January, 2022.
Yumika Shiba

COMP402 Intermediate Report

Background

Transposable elements (TEs) are mobile DNA elements that make up a large portion of the eukaryotic genome[1]. TEs account for almost half of the human genome[2] and their insertions are sources of mutations[1] and diseases[3].

Although not all TEs of the human lineage are mobile at present day, there are TE families whose members are still mobile. The largest families of mobile TEs are *Alu*, *SVA*, and *L1*, each of which belongs to a class of TE called retrotransposons (Class I) that make copies of themselves and insert them elsewhere[4]. The activity of these mobile elements create insertion polymorphism, and in humans, which are diploid, there can be three possible genotypes in the population for a given locus: homozygous TE present, heterozygous, and homozygous TE absent (Fig 1). Note that detecting insertion polymorphism and genotyping are different: whereas the former is about detecting the presence or absence of TE insertion, the latter, genotyping, is about determining which pair of two alleles (TE present or absent) is present at a given locus.

TE insertion polymorphism has been shown to have functional consequences: for example, *Alu*, *SVA*, and *L1*, can potentially cause diseases by inserting into human genes[4,5]. In addition, some polymorphic TEs can contain regulatory sequences, including a promoter region where transcription factor binding sites can prompt chromatin modification and modulation of nearby gene expression. Furthermore, TEs such as *Alu* can provide alternative splicing sites to existing genes[3]. Therefore, when polymorphic TEs transpose, there can be functional consequences, such as changes in gene expression levels of nearby genes and or changes in chromatin accessibility, which can subsequently lead to phenotypic variation. Given the significance of TE insertion polymorphism, it is crucial to obtain accurate genotypes, for example, to study the effect of the TE insertion on gene regulation or to identify individuals with higher risk of a disease if the presence or absence of a TE in a particular locus is associated with a known, adverse phenotype. To summarize, accurate TE genotyping bears huge implications in genomic medicine[6].

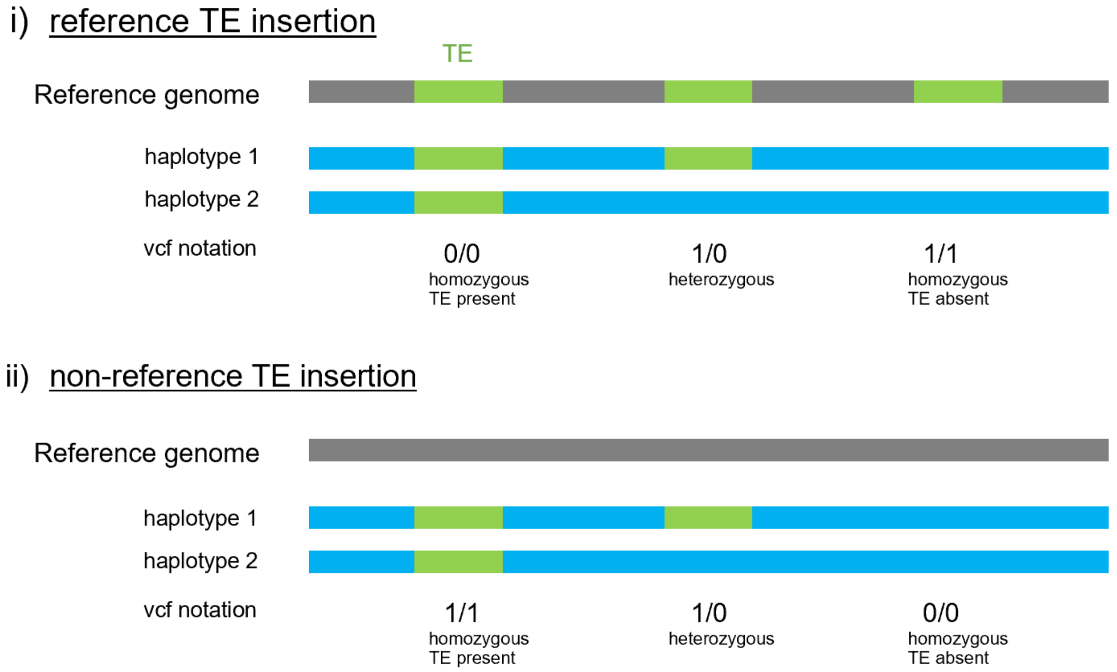


Figure 1. The three possible genotypes for polymorphic TE insertion in humans for i) reference TE insertion (TE present in the reference genome that can be absent in other individuals) ii) non-reference TE insertion (TE absent in the reference genome that can be present in other individuals). The green portions represent TEs. The haplotypes are from the individual/BAM/CRAM to be genotyped. Notice that the meaning of each of 0/0 and 1/1 is reversed between reference and non-reference cases (0/0 meaning homozygous TE present in the reference case but meaning homozygous TE absent in the non-reference case. The reverse holds for 1/1.).

Multiple tools have been developed to genotype polymorphic TE insertions[6,7,8,9]. Among these are xTea[6], which genotypes non-reference TE insertion (TE locus present in the reference human genome that can be absent in other individuals), MELT2 (version 2.1.4 of a program called MELT)[7], which genotypes both non-reference TE insertion and reference TE insertion (TE locus absent in the reference genome that can be present in other individuals), and TypeTE[8], which genotypes non-reference and reference insertions of *Alu*, the most abundant type of mobile TE in humans. xTea is recently published and is the only method that can incorporate both long- and short-read sequencing data, and MELT2 has become one of the most widely used tools for polymorphic TE detection and genotyping TE using short-read sequencing. While xTea and MELT2 use data from read alignment to a reference genome, TypeTE performs re-alignment of short-reads to a reconstituted mini-genome that contain TE present and absent alleles for each locus (Fig 2. See Method.).

While there are many tools that genotype non-reference TE insertions, not many tools exist for genotyping reference TE insertions, which is partly due to the much greater number of non-reference TE insertions present among human genomes[7]. However, reference TE insertions can also affect gene expression; for instance, they have been shown to be an important source of regulatory variants associated with gene expression in Lymphoblastoid cell lines and in iPSCs (induced pluripotent stem cells)[10]. In addition, although tools exist for genotyping reference TE insertions, such as MELT2 and TypeTE, their algorithms still have room for improvement. When using low-coverage 1000 Genome Project (GP) data (~5-7X), the genotype prediction accuracy of MELT2 and TypTE have been shown to be 71.00% and 91.56%, respectively[8]; the 29.00% or 8.44% could mean that some regulatory TEs, potentially important ones, are being mis-labeled, which can prevent researchers from producing accurate results on analyses that rely on the genotypes or to incorrectly inform a person of their disease risk.

