```
1 procedure GET_VARIANTS(bam, amplicon_coords, abs_thresh, proportion_thresh):
2       called_variants := ∅
3       # Process each amplicon coordinate for a given sample BAM file
4       for (chromosome, start, end) ∈ amplicon_coords:
5             amplicon_variants := ∅
6             num_read_pairs := 0
7             # Collect all the read pairs which overlap the amplicon
8             read_pairs := READS_OVERLAP_AMPLICON(BAM, chromosome, start, end)
9             for (read1, read2) ∈ read_pairs:
10                  num_read_pairs++
11                  # Compute the variants for each read in the pair
12                  variants1 := READ_VARIANTS(read1)
13                  variants2 := READ_VARIANTS(read2)
14                  # Collect the variants which appear in both reads in the pair
15                  shared_variants := variants1 ∩ variants2
16            # Only keep variants which are inside the amplicon
17            for variant ∈ shared_variants:
18                  if start ≤ variant.position ≤ end:
19                        # Count the frequency of this read-pair variant
20                        if variant ∈ amplicon_variants:
21                              variant.count++
22                        else:
23                              variant.count := 1
24                              INSERT(variant, amplicon_variants)
25            # keep only those read-pair variants which are above our thresholds
26            for variant ∈ amplicon_variants:
27                  variant_proportion := variant.count / num_read_pairs
28                  if (variant.count ≥ abs_thresh and
29                      variant_proportion ≥ proportion_thresh):
29                        INSERT(variant, called_variants)
30      # Return the set of all variants called for this sample BAM file
31      return called_variants
```