Unsupervised Ensemble Learning for Genome Sequencing

Alba Pagès-Zamora, Idoia Ochoa, Gonzalo Ruiz Cavero,
Pol Villalvilla-Ornat

PII: S0031-3203(22)00202-3
DOI: https://doi.org/10.1016/j.patcog.2022.108721
Reference: PR 108721

To appear in: *Pattern Recognition*

Received date: 8 August 2021
Revised date: 1 April 2022
Accepted date: 18 April 2022

Please cite this article as: Alba Pagès-Zamora, Idoia Ochoa, Gonzalo Ruiz Cavero, Pol Villalvilla-Ornat, Unsupervised Ensemble Learning for Genome Sequencing, *Pattern Recognition* (2022), doi: https://doi.org/10.1016/j.patcog.2022.108721

Highlights

- The variant calling step in next generation sequencing technologies is formulated as a classification problem.
- An unsupervised ensemble classification method is proposed as a variant caller for DNA sequencing.
- An EM-based variant calling algorithm that estimates the maximum a posteriori class to take a decision is presented.
- The number of classes to be decided is greater than the number of different labels that are observed.
- Experimental results with real human DNA sequencing data support the approach.

Unsupervised Ensemble Learning for Genome Sequencing

Alba Pagès-Zamora[a,*], Idoia Ochoa[b], Gonzalo Ruiz Cavero[b], Pol Villalvilla-Ornat[a,**]

[a] *SPCOM Group, Universitat Politècnica de Catalunya - BarcelonaTech (UPC), C/Jordi Girona 31, 08034, Barcelona, Spain (e-mail: alba.pages@upc.edu).*

[b] *Tecnun, University of Navarra, Manuel Lardizábal 13, 20018 San Sebastián, Spain (e-mail: iochoal@unav.es ; a905186@alumni.unav.es)*

Abstract

Unsupervised ensemble learning refers to methods devised for a particular task that combine data provided by decision learners taking into account their reliability, which is usually inferred from the data. Here, the variant calling step of the next generation sequencing technologies is formulated as an unsupervised ensemble classification problem. A variant calling algorithm based on the expectation-maximization algorithm is further proposed that estimates the maximum-a-posteriori decision among a number of classes larger than the number of different labels provided by the learners. Experimental results with real human DNA sequencing data show that the proposed algorithm is competitive compared to state-of-the-art variant callers as GATK , HTSLIB, and Platypus.

Keywords: expectation maximization algorithm, variant calling, genome sequencing, unsupervised multi-class ensemble classifier, GATK

1. Introduction

Ensemble learning refers to methods devised for a particular task that fuse data provided by decision agents, e.g., algorithms or annotators in crowdsourced applications, and infer the reliability of these agents to be considered when the data is combined. Comprehensive surveys can be found in [1, 2]. In particular, unsupervised ensemble classification deals with the problem of designing a meta-learner to classify objects without using ground-truth data to train the learners, and based only on the tags provided by individual decision agents [3]. Applications are found in diverse areas such as medicine and biology, e.g., [4][5] where decision agents are either algorithmic techniques or individuals, respectively; team decision-making strategies [6]; and 5G communication systems [7] where decision agents are sensors, among others.

Based on the seminal paper of Dawid and Skene [8], a vast majority of works estimate the statistics of the decision agents and use these estimates either to solve a Maximum Likelihood

[*] Corresponding author

[**] Present Address: ETH Zurich, Erdbeobachtung u. Fernerkundung, HCP G 33.1, Leopold-Ruzicka-Weg 4, CH-8093 Zurich, Switzerland (e-mail: pvillalvilla@ethz.ch)

(ML) detection problem or to initialize the Expectation-Maximization (EM) algorithm in [8]. For instance, works in [9, 10] solve a binary classification task and advocate a spectral decomposition technique of the second-order statistics of agent responses that yields the reliability parameters of the agents. In a multi-class setting, [11] utilizes third-order moments and orthogonal tensor decomposition to estimate the unknown reliability parameters and then initialize the EM algorithm of [8]. Also, in [12] a MAP approach is adopted to solve a multi-class ensemble classification problem using moment-based estimates of the confusion matrices of the decision agents. A broadly adopted assumption in the literature is to consider conditionally independent agents, meaning that there is no communication among different decision agents. Under this assumption, the likelihood function of the individual labels breaks down into factors so that the number of parameters of the likelihood function is reduced and the ensemble classification problem becomes more doable.

As evidence in [13], pattern recognition plays a pivotal role in the area of bioinformatics with significant contributions to DNA sequence analysis, including sequence comparison and gene prediction, and DNA microarray data analysis. For instance, a spectral distortion measure is proposed in [14] for finding similarities between DNA or protein sequences, which helps life-science researchers understand the information content and functions of biological sequences. In [15] the distribution of D_2 statistic, widely used in alignment-based methods, is analysed to give usable approximations for ranges of parameters frequently encountered in the study of biological sequences. Further, an spectrum analysis approach based on the Fourier transform is proposed in [16] to solve the prediction of exons, which are protein coding subregions. Regarding array data analysis, [17] presents a sparse regularized Tucker tensor regression approach to perform feature selection on genomic data. All these techniques apply to sequenced genomes, and this paper focuses on the use of ensemble learning methods to improve the genome sequencing pipeline.

As it will be explained, the procedure followed by Next Generation Sequencing (NGS) technologies for DNA[1] sequencing includes a step named *variant calling*. In brief, variant calling consists on the identification of *variants* or discrepancies between (i) the sequenced genome and (ii) that of a reference genome of the same species as the sequenced genome. Due to the high similarities across genomes of the same species, the called variants are useful to compress the information of a sequenced genome since a genome can be reconstructed given the identified variants and the reference genome. Therefore, the called variants are generally the input to downstream analyses. In the clinical setting, the identification of variants is critical to diagnose, design treatments, and study cancer development, among others. For example, in

[1] DNA stands for Deoxyribonucleic Acid.

[18] they show that variants in specific genes can cause sever hypercholesterolemia, and in [19] they discover previously undetected mutations (variants) in genes BRCA1, BRCA2, CHEK2, TP53, and PTEN that increase the risk of breast cancer. As such, precision in variant calling is of utmost importance.

Several algorithms for variant calling exist, with the most relevant specially in the clinical and research setting [20] being: i) the Genome Analysis Toolkit (GATK) software package[2] first introduced in 2011 [21] and subsequently updated since 2013 [22] (last release in Feb. 2022), which is recommended by the Broad Institute in its *Best Practices* pipeline for genome sequencing [23] and currently one of the most-widely used; ii) the High Throughput Sequencing LIBrary (HTSLIB)[3], which uses the Samtools suite developed by The Wellcome Trust Sanger Institute [24] (last release in Feb. 2022); and iii) Platypus, developed by Oxford University [25]. These unsupervised methods employ Bayesian statistics to infer the existence of a variant, and although they produce similar sets of variants, differences exist. Moreover, it has been reported that the set of identified variants generally contain many incorrectly called variants and miss several true ones [26]. Hence novel variant callers that can improve on the accuracy of the identified variants are important.

In this work we show that variant calling can be formulated as an unsupervised multi-class ensemble classification problem, and present EMVC, an ensemble classifier based on the EM algorithm that solves the variant calling step. The performance of the EMVC algorithm is evaluated using real genomic data available for chromosome 20 of one particular human individual denoted by the name NA12878. The genome of this individual has been thoroughly characterized by the National Institute of Standards and Technology's (NIST) Genome In A Bottle (GIAB) consortium, and a set of high-confidence variants, i.e., gold standard or ground truth, exists [27]. The numerical experiments presented in our work show that EMVC results are competitive to those obtained with the state-of-the-art variant callers GATK, HTSLIB, and Platypus.

The paper is organized as follows. Section 2 provides an overview of the pipeline followed by NGS technologies for DNA sequencing. Then, in Section 3 we present the data model and formulate the variant calling step as an unsupervised ensemble classification problem. This problem is solved by means of an EM-based algorithm presented in Section 4. The performance of the proposed algorithm is assessed using a real data, and results are compared to GATK, HTSLIB, and Platypus in Section 5. Finally, Section 6 concludes the paper.

[2] Available at `https://gatk.broadinstitute.org/hc` and `https://github.com/broadinstitute/gatk/releases` (last Release in Feb. 2022)

[3] Available at `http://www.htslib.org` and `https://github.com/samtools/samtools/releases/` (last release in Feb.2022)

Notation: Unless otherwise noted, lowercase bold letters, $\mathbf{x}$, denote vectors, uppercase bold letters, $\mathbf{X}$, represent matrices, and calligraphic uppercase letters, $\mathcal{X}$, stand for sets. The (i, j)th entry of matrix $\mathbf{X}$ is denoted by $\mathbf{X}(i, j)$.