To tackle the problem of accurate genotyping, machine learning has been used and shown to improve the genotyping accuracy of single nucleotide polymorphisms (SNPs) and structural variants (SVs)[11,12,13,14]. For example, CNN has been used for genotyping SNPs in DeepVariant[11] and for SVs in DeepSV[12], and hidden markov model (HMM) has been used for genotyping SNPs and SVs in vi-HMM[13]. Furthermore, support vector machine (SVM) has been shown to improve the genotyping accuracy of non-reference TE insertion in xTea[6]. These suggest that, by employing machine learning, it may be possible to improve the accuracy of genotyping reference TE insertion, which is the goal of the project.

The remaining parts of the report starts by a description of TypeREF, a program for genotyping reference TE insertion built by one of the author's supervisors, as well as its benchmarking. This is done before getting into the details of machine learning models because TypeREF is (i) the program in which a machine learning model may be incorporated if it can improve genotyping accuracy and (ii) feature extraction step relies on TypeREF. Then, the preliminary results of implementing machine learning models for genotyping reference *Alu* insertions based on read counts generated by a new version of the program TypeTE called TypeREF will be presented. The program TypeREF currently uses a genotype likelihood framework[15,16] to infer genotype using the count of short-read pairs re-mapped to either presence or absence allele of each TE locus analyzed (Fig 2). As an alternative, logistic regression, SVM, random forest, SVM, and multilayer perceptron (MLP) were employed. Lastly in Discussion, interpretations of the results, limitations, and future steps of the project will be discussed, including other features that may be more informative than read counts.

Materials and Methods

Method Overview

This section begins by describing methods for evaluating the reference TE genotyping accuracy of TypeREF. This is necessary, as TypeREF is the program in which machine learning will be incorporated if the new methods outperforms the current performance of its genotyping algorithm. Thus, it is first necessary to obtain a readout of the baseline performance of TypeREF which can then be used to compare performances of methods involving machine learning. Then, implementation details of the machine learning models, namely logistic regression, random forest, SVM, and MLP will be described.

1. Evaluation of TypeREF performances

From TypeTE to TypeREF

As introduced previously, TypeTE[8] is a pipeline that can genotype both non-reference and reference TE insertions, so does MELT2[7], a widely used TE genotyping tool. The primary difference between MELT2 and TypeTE is that for reference TE insertions, whereas MELT2 genotypes TEs based on short-reads mapped to a reference genome[7], TypeTE genotypes TEs based on the same reads, however, re-mapped to mini-genomes created for each locus consisting of a reference allele (extracted from a reference genome, containing TE) and an alternative allele (does not contain the TE), respectively[8,16]. From now on, the piece of genome containing TE will be called REF and the one without will be called ALT. The top and bottom of Figure 2 show the REF and ALT alleles - two distinct sequences - that make up one "mini-genome", used in TypeTE (Fig 2). Likely due to this modification, TypeTE has been shown to perform comparably to MELT2 for non-reference TE insertion genotyping and to outperform MELT2 for genotyping reference insertions of *Alu* elements, even though both TypeTE and MELT2 apply the same genotype likelihood approach[15] based on read counts and mapping quality[7,8].

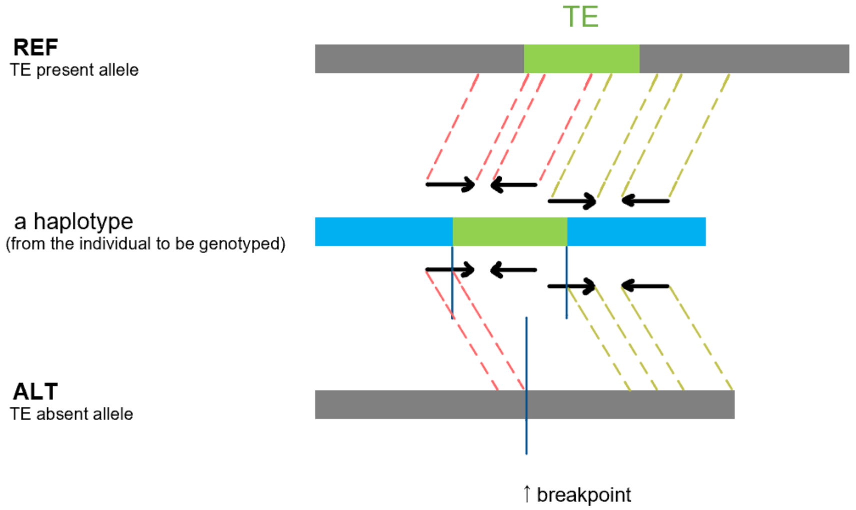


Figure 2. An example of how sequence reads from a bam/cram map to a REF (TE present) allele and to an ALT (TE absent) allele, the two alleles of a mini-genome which TypeREF produces for each TE locus. Each arrow represent a read and →← represents a read-pair (c.f. paired-end sequencing). Notice that when a haplotype contains the TE, which is the case presented above, any portion of a read that overlaps with the TE cannot be mapped to ALT since ALT does not contain TE.

Nevertheless, there are two main limitations with TypeTE: First, because the program has many dependencies, its installation is cumbersome; second, it works only for *Alu*, although there are other types of polymorphic TEs, namely *SVA* and *L1*, which are less abundant than *Alu* but are also one of the most active mobile TEs and have been implicated in diseases[17]. In order to overcome these issues, Dr. Clément Goubert, one of the author's supervisors, has built a new pipeline called TypeREF that uses Nextflow with a Docker/Singularity container that allows the tool to be cross-platform and reproducible. The way TypeREF genotypes TE is identical to that of TypeTE, which, at each locus, computes the likelihood of each possible genotype (0/0, 1/0, or 1/1) based on read count at each of the REF and ALT allele and their associated mapping quality[15]. Before proceeding with TypeREF, benchmarking of the program is required to establish its baseline performance with a non-machine learning genotyping algorithm.

Data

In order to benchmark the performance of TypeREF, genotypes previously obtained by PCR for 39 *Alu* loci in 45 individuals from the 1000 Genome Project (GP) were used[8]. These individuals belong to the CEU (Utah Residents (CEPH) with Northern and Western European ancestry) population[18] and their PCR genotypes for *Alu* were used as a ground truth set ("correct" genotypes). The data, which consists of a total of 1775 PCR genotypes, was obtained from Dr. Goubert's collaborator, Dr. Lindsay Payer [8]. Here, PCR is considered as the "gold standard" to establish TE genotypes.

