA CASAVA builds. In CASAVA 1.8, it uses orphan reads and anomalous read pairs to detect indels.

assembleIndels detects indels in five stages (Figure 5):

1. Compute clusterings of non-aligned 'orphan reads'.
2. Compute clusterings of anomalous read pairs, with an insert size that is anomalously large (possible deletion) or small (possible insertion).
3. Combine clusters that appear to correspond to the same event.
4. Assemble them into contigs.
5. Align the contigs back to the genome, using the positions of associated 'singleton' reads to narrow the search to a couple of thousand bp or so.

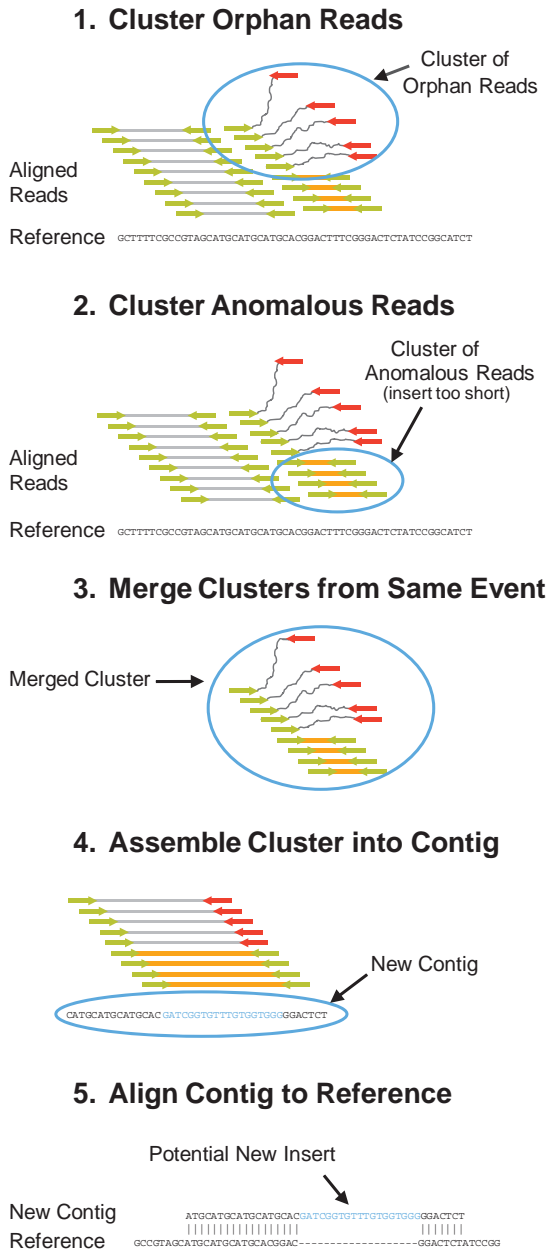
Candidate Indels

CASAVA combines the potential indels identified by the assembleIndels module and by gapped alignments (note that the evidence from gapped alignments and the assembleIndels module will be combined if both are available). It then identifies the candidate indels, based on the number of read alignments that contain the indel. These alignments may be from the primary alignment or from reads used by the assembleIndels module to assemble each contig.

If the number of reads supporting a potential indel is less than 3 or less than 2% of the total depth at the indel site, the indel cannot become a candidate. For short indels (≤ 4 bp), the number of supporting reads must be $\geq 10\%$ of the total depth for the indel to become a candidate. All other potential indels become candidate indels, subject to realignment and indel calling.