2. Genome Sequencing Overview

DNA chains consists of two strands of millions of nucleobases of type $\{A, C, G, T\}$[4] arranged following a helicoidal shape so that each nucleobase on one strand chemically bonds with another nucleobase on the other strand. Each pair of bonded nucleobases, which can be either GC or AT, are called *base-pair* and they constitute the building block of the DNA. The DNA is stored in the nucleous of cells and it is organized into chromosomes, whose number and length differ between organisms. Cells might be classified into diploid or haploid, where diploid means that the cell has two different copies of each type of chromosome as opposed to haploid when the cell has one copy only. For instance, human cells are diploid with 23 pairs of chromosomes so that, in a simplified way, each copy is inherited from one of the progenitors, i.e., mother and father.

NGS technologies provide massive parallel sequencing techniques that yield yet to be envisioned research opportunities in the biological science field. As a groundbreaking application, individualized therapies based on the patient's genome have nowadays become a reality thanks to these low-cost NGS technologies. These technologies take advantage of the similarity among the genome of individuals belonging to the same species. As an example, human DNA is 3×10^9 base-pairs long but, on average, DNA of two human beings differs in only 0.1% [28]. This evidence allows to establish a *reference* genome for each species, which is usually avaliable and obtained through more advanced expensive sequencing methods.

NGS technologies for DNA sequencing proceed as follows. First, as shown in Fig. 1, a library of the DNA sample is prepared so that multiple copies of the same DNA are cut into small fragments, typically of the order of hundreds of base-pairs. Then, these fragments are sequenced by a parallel sequencing machine, which performs the *base calling*. Each sequenced DNA fragment is called *read* and is an ordered sequence of hundreds of nucleobases. The ordered sequence of nucleobases corresponds to one strand of the DNA fragment, being the other strand obvious since base-pairs can be either AT or GC. Indeed, the reads can come from either one strand, and they can be thought of as short strings sampled at random from the original genome. Thus, NGS technologies produce a collection of millions of fragments of hundreds of nucleobases, called reads, instead of the whole genome sequence. It is important to

[4] Adenine, Cytosine, Guanine, and Thymine.

take in mind that in diploid cells reads indistinctly correspond to one of the two copies of the chromosome of the DNA sample. Also, since base calling is subject to errors, each read comes with a sequence of quality scores (Q-scores) of the same length indicating the reliability of each nucleobase of the read. The reads and the corresponding Q-scores are stored in the widely used FASTQ format.

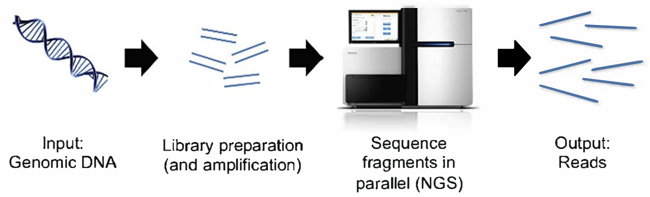

Figure 1: NGS technologies require a library preparation of the DNA sample, which includes cutting the DNA into small fragments that are used as input to the sequencing machine which performs the sequencing in parallel. Figure extracted from [28].

In the typical analysis pipeline, the next step after the base calling is the *alignment* process that determines the location of each read in the reference genome of the species to which the DNA sample belongs to. This is achieved through a mapping algorithm that compares the sequence of each read to the reference genome, and tries to locate the segment of the reference sequence that matches the read, while tolerating a certain amount of mismatches. The alignment information is stored in the standard SAM format [24] together with the original reads and the Q-scores. Figure 2 describes the pipeline of genome sequencing with the corresponding files generated at each step. The sizes of the files correspond to a human genome with a *coverage* of 200, which indicates the average number of reads per nucleotide position.

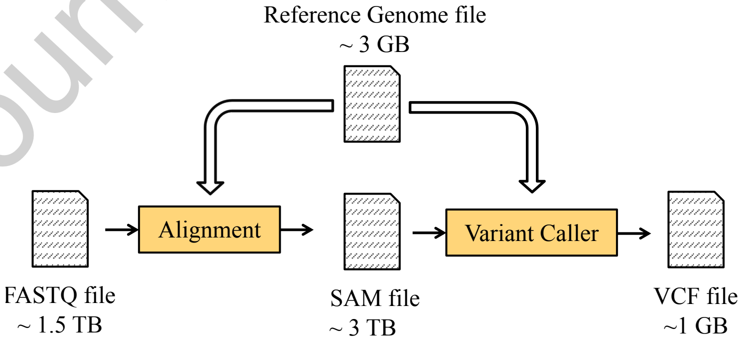

Figure 2: Typical pipeline of genome sequencing with corresponding generated files at each step.

The final step of the analysis pipeline is the so-called *variant calling* or *caller*, an algorithm that given the nucleobase of the reads, their quality scores and the mapping information, decides the discrepancies or *variants* between the original genome and the reference sequence. Typically, the variants can be either a single nucleotide variation (e.g., from an A in the reference genome

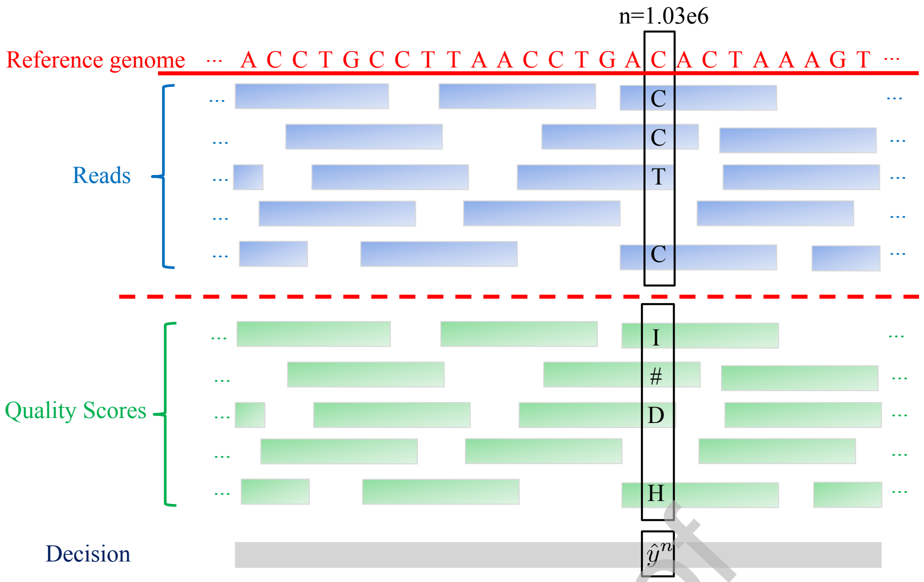

Figure 3: Sequence of reads allocated in the reference genome and corresponding Q-scores.

to a C in the original genome) or INDELS, that stands for insertions or deletions. The set of variants, together with some extra information such as the quality of the variant calling, are stored in a VCF file [29], which is in the order of one gigabyte for human genomes with 3 million variants on average. The most widely used variant calling algorithm is the Genome Analysis Toolkit (GATK) software package.

3. Variant Calling Data Setup

The variant calling step can be formulated as an unsupervised ensemble classification problem in which learners are defined using the Q-scores. After the alignment, the reads and the corresponding Q-score sequences are allocated with respect to the reference sequence as represented in Fig.3. Quality scores indicate the level of confidence of each nucleobase of the read and they are assumed to relate to the error probability P_e of that nucleobase as follows

$$Q = \lceil -10 \log_{10} P_e \rceil \tag{1}$$

In the scale *Phred*+33, Q-scores take values $Q \in \{0, 1, ..., 40\}$ and are stored in a FASTQ file using the ASCII characters of integers $[33 : 73]$, where each integer corresponds to $Q + 33$. Figure 3 shows a toy example where CCTC are the labels or observed data available for the nucleobase at position $n = 1.03 \times 10^6$ from 4 reads with Q-scores equal to $(40, 2, 35, 39)$ since the ASCII characters (I, #, D, H) correspond to the integers $(73, 35, 68, 72)$, respectively.

The mathematical relation between quality scores and probability of error in Eq. (1) must be handled wisely, and it basically reveals that the higher the Q-score, the lower the P_e of the nucleobase of that read. Moreover, as Q-scores represent a large fraction of the storage space required by FASTQ and SAM files, they are usually compressed into a reduced number of quality bins that reduce the data storage requirements significantly without affecting the reliability of the sequencing results. A typical binning is the one used by Illumina [30] given in Table I. It is worth to mention that latest Illumina machine uses this scheme and another one with 4 bins only.