The input of TypeREF consisted of the following: genomic alignments (BAM/CRAM) of individuals to be genotyped; the reference genome, hg19 (hs37d5); candidate positions of reference TE insertions to genotype, stored in RepeatMasker tracks (Reference TE positions) and a genotype file (vcf) obtained from MELT2 (MELT2 performs both discovery of polymorphism and their genotyping); a text file listing the names/paths of all the individuals (BAM/CRAM files) to be genotyped. The genotype alignments were downloaded from http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase1/ and http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/phase3/. The rest of the data were provided by Dr. Goubert.

Based on these data, genotype predictions for reference *Alu* insertions were obtained from TypeREF. Similarly, MELT2 genotypes were obtained from Goubert et al., 2020[8], which also accounted for 1775 (= 39 loci × 45 individuals) genotypes.

Evaluation of TypeREF performance for *Alu*

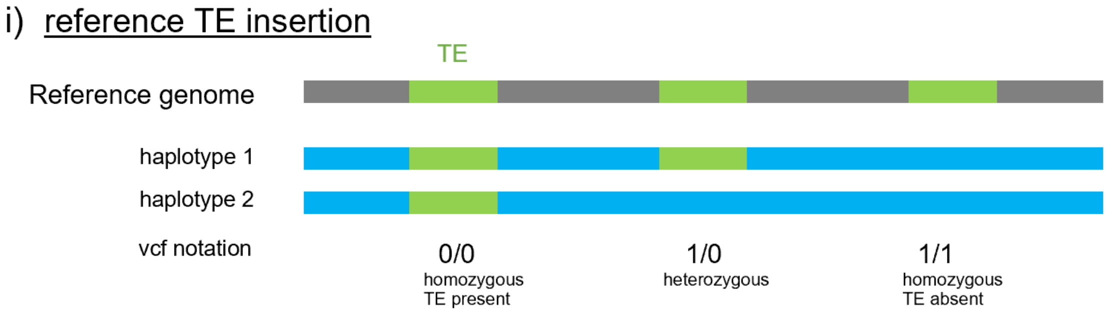
The performance of TypeREF was evaluated by comparing the genotype prediction made by TypeREF (0/0, 0/1, 1/1) to the "true" genotypes obtained from the PCR data. Percentage of matching genotypes between the method (TypeREF or MELT2) and PCR was used: this was calculated by dividing the number of loci whose method (TypeREF or MELT2) genotype and PCR genotype matched by the number of the [loci × individual] combinations (n=1775). In the end, the calculated percentage of matching genotypes for TypeREF and MELT2, respectively, were compared.

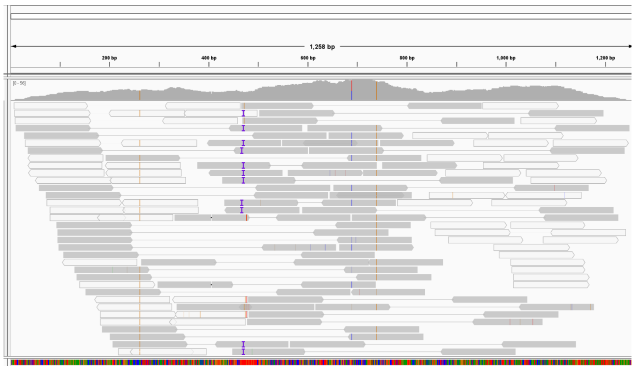
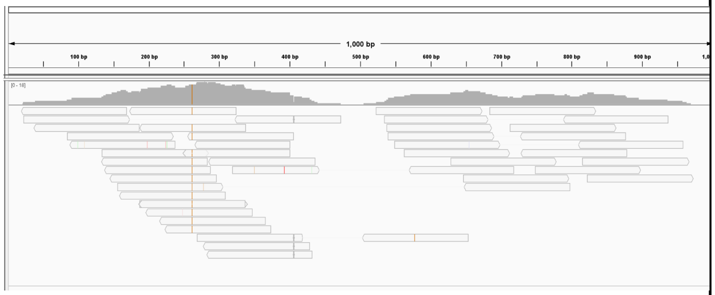
2. Implementation of machine learning models

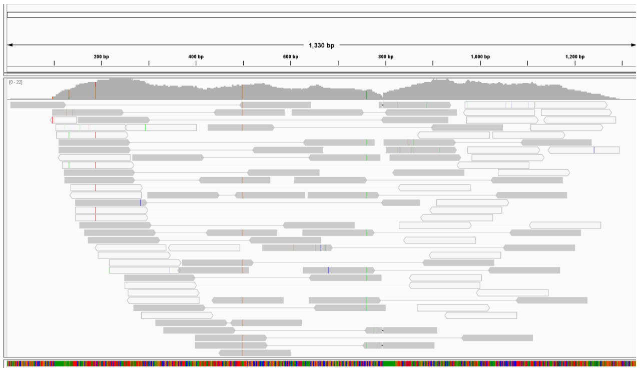
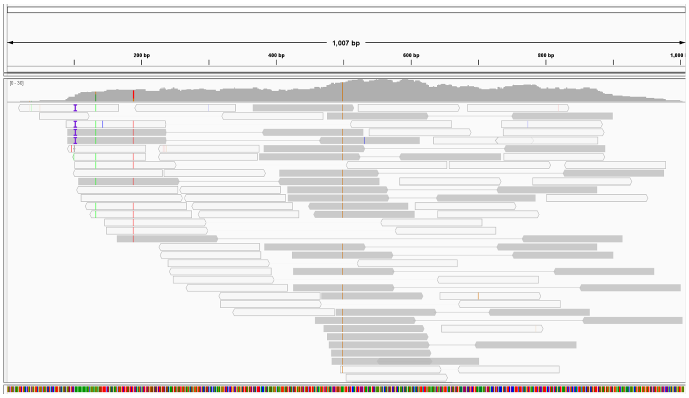
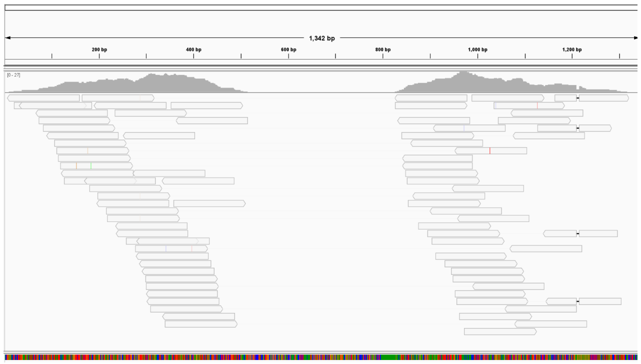
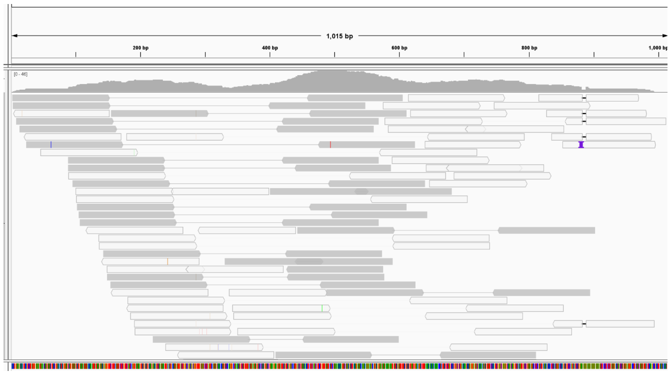
Overview