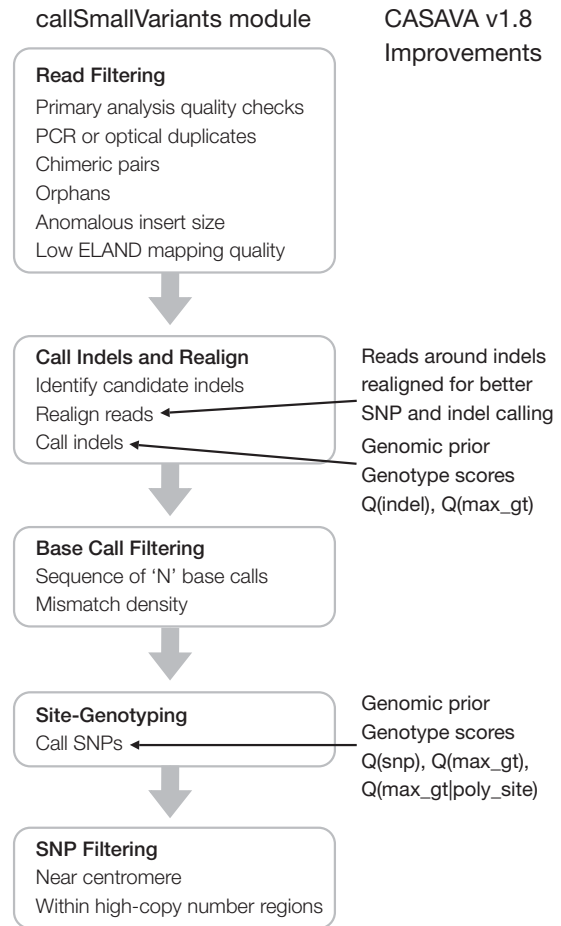
GTAAAGAATGATAACAGTAACACACTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTGAGACTAAATATAACGTACCATTAAAGAGCTACCCTCTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACG
 AAAATCAACGTACCGTAACGAACGTATCATTAAGATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTGAGACTAAATATAACGTACCATTAAAGAGCTACCCTCTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACG
 CGTAACGACGAAAAGAAATGATAACAGTAACACACTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTGAGACTAAATATAACGTACCATTAAAGAGCTACCCTCTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACG
 CATTAACGTACCATTAAAGAGCTACCCTGTAACGAGTAACACACTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTGAGACTAAATATAACGTACCATTAAAGAGCTACCCTGTAACGAGTAACACACTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACG
 GTAAAGAATGATAACAGTAACACACTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTGAGACTAAATATAACGTACCATTAAAGAGCTACCCTCTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACG
 GAAAAGATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTGAGACTAAATATAACGTACCATTAAAGAGCTACCCTCTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTGAGACTAAATATAACGTACCATTAAAGAGCTACCCTCTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACG
 AAACAGGTATCAATTGAGACTAAATATAACGTACCATTAAAGAGCTACCCTGTAACGAGTAACACACTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTGAGACTAAATATAACGTACCATTAAAGAGCTACCCTGTAACGAGTAACACACTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACG
 AAAATCAACGTACCGTAACGAACGTATCATTAAGATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTGAGACTAAATATAACGTACCATTAAAGAGCTACCCTGTAACGAGTAACACACTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACG
 GAAAAGATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTGAGACTAAATATAACGTACCATTAAAGAGCTACCCTCTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACGTACCGTAACGAACGTATCAATTGAGACTAAATATAACGTACCATTAAAGAGCTACCCTCTTCTGTAAACCTTAAGATTACTTGATCCACTGATTCAACG

Figure 5: assembleIndels Algorithm



The five stages of indel detection by assembleIndels.

Figure 6: Changes to the Small Variant Calling Workflow



Variant Calling Improvements

The improvements in variant calling are (Figure 6):

- SNP and small-indel calling are now unified into a single process in the small-variant calling module, allowing both SNP and indel calls to be based on a consistent set of read realignments, including reads re-aligned by the assembleIndels module.
- The indel-caller has been redesigned to gather candidate indel information from both gapped alignments and from the assembleIndels module contig assemblies, which greatly improves sensitivity. This also means that small indels can be called directly from gapped alignments without running assembleIndels if a quick analysis is desired.
- The indel-caller has been extended from the CASAVA 1.7 method to better handle dense and overlapping indels, further improving indel sensitivity. Additionally, the indel-caller has been extended to report and genotype breakpoints for cases where the complete indel is either unknown or too large to be represented by the indel-calling model.
- Reads are now locally realigned as part of the small variant calling process, reducing the possibility of false positive SNPs due to reads which overlap an indel by only a few bases.