Bin (learner)	1	2	3	4	5	6	7
Q-score range	2-9	10-19	20-24	25-29	30-34	35-39	≥ 40

Table I: Q-scores binning proposed by Illumina and used by EMVC.

In our setup, we define one learner for each bin so that all labels with a Q-score in the range of a bin are associated to the corresponding learner. The binning used in this work is the same proposed by Illumina and given in Table I. Note that those labels with a Q-score equal to $\{0, 1\}$ are not considered since due to the high probability of error, as indicated by Eq. (1), no call is produced in these cases. For instance, in the toy example of Fig. 3 at position $n = 1.03 \times 10^6$, learner $m = 1$ (with Q-scores between $2-9$) provides one label equal to C, learner $m = 6$ (with Q-scores between $35-39$) provides labels $\{C, T\}$, and learner $m = 7$ (with Q-score equal to 40) gives a C. The rest of learners do not tag that position.

3.1. Problem formulation

Let's assume that a DNA sample with N base-pairs is sequenced and $M = 7$ learners tag each nucleobase position with none, one or multiple labels that correspond to the set $\{A, C, G, T\}$. Nucleotide positions are indexed by $n \in \{1, \ldots, N\}$ and learners by $m \in \{1, \ldots, M\}$. We further assume that learner m provides P_m^n labels for nucleotide position n, and that the responses of learner m for position n are denoted by the set

$$\mathcal{R}_m^n = \{r_m^n(1), \ldots, r_m^n(P_m^n)\}$$

where $r_m^n(p)$ denotes the p^{th} label that learner m provides for nucleotide position n. Note that different learners may provide a different number of labels for the same nucleotide position n, i.e., in general $P_m^n \neq P_{m'}^n$, and that the same learner m may provide a different number of labels for different nucleotide positions, i.e., in general $P_m^n \neq P_m^{n'}$. Labels are modelled here as discrete random variables (r.v.'s) that take $L = 4$ different values, i.e.,

$$r_m^n(p) \in \{1, 2, 3, 4\}$$

for $p = 1, \ldots, P_m^n$, which correspond to the four different nucleotides $\{A, C, G, T\}$, respectively. For convenience, let's denote the number of times learner m tags position n with label l by $s_m^n(l)$, so that $\sum_{l=1}^{L} s_m^n(l) = P_m^n$.

Sequencing basically consists in deciding the nucleobase for each position of the DNA sample at hand. We model these nucleobases as latent or hidden r.v.'s denoted by $\{y^n\}_{n=1}^N$. In haploid cells, with only one copy of each chromosome, at each position n we might have four different values which are $\{A, C, G, T\}$. However, in diploid cells we have to decide a combination of two nucleobases for each position n, one for each of the two copies. Therefore, in diploid cells , y^n belongs to $K = 10$ different classes denoted by

$$y^n \in \{1, 2, \ldots, 10\} \tag{2}$$

which one by one correspond to the pairs $\{AA, CC, GG, TT, AC, AG, AT, CG, CT, GT\}$, which are all available unordered combinations of two nucleobases. Thus, in the toy example in Fig. 3, the variant calling algorithm should decide the estimate $\hat{y}^n$ among K possible values ($K = 10$ for diploid cells and $K = 4$ for haploid cells) for position $n = 1.03 \times 10^6$, given that learner $m = 1$ provides label C, learner $m = 6$ provides labels $\{C, T\}$, and learner $m = 7$ gives a C, and the rest of learners do not tag that position.

For convenience, and before presenting the EM-based algorithm as a variant calling algorithm, let's denote the set of hidden r.v.'s by $\mathcal{Y} = \{y^n; n = 1, \ldots, N\}$; and the tags of learner m by $\mathcal{R}_m = \cup_{n=1}^N \mathcal{R}_m^n$, the tags given at position n by all learners by $\mathcal{R}^n = \cup_{m=1}^M \mathcal{R}_m^n$, and the set of all labels by $\mathcal{R} = \cup_{n=1}^N \mathcal{R}^n = \cup_{m=1}^M \mathcal{R}_m$.

4. EM-based Variant Calling

At this point, we are ready to formulate the variant calling step as an unsupervised multiple-class ensemble classification problem as follows. Given the labels for all nucleobase positions, i.e., $\{\mathcal{R}^n = \cup_{m=1}^M \mathcal{R}_m^n; n = 1, \ldots, N\}$ grouped into $M = 7$ learners as explained in Section 3, decide the class of $\{y^n; n = 1, \ldots, N\}$ out of $K = 10$ possible classes given by Eq. (2). The problem is solved using an EM-based iterative approach followed by a final Maximum A Posteriori (MAP) decision. For that, we regard $\mathcal{R}$ as the *incomplete* observation set; $\{\mathcal{R}, \mathcal{Y}\}$ as the *complete* observation set; and θ as the set of parameters to estimate. Initialized with $\hat{\theta}^0$, at iteration $t + 1$, with $t \geq 0$, the general formulation of the EM algorithm is as follows.

S1) *E-step:* given an estimate $\hat{\theta}^t$, compute the conditional expectation of the log-likelihood function

$$Q(\tilde{\theta}; \hat{\theta}^t) \coloneqq \mathbb{E}_{\mathcal{Y}}\{\log f(\mathcal{R}, \mathcal{Y}; \tilde{\theta}) \mid \hat{\theta}^t, \mathcal{R}\}, \tag{3}$$

where $\tilde{\theta}$ denotes a 'trial' value of θ.

S2) *M-step:* obtain the estimate for the next iteration as

$$\hat{\theta}^{t+1} = \arg\max_{\tilde{\theta}} Q(\tilde{\theta}; \hat{\theta}^t). \tag{4}$$

In our setup, the parameters to be estimated are $\theta = \{\{\mathbf{\Gamma}_m\}_{m=1}^M; \{\pi_k\}_{k=1}^K\}$, where $\mathbf{\Gamma}_m \in \mathbb{R}^{L\times K}$ is the confusion matrix of learner m, and π_k is the *a priori* probability of class k. The entry at row i and column j of the confusion matrix of learner m is equal to

$$\mathbf{\Gamma}_m(i,j) = \Pr\left(r_m = i | y = j\right), \tag{5}$$

for $i = 1, \ldots, L$ and $j = 1, \ldots, K$; where $\Pr(\cdot)$ denotes probability and r_m refers to a generic label of learner m. It is therefore assumed that the confusion matrix entries are independent of the nucleotide position n. Interestingly note that, unlike other EM-based algorithms for ensemble learning, in our case the cardinality of the observation set $L = 4$ is different from the number of classes $K = 10$. The a priori probability of the K classes are equal to $\pi_k := \Pr(y^n = k)$ and are assumed to be independent of the nucleotide position n.

4.1. Expectation step

At this point we want to obtain an expression for $Q(\tilde{\theta}; \hat{\theta}^t)$ in Eq. (3). First, note that assuming the r.v.'s associated to different nucleobase positions n are independent, the log-likelihood function is equal to

$$\log f(\mathcal{R}, \mathcal{Y}; \tilde{\theta}) = \log\left(\prod_{n=1}^N f(\mathcal{R}^n, y^n; \tilde{\theta})\right) = \sum_{n=1}^N \log f(\mathcal{R}_1^n, \ldots, \mathcal{R}_M^n, y^n; \tilde{\theta}), \tag{6}$$

where in the second equality we use $\mathcal{R}^n = \cup_{m=1}^M \mathcal{R}_m$. The conditional expected value of Eq. (6) required in Eq. (3) can be obtained as follows

$$\mathbb{E}_{\mathcal{Y}}\{\log f(\mathcal{R}, \mathcal{Y}; \tilde{\theta}) | \hat{\theta}^t, \mathcal{R}\} = \sum_{n=1}^N \sum_{k=1}^K \log f(\mathcal{R}_1^n, \ldots, \mathcal{R}_M^n, y^n = k; \tilde{\theta}) \cdot \Pr(y^n = k | \hat{\theta}^t, \mathcal{R}) \tag{7}$$