The main idea of the method is to show a proof of principle for using machine learning algorithms to classify genotypes based on a number of features (signatures of possible genotypes) obtained from the read realignments produced by the program TypeREF. For this report, three simple features are used: (i) the number of reads mapped in proper pair to the REF allele, (ii) the number of reads mapped in proper pair to the ALT allele, and (iii) sum of (i) and (ii). These features were chosen because their proportions characterize each of 0/0, 1/0, and 1/1 genotypes (Table 1). Other possible features will be discussed later in Discussion.

Table 1. Reads are shown in pairs. Note that the average length of an *Alu* element is about 300bp, and 500bp downstream and upstream of an *Alu* insertion breakpoint is taken for the alleles; therefore, the length of REF will be approximately be 1300bp (500bp + 300bp + 500bp) and the length of ALT will be approximately 1000bp (500bp + 500bp). Reads mapped in proper pair for each allele are represented in dark grey; white reads are reads mapped to both REF and ALT alleles and are discarded. Zooming in will increase the resolution of the images.



genotype	**REF** (TE present. TE is represented in green.)	**ALT** (TE absent)
0/0 (TE/TE)		

1/0 (TE/ no)	 1,330 bp 200 bp 400 bp 600 bp 800 bp 1,000 bp 1,200 bp	
1/1 (no/ no)	 1,342 bp 200 bp 400 bp 600 bp 800 bp 1,000 bp 1,200 bp	 1,015 bp 200 bp 400 bp 600 bp 800 bp 1,000 bp

Learning algorithms

Logistic regression, random forest, SVM, and MLP models were implemented. Each algorithm was chosen because: logistic regression is simple linear algorithm and was deemed appropriate given the limited number of features (three); random forest often has good predictive accuracy[19] and have been applied widely across disciplines[20]; SVM has been shown to improve genotyping accuracy of non-reference insertions in xTea[6]; MLP is a deep learning algorithm while the other three are not, however, because all of them can perform supervised learning, including MLP can potentially allow examining the performance of deep learning model vs those that are not.

Logistic regression, random forest, and SVM were implemented using scikit-learn[21] and MLP was implemented using Tensorflow[22]. pandas[23] and numpy[24] libraries were used to store and manipulate training, testing, and validation data. The code, inside a Jupyter notebook (6.3.0), written using Python 3.8.8, is available at: https://github.com/OrangeFrog210/COMP402-565Project.

Data preprocessing

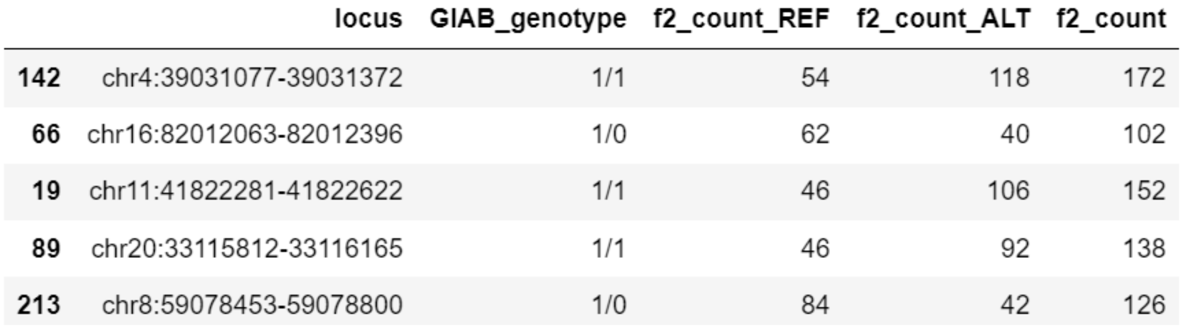
Data

Read alignment files, which contain realignment of reads against TypeREF mini-genomes (each of a mini-genome contain two sequences: REF, which contained *Alu* and ALT, which did not contain *Alu*) were obtained from Dr. Goubert; it consisted of 100 0/0 loci (Alu/Alu) and 460 1/0 (Alu/-) or 1/1 (-/-) loci. The genotypes of these 560 samples, served as true labels, were obtained from Genome in a Bottle (GIAB)[25] (https://www.nist.gov/programs-projects/genome-bottle). Note that this represents a toy dataset made to test the implementation of the model, and will not be the entirety of data that will be used to evaluate the performance of machine learning models on genotyping reference TE insertions. In Discussion, what the data will be replaced by once data becomes available, will be mentioned.

Feature extraction

A total of three features were extracted from bam files using bash scripts and samtools (1.12)[26]. These consisted of i) number of reads mapped in proper pair on the REF allele, ii) number of reads mapped in proper pair on the ALT allele, and iii) sum of i) and ii) (reads mapped in proper pair on either of the mini-genomes). These features were extracted for each of the 560 loci. Justification for choosing these features was provided earlier in Overview in Method.

Table 2. Organization of data in a pandas dataframe



	locus	GIAB_genotype	f2_count_REF	f2_count_ALT	f2_count
142	chr4:39031077-39031372	1/1	54	118	172
66	chr16:82012063-82012396	1/0	62	40	102
19	chr11:41822281-41822622	1/1	46	106	152
89	chr20:33115812-33116165	1/1	46	92	138
213	chr8:59078453-59078800	1/0	84	42	126

Table 3. Description of the fields in Table 2

Column name	Description	samtools command for extracting the feature
locus	Identifier of the locus where TE is present in the reference genome	N/A (Not applicable)
GIAB_genotype	The "true" genotype obtained from GIAB (label). It is later converted to integers: 0/0 → 0, 1/0 → 1, 1/1 → 2	N/A

f2_count_REF	Number of reads mapped in proper pair on the REF allele (contains TE)	COUNT_2_REF=$(samtools view -f 0x2 mapped.sorted.bam | awk '{if ($3~"_genome" && $7=="=")
f2_count_ALT	Number of reads mapped in proper pair on the ALT allele (no TE)	COUNT_2_ALT=$(samtools view -f 0x2 mapped.sorted.bam | awk '{if ($3~"_alternative" && $7=="=") {print $0}}' | wc -l)
f2_count_count	The number of reads mapped in proper pair to REF or ALT	COUNT_2=$(samtools view -f 0x2 mapped.sorted.bam | awk '{if ($7=="=") {print $0}}' | wc -l)

Training, testing, and validation

Regardless of the learning algorithm, training and test sets were splitted into 80:20. The models were trained with a train dataset and prediction of genotypes was then performed on a test dataset. For logistic regression, random forest, and SVM, input data, namely the values of features, were normalized by subtracting the mean of training data from the original value and dividing it by standard deviation of the training data. 5-fold cross validation (CV) was used to validate the models.

