**SNP and Indel Detection**

**1. From reads to pileup**

1.1 Tomato ((Solanum lycopersicum) reference genome (build SL2.40) was downloaded from Sol Genomics Network. The reference genome in the FASTA format was first indexed with the “BWA index” utility to construct suffix array.

1.2 Although all sequence libraries are either paired-end or mate paired, all reads were treated as if they were single-end for the purpose of SNP and indel detections. “BWA (version 0.5.9-r16) aln” command was used to map a read to the reference genome. The input to the program is a FASTQ file containing individual reads, and the program outputs the suffix array coordinates of mapped reads in a SAI file. For reads with different lengths, the following parameters were used (table 1). In the table, -n option refers to the maximal edit distance allowed between the whole read sequence and the aligned reference genome, -o option refers to the maximum number of gap openings, and –e option refers to the maximum number of gap extensions. Overall the maximal edit distance, which includes both mismatches and gap openings, was set to be less than 4% of the read length.

Table 1. BWA alignment parameters for single reads with different lengths

|  |  |
| --- | --- |
| read length (bp) | Alignment parameter |
| 42 | -n 1 -o 1 -e 0 |
| 79 | -n 3 -o 1 -e 1 |
| 80 | -n 3 -o 1 -e 1 |
| 101 | -n 4 -o 1 -e 2 |
| 151 | -n 6 -o 2 -e 2 |

1.3 “BWA samse” command was used to generate alignments in the SAM format. The program takes SAI file as input and converts suffix array coordinates to chromosomal positions.

1.4 Given a SAM input from 1.3, a C++ program, rmRedunSam, was developed to filter reads. The program selects reads that are uniquely mapped to a chromosome. Furthermore, the uniquely mapped reads need to bear a CIGAR string, a mandatory field in the SAM format, which contains only M (for match/mismatch), I (insertion), or D (deletion) operations. Then, the C++ program removes potential PCR duplicates. Two individual reads are deemed to be potential PCR duplicates if they meet the following 3 conditions: (a) both reads mapped to the same chromosome; (b) both reads mapped to the same strand of the chromosome; (c) either starting or ending coordinates for the two reads in the chromosome are identical. If two reads are duplicates, the read with less edit distance is retained. If two duplicated reads have same edit distance, the read with smaller number of gaps is retained. The program outputs the filtered reads in the SAM format.

1.5 samtools program (version 0.1.16-r963:234) was applied to take the filtered SAM file as the input, convert SAM to BAM format, and sort reads based on their mapped leftmost coordinates.

1.6 A pipeline comprising the steps from 1.2 to 1.5 was developed in Perl. Each Pennellii (or M82) FASTQ file is subject to the pipeline processing and produces a sorted BAM file. A total of 82 FASTQ files were processed for Pennellii, and 18 FASTQ files for M82.

1.7 All sorted BAM files for each species were merged into a single BAM file by using “samtools merge” command.

1.8 Use “samtools mpileup” command to generate pileup for the single BAM file with the option “ -Q 0 -d 1000000”.  Under this option, mpileup will count low-quality bases (-Q 0) and essentially process all reads (by default the depth is capped at 8000).

**2. Pennellii SNP detection with respect to reference genome**

A C++ program, pileupFilter, was developed to detect SNPs and small indels given a Pennellii pileup file.

For SNP detection, we require a position in the reference genome to be covered by at least 5 reads which exclude the ones with a base calling of ‘N’ at that position. If at least 90% of reads have deletions or different nucleotides with respect to the base in the reference genome, then we call mutation (substitution or deletion) at that position. In the case a reference base is N, no SNP will be detected.

For deletion detection, a position in the reference genome needs to be covered by at least 5 reads excluding the ones with start or end mapped to that position. If at least 80% of the reads have deletions at the position, then a deletion event occurs.

For insertion detection, a position in the reference genome needs to be covered by at least 5 reads excluding the ones with end mapped to that position. If at least 80% of the reads have insertions at the position, then an insertion event occurs.

**3. M82 SNP detection with respect to reference genome**

The C++ program was also applied to detect SNPs and small indels given a M82 pileup file. The parameter setting is same as Pennellii.

**4. Pennellii SNP detection with respect to M82**

To directly compare Pennellii with M82 (M82 as reference), another C++ program, combine2PileFiles, was developed to detect SNPs and indels by comparing Pennellii with M82 pileup file.

For SNP detection, we require a position in both Pennellii and M82 pileups to be covered by at least 5 reads which exclude the ones with a base calling of ‘N’ at that position. Then we calculate the percentages of A/C/G/T among reads’ nucleotides for Pennellii and M82, respectively. If the dominant nucleotide for Pennellii and the dominant nucleotide for M82 are both at least 90%, and if they are different, then an event of SNP occurs.

For deletion detection, a position in both Pennellii and M82 pileups needs to be covered by at least 5 reads excluding the ones with start or end mapped to that position. Then we calculated the percentage of deletions (the number of deletions divided by the number of reads) for Pennellii and M82, respectively. If Pennellii has a deletion percentage of at least 80% and M82’s deletion percentage is at most 10%, then an event of deletion occurs to Pennellii with respect to M82. On the other hand, if the deletion percentage for Pennellii is less than or equal to 10% but M82 is at least 80%, then an event of insertion occurs to Pennellii with respect to M82.

For insertion detection, a position in both Pennellii and M82 pileups needs to be covered by at least 5 reads excluding the ones with end mapped to that position. Then we calculated the percentage of insertions (the number of insertions divided by the number of reads) for Pennellii and M82, respectively. There are several scenario needed to be considered.

(I) If Pennellii has an insertion percentage of at least 80% and M82’s insertion percentage is at most 10%, then an event of insertion occurs to Pennellii with respect to M82. On the other hand, if the insertion percentage for Pennellii is less than or equal to 10% but M82 is at least 80%, then an event of deletion occurs to Pennellii with respect to M82.

(II) If both species have insertion percentages greater than or equal to 80%, then we compare what base(s) are inserted between the two species. We calculate the percentage of dominant inserting base(s) among all insertions for each species. In the following pileup example,

SL2.40ch01 26806 T 7 ,+2aa.+2AA.+2AA,+2aa.+2AA,+2aa.+1T

all 7 reads have insertions between the reference position 26806 and 26807. Six reads insert ‘AA’ and 1 read inserts ‘T’. Thus, the percentage for dominant inserting base(s), ‘AA’, among all insertions is 85.7% (6/7). Given the dominant inserting base(s), two bases are deemed to be inserted.

If the percentages of dominant inserting base(s) are at least 80% for both species, then we consider the following scenario.

(IIA) if the number of bases inserted for Pennellii is same as M82, and if the inserting bases are different between two species, then we simply assign an event of SNP.

(IIB) if the number of bases inserted for Pennellii is greater than M82, we simply call an event of insertion for Pennellii with respect to M82.

(IIC) On the other hand, if the number of bases inserted for Pennellii is less than M82, we simply call an event of deletion for Pennellii with respect to M82.