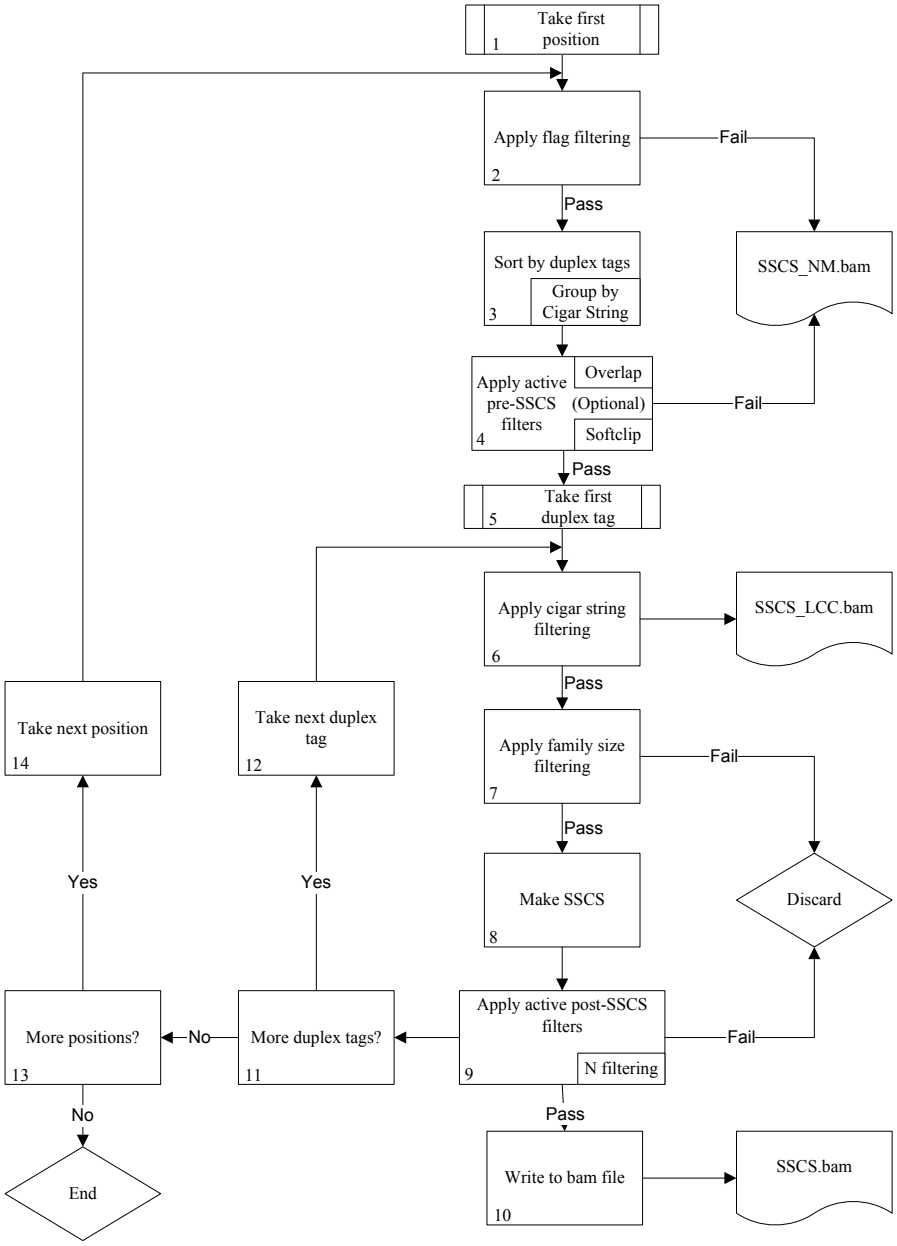

Algorithm for ConsensusMaker.py.

ConsensusMaker.py takes a Samtools generated sorted paired-end *.bam file as input. The script attempts to load all reads that map to the same genomic position (i.e. reads sharing the same POS field) into memory, but applies several filters while doing so. Importantly, all PCR duplicates needed for the creation of a SSCS molecule will map to the same position in the reference genome, and will share the same tag sequence. Reads can be filtered based on the value of their bitwise flag (i.e. FLAG field) (Step 2). We currently group and filter read by four main classes: 1) reads forming a proper read pair (FLAG values 83, 99, 147, 163), 2) paired, but not properly paired (FLAG values 65, 81, 97, 113, 129, 145, 161, and 177), 3) single-mapping reads (FLAG values 69, 73, 89, 117, 133, 137, 153, and 181), and 4) non-mapping read pairs (FLAG values 77 and 141). We recommend filtering out the non-mapping read pairs and any other FLAG values not

indicated by the other three classes of reads. Reads not passing the FLAG filter are saved in a file (SSCS_NM.bam) in case they are needed for troubleshooting purposes. The filtered reads sharing the identical tag sequence are further sorted and sub-grouped by their CIGAR string (Step 3). Reads can also be optionally filtered to remove reads that overlap at their 3'-end or harbor soft-clipped bases (Step 4). Overlapping reads are found by first adding the read length to the value in the POS field and then comparing this sum to the value in the MPOS field. If the sum is greater than the MPOS field value, the read is considered overlapping. Soft-clipped reads are determined by the existence of a 'S' in the CIGAR string. Reads failing these filters can be written to a separate file (SSCS_NM.bam) or discarded from further analysis.

Within each family, the number of reads with each CIGAR string is checked to find the most common CIGAR string. Only reads sharing the most common CIGAR string are considered for consensus making; all others can written out to a separate file (SSCS_LCC.bam) or discarded (Step 6). For example, if a family has four reads, three of which have a cigar string of 84M, and one of which has a cigar string 30M1D54M, only those reads with a cigar string of 84M will be considered as members of the family for consensus making. Those reads that remain are then checked to see if the number of reads in a tag family is greater than a minimum cutoff value (Step 7). We currently use a miminum membership of three reads. In addition, we have also instituted a maximum membership cutoff that limits the family size and reduces the computational time to process large families. We currently set this limit to 1,000 members. If the number of reads is below the minimum cutoff, the entire tag family is discarded and a consensus is not made. If the family size is greater than the maximum cutoff value, then the number of reads corresponding to the maximum cutoff value are randomly selected from the family and used to make the consensus. The rest of the reads in the family are ignored.

After all the filters are applied, reads that share the same tag sequence and CIGAR string are subjected to consensus making (Step 8). The consensus sequence each position in a read is determined by a majority rules algorithm. Specifically, for each position index, the proportion of A's, T's, C's, and G's is calculated from every read in a tag family (e.g. #A's / # reads in tag family). The consensus for each position index is the base that occurs with a proportion greater than a user defined minimum value. If no proportion is greater than the minimum cutoff, the position is considered undefined and a 'N' is used as the consensus value for that position. Following consensus making, the SSCS may optionally be filtered based on the proportion of positions containing N's (Step 9). At this stage, the duplex tag from the header field, the genomic position, the cigar string, the flag, and the mate position are retained from the original reads, while the quality scores, mapping quality and optional tag fields can either replaced with dummy values or discarded. The final SSCS is written to a file (SSCS.bam) (Step 10). If additional tag families are available for consensus making that map to the same genomic position (i.e. share the same value for the POS field), then they are submitted for consensus making using the same filtering and consensus making algorithms (Steps 11 & 12). If no more families are available, all the reads mapping to the next position in the reference genome are loaded into memory, and the process begins again (Steps 13 & 14).