Logistic regression

Logistic regression was implemented using scikit-learn. For penalty, L2 regularization was used to control for overfitting. Contribution of each feature was examined by obtaining weights of the trained model using the coef_ function.

Random forest

Random forest was implemented using scikit-learn. The number of trees in the forest was set to 20, no limit was set to the maximum depth of the tree, and the number of features to consider (when looking for the best split) was set to “auto”, meaning the choice was left to the classifier.

SVM

SVM was implemented using scikit-learn. Maximum number of iterations was set to 1000 and the C parameter, which tells the SVM optimization method the amount to avoid misclassifying each training example, was set to the default value 1. Contribution of each feature was examined by obtaining weights of the trained model using the coef_ function.

MLP

MLP was implemented using Tensorflow. MLP was run for 200 epochs with a batch size of 20. There were three layers in total: an input layer, a hidden layer, and an output layer. For activation functions, relu was used for the input layer and the hidden layer, and softmax function was used for the output layer. For the input layer, input dimension was set to 3, based on the number of features. Categorical cross entropy loss was used as the loss function, and adam was used as the optimizer with learning rate 0.01.

Evaluation metrics

Genotyping accuracy, obtained from cross_val_score, was used to evaluate the models.

Results

1. Evaluation of TypeREF performances

Genotyping accuracy: TypeREF, and TypeREF vs MELT2

In order to evaluate the genotyping accuracy of TypeREF, including to obtain its baseline performance, genotypes predicted by TypeREF were compared to the ground truth genotypes provided from the PCR data. When the percentage of matching genotypes between TypeREF and PCR was calculated, a result of 80.4% match was obtained out of a total of 1775 genotypes. In order to assess the relative performance of TypeREF to other programs, the percentage of matching genotypes between MELT2 and PCR was also calculated, and a mean accuracy of 52.1% was obtained. Figure 3 shows the result as well as the distribution of genotyping accuracy of the 39 loci, as mean genotyping accuracy for each of the 39 loci over 45 individuals were calculated (Fig 3).

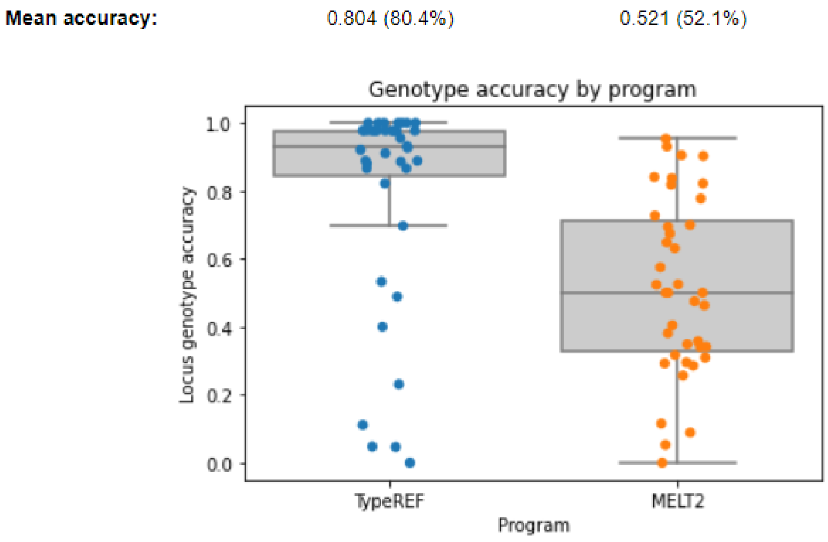


Figure 3. Genotype accuracy of TypeREF vs MELT2. The truth set is genotypes provided by PCR for 39 *Alu* loci in 45 individuals from the 1000 Genome Project (GP). Each dot represents the genotyping accuracy of an individual locus

(there are 39 dots in total, representing the 39 loci), which was obtained by taking the mean genotyping accuracy for that locus based on the results of 45 individuals.

2. Performance of SVM and MLP for genotyping reference Alu insertions

Performance of logistic regression

Logistic regression was implemented using scikit-learn with L2 regularization. The mean accuracy of 5-fold CV was 0.879.

Performance of random forest

Random forest classifiers were implemented using scikit-learn. The mean accuracy of 5-fold CV was 0.897.

Performance of SVM

SVM was implemented using scikit-learn, with two types of kernel functions. The mean accuracy of 5-fold CV for SVM for the model with linear kernel and for the model with rbf kernel, were 0.886 and 0.888, respectively.

Performance of MLP

MLP was run for 200 epochs with a batch size of 20. Figure 4a) shows the change in accuracy with respect to epoch and Figure 4b) shows the change in loss with respect to epoch (Fig 4 a) b)). In each fold of the validation steps, accuracy plateaued near 15 epochs at a value around 0.89.

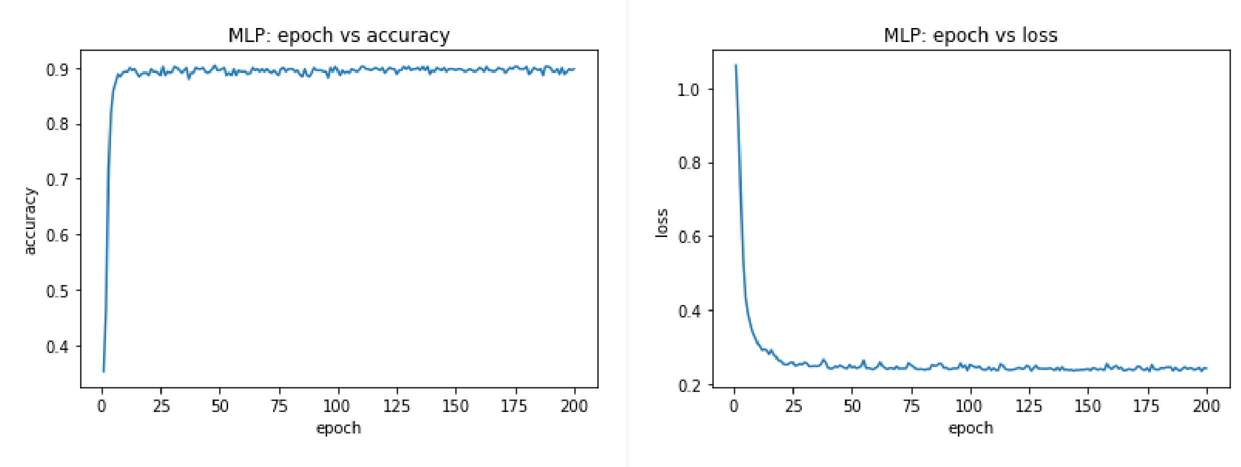


Figure 4a). Accuracy of a MLP model plotted against epoch

b). Loss of a MLP model plotted against epoch

Performance of the models

As shown in the following table, accuracy of SVM barely differed between the linear and rbf kernels. The performances among models were comparable, with logistic regression having the lowest performance but only slightly below the others (Table 4). Note that mean accuracy (from 5-fold CV) closer to and even above 0.90 were observed for logistic regression from time to time in different runs.