Let's denote the conditional *a posteriori* probabilities of the classes by

$$\alpha_{n,k}^t = \Pr(y^n = k | \hat{\theta}^t, \mathcal{R}) = \Pr(y^n = k | \hat{\theta}^t, \mathcal{R}^n) \tag{8}$$

for $n = 1, \ldots, N$ and $k = 1, \ldots, K$, where in the second equality we assume that $\alpha_{n,k}^t$ only depends on the labels for position n. Then, substituting Eq. (8) into Eq. (7), the log-likelihood function becomes

$$Q(\tilde{\theta}; \hat{\theta}^t) = \sum_{n=1}^N \sum_{k=1}^K \alpha_{n,k}^t \big(\log f(\mathcal{R}_1^n, \ldots, \mathcal{R}_M^n | y^n = k; \tilde{\theta}) + \log \Pr(y^n = k; \tilde{\theta})\big)$$

Assuming learners are conditional independent among them and that $\tilde{\theta} = \{\{\tilde{\mathbf{\Gamma}}_m\}_{m=1}^M; \{\tilde{\pi}_k\}_{k=1}^K\}$, we further obtain

$$Q(\tilde{\theta}; \hat{\theta}^t) = \sum_{n=1}^{N} \sum_{k=1}^{K} \alpha_{n,k}^t \left(\log \tilde{\pi}_k + \sum_{m=1}^{M} \log f(\mathcal{R}_m^n | y^n = k; \tilde{\mathbf{\Gamma}}_m) \right). \tag{9}$$

Indeed, the likelihood function of $\mathcal{R}_m^n$, i.e., the labels provided by learner m for position n, given $y^n = k$ is the correct value is equal to

$$f(\mathcal{R}_m^n | y^n = k; \tilde{\mathbf{\Gamma}}_m) = \prod_{l=1}^{L} \left(\tilde{\mathbf{\Gamma}}_m(l,k) \right)^{s_m^n(l)} \tag{10}$$

Then, substituting Eq. (10) into Eq. (9) we get

$$Q(\tilde{\theta}; \hat{\theta}^t) = \sum_{n=1}^{N} \sum_{k=1}^{K} \alpha_{n,k}^t \left(\log \tilde{\pi}_k + \sum_{m=1}^{M} \sum_{l=1}^{L} s_m^n(l) \log \tilde{\mathbf{\Gamma}}_m(l,k) \right). \tag{11}$$

At iteration t, the *E-step* merely needs to calculate the set of a posteriori probabilities $\{\alpha_{n,k}^t; n = 1, \ldots, N \text{ and } k = 1, \ldots, K\}$. Using the Bayes' theorem, these probabilities can be computed as follows:

$$\begin{aligned} \alpha_{n,k}^t &= \frac{\Pr(y^n = k, \mathcal{R}^n; \hat{\theta}^t)}{\Pr(\mathcal{R}^n; \hat{\theta}^t)} \\ &= \frac{\Pr(\mathcal{R}^n | y^n = k; \hat{\theta}^t) \Pr(y^n = k; \hat{\theta}^t)}{\sum_{k'=1}^{K} \Pr(\mathcal{R}^n | y^n = k'; \hat{\theta}^t) \Pr(y^n = k'; \hat{\theta}^t)} \\ &= \frac{\hat{\pi}_k^t \prod_{m=1}^{M} \Pr(\mathcal{R}_m^n | y^n = k; \hat{\theta}^t)}{\sum_{k'=1}^{K} \hat{\pi}_{k'}^t \prod_{m=1}^{M} \Pr(\mathcal{R}_m^n | y^n = k'; \hat{\theta}^t)}, \end{aligned} \tag{12}$$

where in the last equality we use the property of conditional independence among learners. Recall that $\Pr(\mathcal{R}_m^n | y^n = k; \hat{\theta}^t)$ in Eq. (12) can be computed using Eq. (10).

4.2. Maximization Step

Parameters $\{\tilde{\pi}_k\}_{k=1}^K$ and $\{\tilde{\Gamma}_m\}_{m=1}^M$ can be estimated separately as they are decoupled in Eq. (11). At the *M-step*, the set of M confusion matrices are updated solving the following optimization problem

$$\begin{aligned} \{\hat{\mathbf{\Gamma}}_m^{t+1}\}_{m=1}^M = &\arg \max_{\forall \{\tilde{\mathbf{\Gamma}}_m\}_{m=1}^M} Q\left(\{\tilde{\mathbf{\Gamma}}_m\}_{m=1}^M, \{\hat{\mathbf{\Gamma}}_m^t\}_{m=1}^M \right) \\ &s.t. \quad \mathbf{1}^T \tilde{\mathbf{\Gamma}}_m = \mathbf{1}^T \qquad \forall m \end{aligned} \tag{13}$$

where $\mathbf{1}$ is an all-ones vector of dimension L. Imposing the constraint using a Lagrange multiplier λ, each confusion matrix can be easily obtained solving

$$\hat{\mathbf{\Gamma}}_m^{t+1} = \arg \max_{\forall \tilde{\mathbf{\Gamma}}_m \in \mathbb{R}^{L \times K}} \sum_{n=1}^{N} \sum_{k=1}^{K} \alpha_{n,k}^t \sum_{l=1}^{L} s_m^n(l) \log \tilde{\mathbf{\Gamma}}_m(l,k) - \lambda \left(\sum_{l'=1}^{L} \tilde{\mathbf{\Gamma}}_m(l',k) - 1 \right), \tag{14}$$

which readily results in

$$\hat{\mathbf{\Gamma}}_m^{t+1}(l,k) = \frac{\sum_n \alpha_{n,k}^t s_m^n(l)}{\sum_{l'=1}^{L} \sum_{n=1}^{N} \alpha_{n,k}^t s_m^n(l')} \tag{15}$$

for $l = 1, \ldots, L$, $k = 1, \ldots, K$ and $m = 1, \ldots, M$.

The set of K a priori probabilities are updated solving the following optimization problem

$$\{\hat{\pi}_k^{t+1}\}_{k=1}^K = \arg \max_{\forall \{\tilde{\pi}_k\}_{k=1}^K} Q\left(\{\tilde{\pi}_k\}_{k=1}^K, \{\hat{\pi}_k^t\}_{k=1}^K\right)$$
$$s.t. \quad \sum_{k=1}^{K} \tilde{\pi}_k = 1 \tag{16}$$

Again, imposing the constraint using a Lagrange multiplier, the estimate at the i^{th} iteration is equal to

$$\hat{\pi}_k^{t+1} = \frac{\sum_{n=1}^{N} \alpha_{n,k}^t}{N}, \tag{17}$$

for $n = 1, \ldots, N$ and $k = 1, \ldots, K$.

4.3. MAP decision

In a nutshell, after proper initialization of $\hat{\theta}^0 = \{\{\hat{\mathbf{\Gamma}}_m^0\}_{m=1}^M; \{\hat{\pi}_k^0\}_{k=1}^K\}$, the EM algorithm iteratively solves Eq. (12), Eq. (15) and Eq. (17) until convergence of $Q(\tilde{\theta}; \hat{\theta}^t)$, upon which a final estimate of $\{\alpha_{n,k}^f\}$, $\{\tilde{\mathbf{\Gamma}}_m^f\}$ and $\{\tilde{\pi}_k^f\}$ is obtained. Finally, a class decision is taken with a MAP rule that uses these final estimates. That is, at each position n we decide the class of y^n as

$$\hat{y}^n = \arg\max_{k=1,\ldots,K} \alpha_{n,k}^f, \tag{18}$$

which readily maps the pair of nucleotides $\{AA, CC, GG, TT, AC, AG, AT, CG, CT, GT\}$. The presented EM-based Variant Calling (EMVC) algorithm is summarized in Algorithm 1 and represents the meta-learner classifier of the formulated ensemble learning problem.

5. Experimental results

The performance of the EMVC algorithm is assessed by means of experiments using real data available for chromosome 20 of one particular human individual denoted with the name NA12878, which has been thoroughly characterized and for which a set of high-confidence variants, i.e., ground truth, exists [27]. Note that sequencing data from this individual has been extensively used for comparative analyses in the past. For example, in [26], where the effect of lossy compression of quality scores on variant calling is analyzed, or in [20], where best practices for variant calling in the clinical setting are proposed.

