SNP Caller Implementation

Anh Tran

2023-03-24

Use requirements

This SNP Caller Implementation uses the following libraries:

```
pandas -- 1.5.3
numpy -- 1.23.5
pysam -- 0.20.0
```

In order for the code to be executed, indexed bam file is needed in your working directory in addition to the reads.bam file.

To make a bam index file I used `samtools -- 1.17`

Then I ran the following code in terminal:

```
samtools index input.bam
```

The code can be executed as: `python3.9 snpcaller reads.bam metadata.tsv`

Other versions of python should work as well.

Deliverables

`snpcaller.py --` contains the code for the SNP Caller Implentation

`snpcaller.py --` will output an `output_file.tsv--` that contains the putative genotype, posterior probabilities of all three genotypes, chromosome position, reference and alternate alleles, as well as the number of overlapping reads.

`metadata.tsv --` this could will work for any file that follows the `putative_snps.tsv` format

Math Model

Determining Prior Probabilites

Given the minor allele frequencies (maf) for each putative SNP in the `putative_snps.tsv` the prior probabilities can be calculated using the Hardy Weinberg Equation.

Hardy-Weinberg Equation:

$$p + 2pq + q = 1$$

The Hardy-Weinberg Equation provides the predicted frequencies of all three genotypes within a population under the five basic assumptions: no mutation, random mating, no gene flow, infinite population size, and no selection.

Under this model with A representing the major allele and B representing the minor allele, the genotype frequencies will be as follows:

$$AA = p^2$$
$$AB = 2pq$$
$$BB = q^2$$

Given the minor allele frequency: $q = maf$ and $p = 1 - maf$ while $2pq = 1 - q - p$.

Interpeting Errors from Phred Scores

Provided in the BAM (Binary Alignment Map) file are quality scores. A quality score measures the accuracy of the base calls in a sequence read. Each base call is given a quality scores and they are on a Phred-scaled probability value. This allows us to find the corresponding error for each putative SNP.

Calculating error from a quality score goes through a simple conversion:

$$P(E_i = 1) = 10^{-Phred\ Score/10}$$

Thus, $P(E_i = 0) = 1 - P(E_i = 1)$.

Calculating Posterior Probabilities

After the calculating the prior probabilities and the corresponding error for each putative SNP. We can calculate the likelihoods for each genotype:

$$\begin{aligned}
P(E_1, E_2, ..., E_n|AA) &= \prod_{i:A} P(E_i = 0) \prod_{i:B} P(E_i = 1) \\
P(E_1, E_2, ..., E_n|BB) &= \prod_{i:B} P(E_i = 0) \prod_{i:A} P(E_i = 1) \\
P(E_1, E_2, ..., E_n|AB) &= (\prod_{i:A} P(E_i = 0|S = A)P(S = A) \prod_{i:B} P(E_i = 1|S = B)P(S = B)) \\
&\quad (\prod_{i:B} P(E_i = 0|S = B)P(S = B) \prod_{i:A} P(E_i = 1|S = A)P(S = A))
\end{aligned} \tag{1}$$

Under the assumption that $P(S = A) = P(S = B)$, $P(E_1, E_2, ..., E_n|AB)$ should be reducible to:

$$P(E_1, E_2, ..., E_n|AB) = \prod_{i:N} \frac{1}{2}$$

The posterior probabilities of each genotype can be calculated using Bayes' Theorem as follows:

$$P(AA|E_1, E_2, ..., E_n) = \frac{P(E_1, E_2, ..., E_n|AA)P(AA)}{P(E_1, E_2, ..., E_n)}$$
$$P(BB|E_1, E_2, ..., E_n) = \frac{P(E_1, E_2, ..., E_n|BB)P(BB)}{P(E_1, E_2, ..., E_n)}$$
$$P(AB|E_1, E_2, ..., E_n) = \frac{P(E_1, E_2, ..., E_n|AB)P(AB)}{P(E_1, E_2, ..., E_n)}$$

Approach

Assuming a Biallelic Model,the goal of my code is the calculate the posterior probabilities for a set of putative SNPs based on their read coverage in a BAM file.

First, I read in both the BAM file and the `putative_snps.tsv` using PySam and Pandas libraries.

For each putative SNP in the SNP file, I extracted the chromosome, position, reference and alternate alleles, as well as the maf using Pandas.

Then I calculated the prior probabilities for the reference, alternate, and heterozygous genotypes based on the maf using the Hardy-Weinburg equation mentioned in the math model.

Using the PySam's fetch and pileup functions, I ran a for loop that found the number of overlapping reads. Then, I iterate through each read at the SNP position and count the number of reads supporting the reference and alternative alleles in the `putative_snps.tsv`.

Using the PySam library, I was able to find the read's quality score for the SNP position. Using the conversion mentioned in the math model, I was able to convert the Phred-based quality score into $P(E_i = 1)$.

I then stored all the information mentioned above in a DataFrame, including; everything from the `putative_snps.tsv` file, overlap counts, reference and alternative allele counts, quality scores, $P(E_i = 1)$, and prior probabilities.

Using the information stored in the DataFrame, I ran a second loop that calculated the likelihoods and posterior probabilities for each overlapped read. I used the counts of the reference and alternate alleles at that SNP position to run loops that would calculate the likelihood. The likelihoods and posterior probabilities were calculated using sum of log method to avoid underflow. Finally, the results will be outputted in an `output_file.tsv--` that contains the putative genotype, posterior probabilities of all three genotypes, chromosome position, reference and alternate alleles, as well as the number of overlapping reads.

The putative genotype was the genotype with the highest probability in my model. Although, all there posterior genotypes from the are outputed in the the final result.

Results

Using the `putative_snps.tsv` and the given BAM file, the overlapped reads contained either all reference or alternate alleles, never a mix of both or a an allele that was not a reference allele.

In addition to that, the quality scores extracted from the BAM file were all the same value, 17. So all the SNPs had the same $P(E_i = 1)$.

Under these circumstances, my putative genotype (the genotype with the highest probability) was consistently homozygous for either the reference or alternate allele that spanned the overlap. These results do seem consistent, as the $P(E_i = 1)$ value were the same for all SNPs positions and the overlapped reads contained either all reference or alternate alleles.

Under different data, the results would be different. Under the assumption of a biallelic model, if my code was to encounter a base in the BAM file that was not the reference or alternative allele, my code discards of the observation. Such an observation could mean that there could be a sequencing error. The way my code deals with it is by counting only the number of reads that match to the reference or alternative allele from the overlapped reads. So the likelihood and posterior probabilities will not count any reads that is not the reference or alternative allele. The code will be able to handle overlapped reads that have a mixture of both the reference and alternative alleles perfectly fine if that were to ever be the case. Lastly, in the case of none of the overlapped reads were to match to the reference or alternate alleles, then the likelihoods would remain initialized at 0 and the posterior probabilities will be undefined. But there are limitations to this approach. By discarding observations, I am losing data and fewer data points reduces the accuracy of predictions.

References

Phred quality score. (2022, March 20). In Wikipedia. Retrieved March 23, 2023, from https://en.wikipedia.org/wiki/Phred_quality_score

The SAM/BAM Format Specification Working Group. (2022). Sequence Alignment/Map Format Specification (Version 1). Retrieved from https://samtools.github.io/hts-specs/SAMv1.pdf