Table 4. Model Performances obtained by taking the mean of 5-fold cross validation.

	Logistic	SVM linear	SVM rbf	MLP	Random forest
Accuracy	0.879	0.891	0.888	0.897	0.897

Contribution of each feature in logistic regression and SVM

For logistic regression and for SVM with linear kernel, the coefficients of model, each corresponding to i) number of reads mapped in proper pair on the REF allele ii) number of reads mapped in proper pair on the ALT allele, and iii) sum of i) and ii), were obtained. For both logistic regression and SVM, the weight corresponding to feature iii) had comparably low magnitude compared to the other two features (Table 5, Fig 5).

Table 5. Coefficients of a logistic regression model and SVM with linear kernel.

	Logistic regression	SVM (linear kernel)
Number of reads mapped in proper pair on the REF allele	2.86	1.831
Number of reads mapped in proper pair on the ALT allele	-2.71	-2.08
Number of reads mapped in proper pair	0.14	-0.25

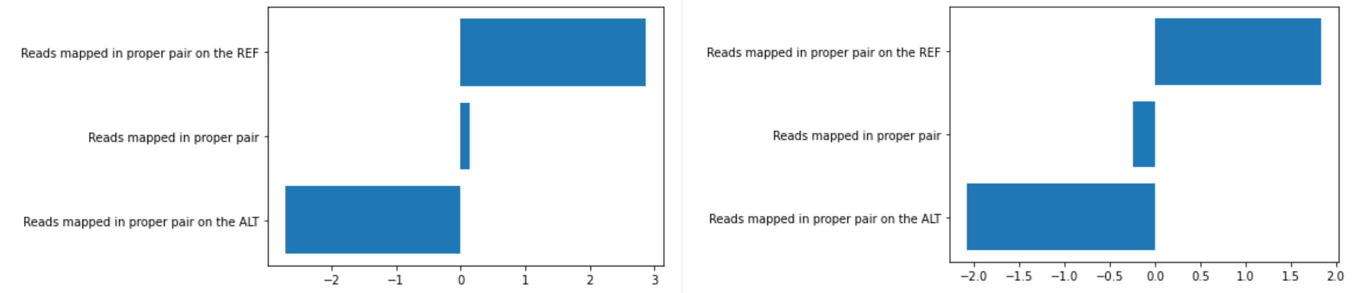


Figure 5 a). Coefficients of a logistic regression model **b).** Coefficients of a SVM model with linear kernel.

Discussion

This report presented machine learning based methods to genotype reference TE insertions and some preliminary results. Despite their implications in healthcare, genotyping of reference TE insertions have not been worked on as extensively as genotyping non-reference TE insertions, and it still has room for improvement, especially with respect to the accuracy of genotypes existing tools like TypeTE and MELT2 produce. The main idea of the method is to use machine learning models, and based on features that characterize each of the possible genotypes, predict genotypes of a locus of a given individual. In order to assess the performance of machine learning models compared to an existing method, baseline performance of TypeREF, a tool for genotyping reference TE insertion, was obtained. Then, using a toy dataset, performances of four machine learning models - logistic regression, random forest, SVM, and MLP - were assessed so as to check the feasibility of the project and to obtain preliminary results.

1. Evaluation of TypeREF performances

When the percentage of matching genotypes between TypeREF and PCR, and MELT2 and PCR were calculated, respectively, it led to 80.4% match and 52.1% match. Because TypeREF remaps alignments to two sequences in a mini-genome, each representing an allele containing TE and the other without TE instead of mapping reads to one reference genome like MELT2 does, it is not surprising that TypeREF outperformed MELT2; this is also consistent with previous results obtained from TypeTE, in which TypeREF is based on[8]. However, for TypeREF, 80% genotype match implies that 1 out of 5 times on average, it will produce an incorrect genotype. If the mis-genotyped locus happened to be important in regulating gene expression of, for example, a disease gene, then incorrectly labeling an individual could potentially mean incorrectly informing the person about his or her disease risk.

In addition, from Figure 3, one can see that although most of the 39 loci have high genotyping accuracy, some above 0.95 and some even at 1.0, there are nine loci with genotyping accuracy below 0.8, four of which below 0.2 (Fig 3). Causes of these low genotyping accuracy for the particular loci are yet to be known, however, by looking at locus-specific genotyping accuracy of machine learning models on these loci, it may be possible to figure out whether the problem is due to the current TypeREF genotyping algorithm or not. If the problem was indeed within the TypeREF algorithm, then the author would expect machine learning models to perform much better at these loci.

2. Performance of the models for genotyping reference *Alu* insertions

Logistic regression, random forest, SVM, and MLP were implemented for genotyping reference *Alu* insertions. For logistic regression, the mean accuracy of 5-fold CVs fell between 0.86-0.91. As logistic regression may be more suitable when the number of features is not too large, it may be interesting to compare model performances when more features are added later in the project (more elaboration on this can be found later in Discussion). In addition, comparing the performances of unregularized vs L1 regularized vs L2 regularized models may also be a possible addition in the project, although it should not be the main focus.

For random forest, the mean accuracy of 5-fold CVs fell between 0.86-0.91, most values being around 0.875-0.89. The author did not set the maximum possible depth of the tree, however, changing the parameter may affect the result and may be interesting to explore.

For SVM, initially, hypothesizing that a model with non-linear decision boundary would lead to more flexibility and a higher prediction accuracy, the author tested linear and rbf kernels, the latter of which has non-linear decision boundary. However, no significant difference in mean prediction accuracy of the models were observed when the kernels were changed (Table 4). This may make sense, as demonstrated in Table 1, the number of reads mapped in proper pair on REF allele and ALT allele are noticeably different among the three genotypes, making them easily distinguishable regardless of the kernel. When more complex features that may not differ among, for example, two of the three genotypes, are used, the choice of kernel may lead to noticeable difference in results. One remark on training of SVM is that normalizing model inputs (X_test and X_train) led to a previously non-converging model to converge.