Algorithm 1 EM Variant Calling (EMVC)

```
Input: $\{\mathcal{R}_m^n; m = 1, \ldots, M,$ and $n = 1, \cdots, N\}$
Output: $\{\hat{y}^n\}_{n=1}^N$, $\hat{\theta}^f = \{\{\hat{\mathbf{\Gamma}}_m^f\}_{m=1}^M; \{\hat{\pi}_k^f\}_{k=1}^K\}$
procedure
    $\hat{\theta}^0 = \{\{\hat{\mathbf{\Gamma}}_m^0\}_{m=1}^M; \{\hat{\pi}_k^0\}_{k=1}^K\}$ and $t \leftarrow 0$    ▷ Initialization
    repeat
        Compute $\alpha_{n,k}^t; n = 1, \ldots, N$ and $k = 1, \ldots, K$ as in Eq. (12)    ▷ E-Step
        Compute $\hat{\mathbf{\Gamma}}_m^{t+1}; m = 1, \ldots, M$ as in Eq. (15)    ▷ M-Step
        Compute $\hat{\pi}_k^{t+1}; k = 1, \ldots, K$ as in Eq. (17)    ▷ M-Step
    until Convergence of $Q(\tilde{\theta}; \hat{\theta}^t)$
    $\alpha_{n,k}^f = \alpha_{n,k}^t; n = 1, \ldots, N$ and $k = 1, \ldots, K$
    $\hat{y}^n = \arg\max_{k=1,\ldots,K} \alpha_{n,k}^f$ as in Eq. (18)
end procedure
```

Chromosome 20 has 60 million of base-pairs and in the experiments we use the reference genome of chromosome 20, and a SAM file with a total of $8,7 \times 10^6$ reads that fully span chromosome 20 of NA12878. This SAM file has a coverage within the range of $10 - 30$ labels per nucleotide position. The EMVC performance is evaluated using the true genome sequencing or ground truth of chromosome 20 of NA12878. Results of the EMVC algorithm are compared to those provided by GATK, HTSLIB, and Platypus in terms of precision and sensitivity. Precision denoted by P and sensitiviy (or recall) denoted by S are computed as

$$P = \frac{\#TP}{\#TP + \#FP} \tag{19}$$

$$S = \frac{\#TP}{\#RE} \tag{20}$$

where TP, FP and RE stand for true positives, false positives and relevant elements, respectively. In our setup the REs are the nucleotide positions of the ground truth where at least one of the couple of nucleotides is different from the nucleotide of the reference genome. These positions are called *variants*[5]. Therefore, positives refer to nucleotide positions where the variant calling algorithm decides there is a variant. The TPs of a variant calling algorithm are given by the number of nucleotide positions where the algorithm correctly identifies a variant, i.e., at least one of the couple of nucleotides of the decided class is different from the reference genome and the decided class coincides with the one of the ground truth. The FPs of a variant calling algorithm are given by the number of nucleotide positions where the algorithm incorrectly

[5] Variants refer to differences in the nucleotides with respect to the reference genome but also to insertions and deletions, named INDELS. The EMVC algorithm detects changes of nucleotides but not INDELS, yet.

identifies a variant, i.e., at least one of the couple of nucleotides of the decided class is different from the reference genome but either this position is not a true variant (i.e., is not a relevant element) or the decided class is different from the couple of nucleotides given by the ground truth.

This section is organized as follows. First, details regarding initialization and convergence of the EMVC algorithm are presented. Then, the performance of the EMVC is evaluated for different values of N and compared to state-of-the-art variant callers using real data.

5.1. EMVC initialization

The cost function of the EM algorithm, given by (11) in the problem at hand, has multiple local minima and a wise initialization of the parameters is required to achieve a competitive performance. In our setup, classes $\{AA, CC, GG, TT\}$ correspond to a position where both the chromosome of the mother and the father have the same nucleotide than the reference genome. Indeed, these classes have a much higher probability of occurrence than the rest because only around 0.1% of the DNA of human beings differ from the reference genome. In view of this, in our experiments the a priori probabilities are initialized as follows

$$\tilde{\pi}_k^0 = \begin{cases} 0.205 & k = 1, \ldots, 4 \\ 0.03 & k = 5, \ldots, 10 \end{cases} \tag{21}$$

where $\sum_{k=1}^{K} \hat{\pi}_k^0 = 1$ must be guaranteed. With regards to the confusion matrices of the learners, all of them take the same initial value given by

$$\hat{\mathbf{\Gamma}}_m^0 = \begin{bmatrix} 0.4 & 0.2 & 0.2 & 0.2 & 0.3 & 0.3 & 0.3 & 0.2 & 0.2 & 0.2 \\ 0.2 & 0.4 & 0.2 & 0.2 & 0.3 & 0.2 & 0.2 & 0.3 & 0.3 & 0.2 \\ 0.2 & 0.2 & 0.4 & 0.2 & 0.2 & 0.3 & 0.2 & 0.3 & 0.2 & 0.3 \\ 0.2 & 0.2 & 0.2 & 0.4 & 0.2 & 0.2 & 0.3 & 0.2 & 0.3 & 0.3 \end{bmatrix} \tag{22}$$

$\forall m = 1, \ldots, M$, where the probability of observing a nucleotide conditioned on a given couple increases if the observed nucleotide belongs to the couple. For instance, the initial probability that a read of learner m is A conditioned that the true class is AA is given by $\mathbf{\Gamma}_m^0(1, 1) = 0.4$; the initial probability that a read is A conditioned that the true class is AG is $\mathbf{\Gamma}_m^0(1, 6) = 0.3$; and the initial probability that a read is A conditioned that the true class is GG is set to $\mathbf{\Gamma}_m^0(1, 3) = 0.2$. Even though the relation between the Q-score and the probability of error in (1) might be used to initialize the confusion matrices of the different learners, we opt here for initializing all matrices the same and let the EMVC algorithm to learn the confusion matrices from the data.

5.2. Convergence of EMVC

In this section we analyze the first 3×10^6 reads of the SAM file of NA12878 and run the EM algorithm. At this point, it is important to remark that the EMVC algorithm only uses the nucleobase positions where at least two reads provide different labels. That is, if for instance at a given nucleobase position all reads provide a label equal to A, the decision taken by the classifier is going to be AA, and the reads of this position are not used by the EMVC algorithm. However, these nucleotide positions where all reads provide the same label are obviously considered in the computation of the precision and sensitivity. Therefore, after analyzing the first 3×10^6 reads of the SAM file that reach up to the nucleotide at position 20.627.099 of the chromosome 20 of NA12878, only the labels of $N = 928.406$ nucleotide positions are used to run the EMVC algorithm. In this particular case we have $M = 6$ learners that correspond to the first 6 bins in Table I and the number of iterations of EMVC is fixed to 50.

Regarding the convergence of the EMVC algorithm, Fig. 4 plots the evolution of the a priori probabilities per EM iteration. Clearly, $\{\hat{\pi}_k^t\}_{k=1}^K$ converge and, interestingly, the final values $\hat{\pi}^f = \{0.2997, 0.1864, 0.1870, 0.3027, 0.0021, 0.0082, 0.0019, 0.0020, 0.0079, 0.0021\}$ are very similar to $\{0.3028, 0.1839, 0.1859, 0.3051, 0.0018, 0.0078, 0.0015, 0.0018, 0.0076, 0.0018\}$, which are the class probabilities provided by GATK.

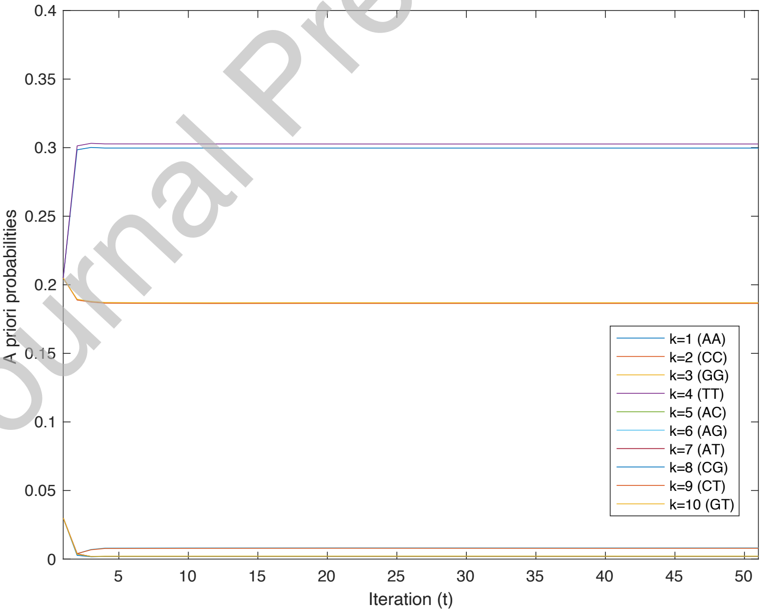

Figure 4: Evolution of the a priori probabilities $\{\hat{\pi}_k^t\}_{k=1}^{K=10}$ over the 50 iterations.

With regards to the convergence of the confusion matrices, Fig. 5 and Fig. 6 show the evolution of the entries of the confusion matrix of learner $m = 1$ and $m = 6$, respectively, i.e., $\{\hat{\mathbf{\Gamma}}_1^t(l,k), \hat{\mathbf{\Gamma}}_6^t(l,k); l = 1, \ldots, 4 \text{ and } k = 1, \ldots, 10\}$. Clearly learner $m = 1$, who is the less reliable with lower Q-score values, is the one that needs more iterations to converge.

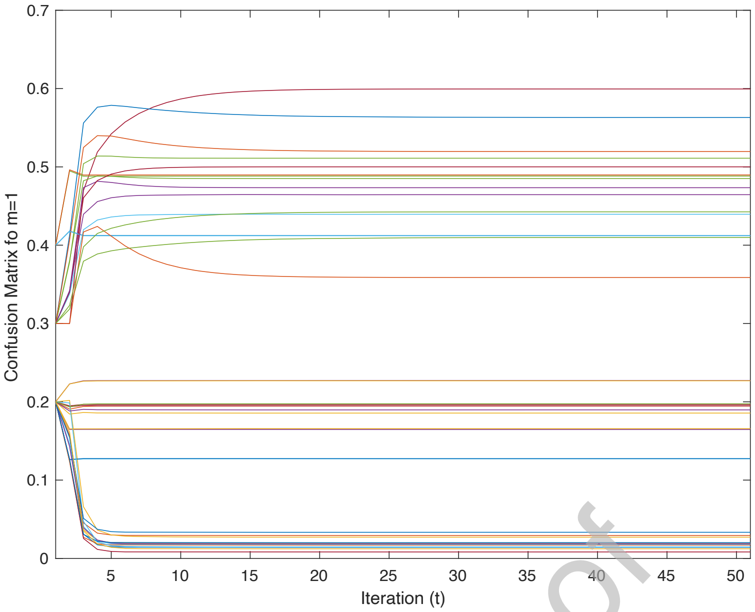

Figure 5: Evolution of the entries of $\hat{\mathbf{\Gamma}}_1^t$ the confusion matrix of learner $m = 1$ per EM iteration.

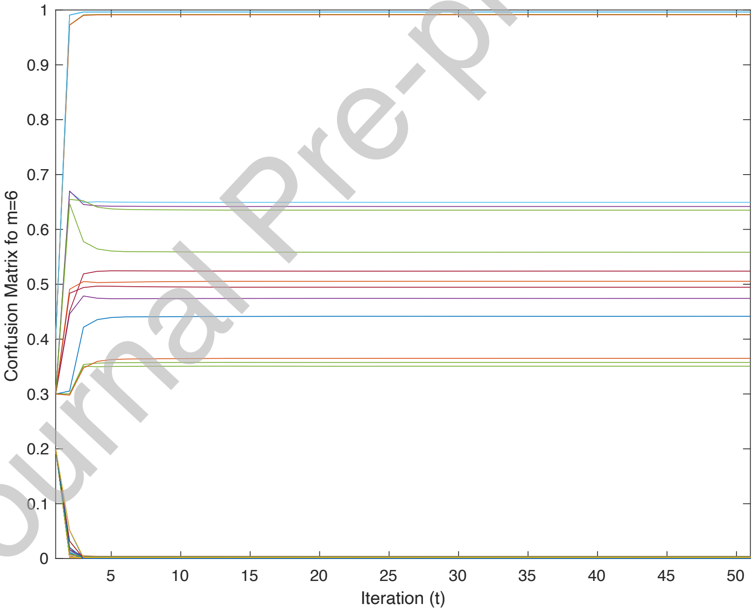

Figure 6: Evolution of the entries of $\hat{\mathbf{\Gamma}}_6^t$ the confusion matrix of learner $m = 6$ per EM iteration.

Eq. (23) and Eq. (24) provide the final values $\hat{\mathbf{\Gamma}}_1^f$ and $\hat{\mathbf{\Gamma}}_6^f$, respectively. As expected, the confusion matrix $\hat{\mathbf{\Gamma}}_6^f$ of learner $m = 6$ estimated by the EMVC algorithm corresponds to a reliable learner, whereas the estimated $\hat{\mathbf{\Gamma}}_1^f$ corresponds to a less reliable learner since $m = 1$ gathers reads with the nine lowest Q-score values. For instance, after convergence of the EMVC algorithm, the probability that a read of learner $m = 6$ is A conditioned that the true class 255 is AA becomes by $\mathbf{\Gamma}_6^f(1, 1) = 0.9964$ for learner $m = 6$ but only $\mathbf{\Gamma}_6^f(1, 1) = 0.4122$ for learner

$m = 1$; or the probability that a read is different from A or G conditioned that the true class is AG is almost 0 for learner $m = 6$ but equal to $\mathbf{\Gamma}_1^f(2,6) + \mathbf{\Gamma}_1^f(4,6) = 0.0494$ for learner $m = 1$.

$$\hat{\mathbf{\Gamma}}_1^f = \begin{bmatrix} 0.4122 & 0.1946 & 0.1856 & 0.1646 & 0.4099 & 0.4395 & 0.5995 & 0.0142 & 0.0170 & 0.0198 \\ 0.2271 & 0.4882 & 0.1272 & 0.1965 & 0.5631 & 0.0293 & 0.0271 & 0.4734 & 0.4851 & 0.0180 \\ 0.1949 & 0.1274 & 0.4897 & 0.2268 & 0.0188 & 0.5111 & 0.0146 & 0.5001 & 0.0334 & 0.5196 \\ 0.1657 & 0.1897 & 0.1974 & 0.4121 & 0.0083 & 0.0201 & 0.3587 & 0.0123 & 0.4645 & 0.4426 \end{bmatrix} \tag{23}$$

$$\hat{\mathbf{\Gamma}}_6^f = \begin{bmatrix} 0.9964 & 0.0028 & 0.0041 & 0.0015 & 0.5584 & 0.6493 & 0.4946 & 0.0000 & 0.0008 & 0.0000 \\ 0.0007 & 0.9918 & 0.0014 & 0.0012 & 0.4416 & 0.0000 & 0.0000 & 0.4742 & 0.3574 & 0.0000 \\ 0.0012 & 0.0015 & 0.9912 & 0.0008 & 0.0000 & 0.3507 & 0.0000 & 0.5240 & 0.0000 & 0.3649 \\ 0.0017 & 0.0039 & 0.0033 & 0.9966 & 0.0000 & 0.0000 & 0.5054 & 0.0018 & 0.6418 & 0.6351 \end{bmatrix} \tag{24}$$

5.3. EMVC Performance

In this section, the precision and sensitivity of the EMVC algorithm is computed in different experiments using real data. In all experiments, $M = 7$ learners given by TableI are used; the number of labels is $L = 4$ and the number of classes is $K = 10$ that correspond to $\{A, C, G, T\}$ and $\{AA, CC, GG, TT, AC, AG, AT, CG, CT, GT\}$, respectively. The initial values of the a priori probabilities are given by (21) and the confusion matrices are all initialized using (22). The number of iterations run by EMVC is equal to 50 and the number of nucleotide positions denoted by parameter N depends on the experiment as detailed below.[6] Table II shows precision and sensitivity of the EMVC algorithm for an increasing number of nucleotides of chromosome 20 of individual NA12878 up to the first 3×10^6 reads. For comparative purposes, Table II also includes the precision and sensitivity achieved by the state-of-the-art variant callers, namely, GATK [21, 22][7], HTSLIB [24][8], and Platypus [25]. Results for these pipelines are obtained using the full chromosome, i.e., the complete SAM file that includes $8,7 \times 10^6$ reads.

For the sake of clarity, take for instance the row 2M (resp.[9] 100K) in Table II. The value $2M$ (resp. $100K$) means that the first 2×10^6 (resp. 100×10^3) reads of the SAM file of NA12878 are considered in the procedure, being the position of the first nucleotide 59.988 and the position

[6] Implementation is done in MATLAB and Python, and it is available upon request to the authors.

[7] Available at https://gatk.broadinstitute.org/hc and https://github.com/broadinstitute/gatk/releases (last Release in Feb. 2022)

[8] Available at http://www.htslib.org and https://github.com/samtools/samtools/releases/ (last release in Feb.2022)

[9] *resp.* is an abbreviated form of respectively.

of the last one 13.798.418 (resp. 750.036). Indeed, for the first 59.987 positions there are no reads available and this initial part cannot be sequenced, which is usual in other SAM files. The total number of nucleotide positions to be sequenced is 13.738.431 = 13.798.418 − 59.987 (resp. 690.049 = 750.036 − 59.987) but only $N = 631.785$ (resp. 40.119) positions show discrepancy in the labels of the reads. Hence, these are the nucleotide positions for which EMVC takes a decision about the class they belong to, i.e., $\{AA, CC, GG, TT, AC, AG, AT, CG, CT, GT\}$. In the rest of nucleotide positions, i.e., a total of 13.738.431 − 631.785 = 13.106.646 (resp. 690.049 − 40.119 = 649.930) positions, the labels of all reads for these positions are equal and they are not used by the EMVC algorithm since the decision is trivial, i.e., one of the first four classes $\{AA, CC, GG, TT\}$.