Contribution of each feature in a model can be represented by the magnitude of coefficients of the model. For logistic regression, the magnitude of the weight for each of the features, namely i) number of reads mapped in proper pair on the REF allele ii) number of reads mapped in proper pair on the ALT allele, and iii) sum i) and ii) were 2.86, 2.71, and 0.14, respectively (Table 5, Fig 5a)). For SVM with linear kernel, the values were 1.831, 2.08, and 0.25 (Table 5, Fig 5 5b)). It is not surprising that the number of reads mapped in proper pair has a much smaller contribution (0.14 and 0.25) i.e. less informative than the other two, since it is merely the sum of the other two features. In addition, it makes sense that the number of reads mapped in proper pair on the REF and that of ALT have the greatest contributions, since their values vary greatly depending on an individual's genotype. For example, as illustrated in the left column of Table 1, the number of reads mapped in proper pair on REF allele (contains TE) is significantly less for a homozygous TE absent (1/1) individual than a homozygous TE present individual.

As shown in Figure 4, the accuracy of MLP plateaued at around 0.89. Even when the model was run for multiple times, this trend persisted, suggesting that modifications may be needed to either or both of input (e.g. sample size, features) or parameters of the model (activation function, number of layers, etc.).

No significant difference among the performances of logistic regression, random forest, SVM, and MLP was observed, each reaching accuracy around and between 0.87 and 0.90 for the majority of cases. While there were some differences across model trainings, the values were consistently around the said values, the highest prediction accuracy observed for SVM being in the range (0.91-0.92). However, while the author trained models multiple times, the accuracy seemed to never surpass 0.90 for logistic regression, 0.91 for random forest, 0.92 for SVM, and 0.90 for MLP.

The said range of prediction accuracy may be the highest possible values that can be obtained from the current models given that the conditions under which the models are put in have several limitations. For example, the data used to train the models contain only three features, which are the counts of reads properly mapped in pairs. The quality of the features may be limiting, as mapping quality of the reads were not filtered, suggesting that bad quality reads might have been counted in the read counts. In addition, there are other features that can be more informative that, when added, may improve the accuracy of the models, which have been used by some of the existing tools for genotyping TE and more generally, for genotyping SVs[6,14]. These include, but are not limited to: the numbers of discordant read pairs, split-read, and fully-mapped reads (Fig 6). For example, discordant read pairs, shown in blue in Fig 6, are read pairs whose insert size (the distance between the two reads in the pair) are unusually long compared to average insert size (Fig 6). This is expected to happen more frequently in mapping of homozygous TE absent individual on a REF genome than in mappings of other two genotypes, since, because the individual lacks the TE in both of the alleles, the read pair must be mapped over the TE, resulting in an insert size that is [the insert size of the pair in ALT genome + length of the TE]; this will be much bigger than the average insert size. Adding such features may help improve genotyping accuracy of the models.

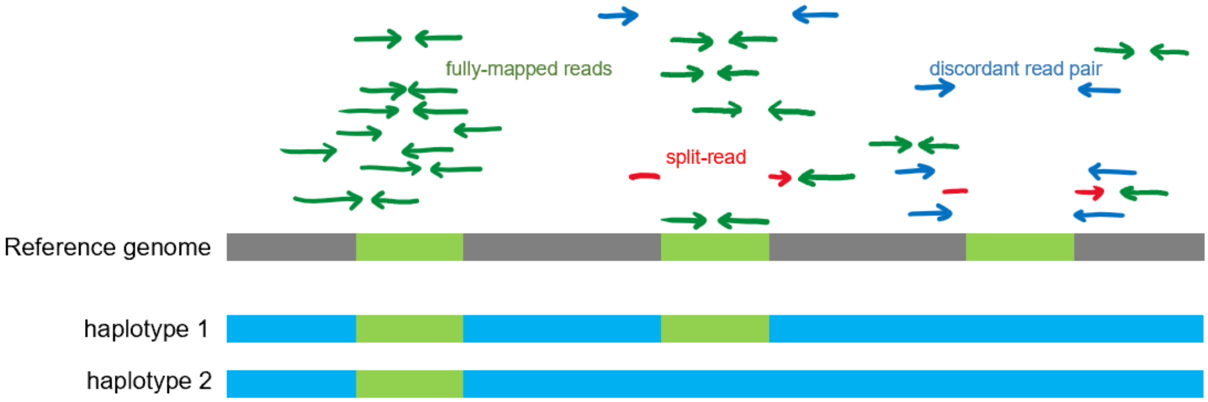


Figure 6. Some of the possible features for the genotypes, namely fully-mapped reads, split-read, and discordant read pair for reference TE insertions. A converse of fully-mapped reads is partially mapped reads, in which part of the read has nowhere in the genome to map to. Each arrow represents a read, and the arrows pointing to each other correspond to a read pair, which consists of a set of two reads. In theory, if an individual is 1/1 (-/-) for a reference TE insertion, there should be no read mapping that overlaps with the region corresponding to TE, as illustrated on the right side of the diagram. This is because the individual has no genomic region that corresponds to the TE for that locus. Similarly, split reads are only expected to occur in 1/0 (TE/-) and 1/1 (-/-) cases where the individual has at least one allele without TE, because the read coming from a no-TE-allele is "interrupted" by the presence of TE in the reference genome.

While the project used logistic regression, random forest, SVM, and MLP, other machine learning methods, as mentioned in Introduction, were found but were either not selected or could not be implemented due to time constraint. For example, while the author attempted to implement a 3-state HMM, it could not be completed due to difficulty in calculating some of the transition probabilities. The issue was that, in the homozygous TE absent case (1/1), because any region without TE in the reference genome can potentially be deemed 1/1, there is no set length for the genotype unlike around 300bp for an insertion of *Alu* and thus, the number of 1/1 *Alu* elements, which influences transition probabilities, can vary depending on how we define it. This makes calculating the transition probabilities involving the 1/1 genotype complex, and consequently, HMM model, which requires transition probabilities, could not be implemented in time. However, the author plans to investigate further in literature to be able to implement a HMM model, whose results may be interesting to compare with the results of others models, especially since it relies solely on the total read count, meaning that there is no need for a complex feature extraction process and is unsupervised.

Another method that was on the table but was not chosen is image-based approaches in which aligned sequencing reads data are converted into images and are fed into Convolutional Neural Network (CNN) for genotyping[11,12]. Since no other methods represent alignments as images and perform genotyping as image classification, adapting the methods and comparing its performance with that of other models would be very interesting; however, because of the heavy pre-processing steps required for this approach, other algorithms were chosen. Nevertheless, implementing CNN may also shed new light on genotyping TE.