# Reads of NA12878	Nucleotide Initial Position	Nucleotide Final Position	# Nucleotides N (EMVC)	P (%)	S (%)
100K	59.988	750.036	40.119	77,01 (EMVC)	96,81 (EMVC)
1M	59.988	6.957.434	373.859	83,23 (EMVC)	97,51 (EMVC)
2M	59.988	13.798.418	631.785	85.29 (EMVC)	97.51 (EMVC)
3M	59.988	20.626.160	928.406	85,03 (EMVC)	97,56 (EMVC)
8,7M (full)				74,31 (GATK)	96,98(GATK)
8,7M (full)				72,88 (HTSLIB)	96,83 (HTSLIB)
8,7M (full)				76,45 (Platypus)	91,90 (Platypus)

Table II: Precision (P) and sensitivity (S) obtained by: EMVC for up to the first 3×10^6 reads of chromosome 20 of NA12878 with a coverage in the range of $10 - 30$, and by GATK [21, 22], HTSLIB [24] and Platypus [25] using the full chromosome 20 of NA12878, i.e., $8,7 \times 10^6$ reads

.

As observed in Table II, the performance of EMVC is superior to the rest of variant callers since it achieves a higher precision and similar sensitivity. Indeed, the precision and sensitivity values of EMVC saturate to 85% and 97%, respectively, using the first 2×10^6 reads and running the EMVC with $N = 631.785$ nucleotide positions. This result means that the EMVC algorithm performs well even without using the full genome, which might help to reduce the computational cost and memory requirements of the sequencing procedure by analysing fragments of the SAM file in parallel.

In the previous experiment, results in Table II are obtained with a single realization of EMVC. Figure 7 and 8 show boxplots of the precision and sensitivity obtained by EMVC using 10 realizations for three different cases {100K, 1M, 2M}. The first case, namely 100K, means that each realization is executed using 100×10^3 reads. The data for the different realizations are obtained by splitting the first 1×10^6 reads of the SAM file into 10 files. In the other two

cases, namely 1M and 2M, the EMVC is executed using 1×10^6 and 2×10^6 reads, respectively. The 10 files of data are obtained from the first 3×10^6 reads of the SAM file using a regular data shift or lag. The empirical mean value and standard deviation of the precision (sensitivity)
300 for 100K are $82,03\% \pm 4,33$ ($97,48\% \pm 0,66$); for 1M are $85,85\% \pm 1,52$ ($97,56\% \pm 0,19$); and for 2M $85,96\% \pm 0,40$($97,58\% \pm 0,05$) . It can be observed that the range of precision and sensitivity results achieved by EMVC are significantly reduced when 2×10^6 reads are analyzed, suggesting that this is a convenient file size if the SAM file was processed in parallel.

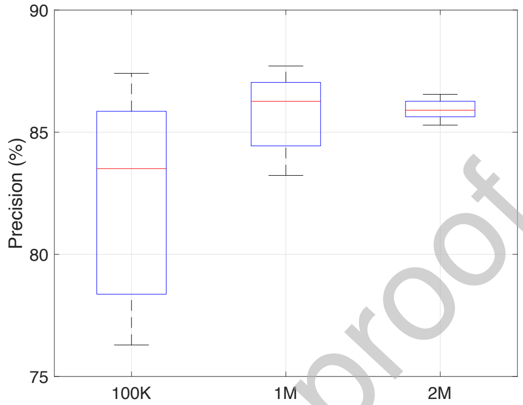

Figure 7: Boxplots of precision obtained by EMVC using 10 realizations with 100×10^3 (100K), 1×10^6 (1M), and 2×10^6 (2M) reads of the SAM file.

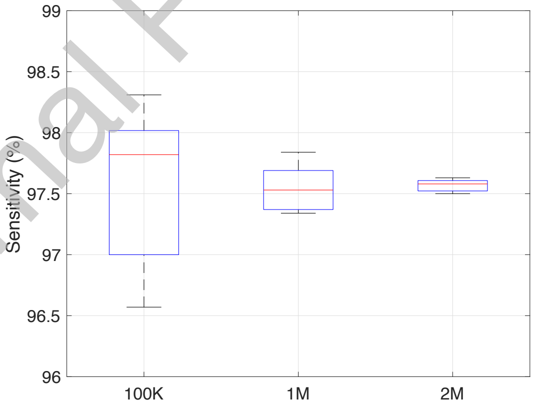

Figure 8: Boxplots of sensitivity obtained by EMVC using 10 realizations with 100×10^3 (100K), 1×10^6 (1M), and 2×10^6 (2M) reads of the SAM file.

Table III shows precision and sensitivity of the EMVC algorithm of different fragments
305 of the SAM file of chromosome 20 of NA12878. The coverage of the SAM file used in these experiments is expected to be in the range of $10-30$ labels of reads per nucleotide position. The SAM file has $8,7 \times 10^6$ reads and it is partitioned into three fragments: one with the first 3×10^6 reads (row 3M in Table III), a second one with the subsequent 2×10^6 reads (row 3M to 5M in Table III), and the final one with the last $3,7 \times 10^6$ reads (row 5M to 8,7M in Table III). Then,

the EMVC algorithm is run for each of these three fragments, and the precision and sensitivity are computed. Note that, for computational limitations, the precision and sensitivity of EMVC in row 8,7M (full*) are calculated gathering the information obtained with the three fragments separately.

# Reads of NA12878	Nucleotide Initial Position	Nucleotide Final Position	# Nucleotides N (EMVC)	P (%)	S (%)
3M	59.988	20.626.160	928.406	85,03 (EMVC)	97,56 (EMVC)
3M to 5M	20.626.060	36.929.891	785.153	35,70 (EMVC)	96,78 (EMVC)
5M to 8,7M	36.929.793	62.965.486	1.407.042	76,70 (EMVC)	97,26 (EMVC)
8,7M (full*)				66,63 (EMVC)	97,27 (EMVC)
8,7M (full)				74,31 (GATK)	97,5 (GATK)
8,7M (full)				72,88 (HTSLIB)	96,83 (HTSLIB)
8,7M (full)				76,45 (Platypus)	91,90 (Platypus)

Table III: Precision (P) and sensitivity (S) obtained by EMVC for the full chromosome 20 of NA12878 split into three fragments, and obtained by GATK [21, 22], HTSLIB [24] and Platypus [25] using the full chromosome 20. Coverage of 20.

As it can be observed in Table III, the EMVC performance for the fragment 3M to 5M decreases significantly with a precision of $35,70\%$, much lower than GATK. After an inspection of the dataset, we observe that between the nucleotide positions 25×10^6 and 30×10^6 there is a region with abnormal values of coverage as high as 500. Typically, this occurs in the presence of groups of reads that are a copy of other ones within the same SAM file, which are called *spurious*. This region makes the EMVC algorithm obtain a global sensitivity of $97,27\%$ and a precision of $66,63\%$ for the full chromosome, which is lower than the rest of variant callers, e.g. GATK is $74,31\%$. Indeed, GATK has a mechanism to deal with these regions, that our method does not include. Still, even without this filtering post-processing step, EMVC achieves competitive figures of precision and sensitivity compared to the state-of-the-art variant callers GATK, HTSLIB, and Platypus.

With regards to the computational cost of the EMVC algorithm, the execution time of EMVC using MATLAB is in the range of $0,18 - 0,20$ seconds per 1.000 processed nucleotides and per iteration. Thus, for instance, the execution time of the fragment 3M of Table III with $N = 928.406$ and 50 iterations costs around 2 hours and a half. This computational load is expected to be significantly reduced if other programming languages that handle big data files more efficiently were used. Indeed, this is proposed as future work.