As an endnote, because most of the machine learning method have not been used in genotyping polymorphic TE insertions, a method that improves genotyping accuracy of reference-TE insertions may also be able to be adapted to genotyping non-reference TE insertions, which are more abundant in the human genome.

TypeREF genotyping method vs Machine learning genotyping

We cannot conclude that machine learning methods outperformed the method used in TypeREF, since the benchmarking of TypeREF used PCR genotypes as a truth set (genotypes) while machine learning models where build on a limited toy dataset that uses a different truth set; however, the genotyping accuracy of 0.86 or above based on three basic features (read counts without quality value of the read mapping) is encouraging, as it is higher than the benchmarked accuracy of 0.804 obtained with the original genotyping algorithm of TypeREF.

Conclusion and future steps

The preliminary results suggest the potential of machine learning models in genotyping reference *Alu* insertions by achieving genotyping accuracy between 0.86 and 0.91. However, because the results are based on toy data set which was used to i) test that feature extraction and implementation of the models are possible and to ii) get an idea of results (genotyping accuracy) that may be achieved, the next step is to use a larger dataset that contains thousands of samples. For this, the author plans to use 1000 GP data, which has many reference *Alu* loci but whose genotypes ("labels") are of highest standards of quality[27]. However, these are not as reliable as more limited but more accurate dataset, such as PCR or the benchmark individual HG002 from GIAB[25] in which SVs, including TEs, have been obtained by combining multiple sequencing technologies, including long-reads. Thus, once models are trained and tested with 1000GP data, the author will then use the GIAB dataset, which is independent from 1000 GP data and has reliable "true labels", to validate the model. The idea is that, if models trained on 1000 GP data performs well on GIAB data, that could suggest that with a large and diverse train set, accurate genotyping may be possible even if the quality of labels are not "perfect". The independence of the datasets can suggest the generalizability of the model if a high accuracy is achieved during validation. Next,the author will be working on (i) examining and extracting different features that are suspected to improve genotyping accuracy of reference insertions (e.g. number of discordant read pairs, number of split-read) and (ii) testing and improving the existing models presented in this report, and (iii) implementing other models, such as HMM and CNN.

References

1. Bourque, G. *et al.* Ten things you should know about transposable elements. *Genome Biol.* **19**, 199 (2018).
2. Initial sequencing and analysis of the human genome. *Nature* **409**, 860–921 (2001).
3. Payer, L. M. & Burns, K. H. Transposable elements in human genetic disease. *Nat. Rev. Genet.* **20**, 760–772 (2019).
4. Which transposable elements are active in the human genome? *Trends Genet.* **23**, 183–191 (2007).
5. Chen, J.-M., Chuzhanova, N., Stenson, P. D., Férec, C. & Cooper, D. N. Meta-analysis of gross insertions causing human genetic disease: novel mutational mechanisms and the role of replication slippage. *Hum. Mutat.* **25**, 207–221 (2005).
6. Chu, C. *et al.* Comprehensive identification of transposable element insertions using multiple sequencing technologies. *Nat. Commun.* **12**, 3836 (2021).

7. Gardner, E. J. *et al.* The Mobile Element Locator Tool (MELT): population-scale mobile element discovery and biology. *Genome Res.* **27**, 1916–1929 (2017).

8. Goubert, C. *et al.* TypeTE: a tool to genotype mobile element insertions from whole genome resequencing data. *Nucleic Acids Res.* **48**, e36 (2020).

9. Chen, X. & Li, D. ERVcaller: Identify polymorphic endogenous retrovirus (ERV) and other transposable element (TE) insertions using whole-genome sequencing data. *bioRxiv* 332833 (2018) doi:10.1101/332833.

10. Goubert, C., Zevallos, N. A. & Feschotte, C. Contribution of unfixed transposable element insertions to human regulatory variation. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **375**, 20190331 (2020).

11. Poplin, R. *et al.* A universal SNP and small-indel variant caller using deep neural networks. *Nat. Biotechnol.* **36**, 983–987 (2018).

12. Cai, L., Wu, Y. & Gao, J. DeepSV: accurate calling of genomic deletions from high-throughput sequencing data using deep convolutional neural network. *BMC Bioinformatics* **20**, 665 (2019).

13. Tang, M., Hasan, M. S., Zhu, H., Zhang, L. & Wu, X. vi-HMM: a novel HMM-based method for sequence variant identification in short-read data. *Hum. Genomics* **13**, 1–12 (2019).

14. Chu, C., Zhang, J. & Wu, Y. GINDEL: accurate genotype calling of insertions and deletions from low coverage population sequence reads. *PLoS One* **9**, e113324 (2014).

15. Li, H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics* **27**, 2987–2993 (2011).

16. Wildschutte, J. H., Baron, A., Diroff, N. M. & Kidd, J. M. Discovery and characterization of Alu repeat sequences via precise local read assembly. *Nucleic Acids Res.* **43**, 10292–10307 (2015).

17. Beck, C. R., Garcia-Perez, J. L., Badge, R. M. & Moran, J. V. LINE-1 elements in structural variation and disease. *Annu. Rev. Genomics Hum. Genet.* **12**, 187–215 (2011).

18. 1000 Genomes Project Consortium *et al.* A global reference for human genetic variation. *Nature* **526**, 68–74 (2015).

19. Caruana, R. & Niculescu-Mizil, A. An empirical comparison of supervised learning algorithms. *Proceedings of the 23rd international conference on Machine learning - ICML '06* (2006) doi:10.1145/1143844.1143865.

20. Murphy, K. P. *Machine Learning: A Probabilistic Perspective*. (MIT Press, 2012).

21. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* **12**, 2825–2830 (2011).

22. Developers, T. *TensorFlow*. (Zenodo, 2021). doi:10.5281/ZENODO.4724125.

23. Reback, J. *et al. pandas-dev/pandas: Pandas 1.3.5*. (Zenodo, 2021). doi:10.5281/ZENODO.3509134.

24. Harris, C. R. *et al.* Array programming with NumPy. *Nature* **585**, 357–362 (2020).

25. Genome in a bottle—a human DNA standard. *Nature Biotechnology* vol. 33 675–675 (2015).

26. Danecek, P. *et al.* Twelve years of SAMtools and BCFtools. *Gigascience* **10**, (2021).

27. Byrska-Bishop, M. *et al.* High coverage whole genome sequencing of the expanded 1000 Genomes Project cohort including 602 trios. *bioRxiv* 2021.02.06.430068 (2021) doi:10.1101/2021.02.06.430068.