6. Conclusions

The variant calling step in next generation sequencing technologies for DNA sequencing is presented here as an unsupervised classification task, where for each nucleotide position of the DNA in diploid cells a decision among the classes $\{AA, CC, GG, TT, AC, AG, AT, CG, CT, GT\}$ must be taken given several labels among the set $\{A, C, G, T\}$ that are provided by the reads. In this paper, we solve the variant calling step as an ensemble classification problem by arranging the read labels into groups according to their quality scores, so that labels of the same group show a similar reliability. A variant caller algorithm based on the EM algorithm is proposed, and experimental results prove that the proposed algorithm is competitive in terms of precision and sensitivity to other state-of-the-art variant callers as GATK, HTSLIB and Platypus. In particular, EMVC obtains in some cases the same sensitivity but improved precision, which corresponds to fewer incorrectly called variants, and can lead to better clinical decisions. The proposed variant caller bins the quality scores, fact that further supports the idea shown in previous studies that the full range of quality scores is not needed to obtain a high quality set of variants. This is interesting to reduce the memory requirements of the SAM file. To the best of our knowledge, this work presents the first variant caller formulated as an ensemble classifier that shows a competitive performance compared to state-of-the-art methods. Moreover, we do believe our work paves the way to the research community to apply other existing ensemble classification algorithms to solve the variant calling problem. Future work includes the development of an improved EMVC algorithm capable of detecting insertions and deletions that typically occur in the DNA; to develop the full pipeline code in more efficient open-source code for the research community; and devise filtering mechanisms to improve the performance of EMVC in DNA regions with spurious reads.

7. Acknowledgements

This work has been funded by the Agencia Estatal de Investigación, Ministerio de Ciencia, Innovación y Universidades of the Spanish Government and ERDF funds (PID2019-104958RB-C41, RED2018-102668-T); and by Gipuzkoa Fellows grant and Ramon y Cajal grant.

References

[1] O. Sagi, L. Rokach, Ensemble learning: A survey, Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 8 (4) (2018) e1249.

[2] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, M. Woźniak, Ensemble learning for data stream analysis: A survey, Information Fusion 37 (2017) 132–156.

[3] V. C. Raykar, S. Yu, L. H. Zhao, G. H. Valadez, C. Florin, L. Bogoni, L. Moy, Learning from crowds, Journal of Machine Learning Research 11 (Apr.) (2010) 1297–1322.

[4] M. Micsinai, F. Parisi, F. Strino, P. Asp, B. D. Dynlacht, Y. Kluger, Picking chip-seq peak detectors for analyzing chromatin modification experiments, Nucleic acids research 40 (9) (2012) e70–e70.

[5] A. Pagès-Zamora, M. Cabrera-Bean, C. Díaz-Vilor, Unsupervised online clustering and detection algorithms using crowdsourced data for malaria diagnosis, Pattern Recognition 86 (2019) 209–223.

[6] J. B. Rhim, V. K. Goyal, Distributed hypothesis testing with social learning and symmetric fusion, IEEE Trans. on Signal Processing 62 (23) (2014) 6298–6308.

[7] M. Usman, K. Insoo, Sensor network-based spectrum sensing for cognitive radio network, in: Int. Conf. on Intelligent Systems Engineering (ICISE), IEEE, 2016, pp. 19–25.

[8] A. P. Dawid, A. M. Skene, Maximum likelihood estimation of observer error-rates using the em algorithm, Applied statistics (1979) 20–28.

[9] F. Parisi, F. Strino, B. Nadler, Y. Kluger, Ranking and combining multiple predictors without labeled data, Proc. of the National Academy of Sciences 111 (4) (2014) 1253–1258.

[10] A. Jaffe, B. Nadler, Y. Kluger, Estimating the accuracies of multiple classifiers without labeled data, in: Artificial Intelligence and Statistics, 2015, pp. 407–415.

[11] Y. Zhang, X. Chen, D. Zhou, M. I. Jordan, Spectral methods meet em: A provably optimal algorithm for crowdsourcing, The Journal of Machine Learning Research 17 (1) (2016) 3537–3580.

[12] P. A. Traganitis, A. Pages-Zamora, G. B. Giannakis, Blind multiclass ensemble classification, IEEE Transactions on Signal Processing 66 (18) (2018) 4737–4752.

[13] A. W.-C. Liew, H. Yan, M. Yang, Pattern recognition techniques for the emerging field of bioinformatics: A review, Pattern Recognition 38 (11) (2005) 2055–2073.

[14] T. D. Pham, Spectral distortion measures for biological sequence comparisons and database searching, Pattern Recognition 40 (2) (2007) 516–529.

[15] S. Forêt, S. R. Wilson, C. J. Burden, Empirical distribution of k-word matches in biological sequences, Pattern Recognition 42 (4) (2009) 539–548.

[16] W.-F. Zhang, H. Yan, Exon prediction using empirical mode decomposition and fourier transform of structural profiles of dna sequences, Pattern Recognition 45 (3) (2012) 947–955.

[17] L. Ou-Yang, X.-F. Zhang, H. Yan, Sparse regularized low-rank tensor regression with applications in genomic data analysis, Pattern Recognition 107 (2020) 107516.

[18] J. Cohen, A. Pertsemlidis, I. K. Kotowski, R. Graham, C. K. Garcia, H. H. Hobbs, Low ldl cholesterol in individuals of african descent resulting from frequent nonsense mutations in pcsk9, Nature genetics 37 (2) (2005) 161–165.

[19] T. Walsh, S. Casadei, K. H. Coats, E. Swisher, S. M. Stray, J. Higgins, K. C. Roach, J. Mandell, M. K. Lee, S. Ciernikova, et al., Spectrum of mutations in brca1, brca2, chek2, and tp53 in families at high risk of breast cancer, Jama 295 (12) (2006) 1379–1388.

[20] D. C. Koboldt, Best practices for variant calling in clinical sequencing, Genome Medicine 12 (1) (2020) 1–13.

[21] A. McKenna, M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernytsky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, et al., The genome analysis toolkit: a mapreduce framework for analyzing next-generation dna sequencing data, Genome research 20 (9) (2010) 1297–1303.

[22] M. A. DePristo, E. Banks, R. Poplin, K. V. Garimella, J. R. Maguire, C. Hartl, A. A. Philippakis, G. Del Angel, M. A. Rivas, M. Hanna, et al., A framework for variation discovery and genotyping using next-generation dna sequencing data, Nature genetics 43 (5) (2011) 491–498.

[23] G. A. Van der Auwera, M. O. Carneiro, C. Hartl, R. Poplin, G. Del Angel, A. Levy-Moonshine, T. Jordan, K. Shakir, D. Roazen, J. Thibault, et al., From fastq data to high-confidence variant calls: the genome analysis toolkit best practices pipeline, Current protocols in bioinformatics 43 (1) (2013) 11–10.

[24] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, The sequence alignment/map format and samtools, Bioinformatics 25 (16) (2009) 2078–2079.

[25] A. Rimmer, H. Phan, I. Mathieson, G. Lunter, G. McVean, Platypus: A haplotype-based variant caller for next generation sequence data (2013).

[26] I. Ochoa, M. Hernaez, R. Goldfeder, T. Weissman, E. Ashley, Effect of lossy compression of quality scores on variant calling, Briefings in bioinformatics 18 (2) (2017) 183–194.

[27] J. M. Zook, J. McDaniel, N. D. Olson, J. Wagner, H. Parikh, H. Heaton, S. A. Irvine, L. Trigg, R. Truty, C. Y. McLean, et al., An open resource for accurately benchmarking small variant and reference calls, Nature biotechnology 37 (5) (2019) 561–566.

[28] I. Ochoa-Alvarez, Genomic data compression and processing: Theory, models, algorithms, and experiments, Ph.D. thesis, Stanford University (2016).

[29] P. Danecek, A. Auton, G. Abecasis, C. A. Albers, E. Banks, M. A. DePristo, R. E. Handsaker, G. Lunter, G. T. Marth, S. T. Sherry, et al., The variant call format and vcftools, Bioinformatics 27 (15) (2011) 2156–2158.

[30] Illumina, Understanding Illumina quality scores, Tech. rep., Technical Note: Informatics (01 2010).

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author Biography

Alba Pagés-Zamora received the MS degree in 1992 and PhD degree in 1996 in Electrical Engineering from the Universitat Politècnica de Catalunya (UPC), Spain. In 2001 became Associate Professor of the Signal Theory and Communications Department at UPC. She has co-authored one patent, 2 book chapters, 19 papers in journals and 60 papers in conferences, and has been principal investigator of 5 national and European research projects.

Idoia Ochoa is an Associate Professor of the Electrical Engineering Department at the University of Navarra (Spain). Previously she was an Assistant Professor at the University of Illinois. She obtained a Ph.D. from Stanford University. She is recipient of the MIT Innovators under 35 award and a Ramon-y-Cajal grant.