bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

SICaRiO: Short Indel Call filteRing with bOosting

Md Shariful Islam Bhuyan[1,2], Itsik Pe'er[1] and M. Sohel Rahman[2]

[1]Columbia University, in the city of New York, NY 10027, United States

[2]Bangladesh University of Engineering and Technology, Dhaka-1000, Bangladesh

Abstract

Despite impressive improvement in the next-generation sequencing technology, reliable detection of indels is still a difficult endeavour. Recognition of true indels is of prime importance in many applications, such as, personalized health care, disease genomics, population genetics etc. Recently, advanced machine learning techniques have been successfully applied to classification problems with large-scale data. In this paper, we present SICaRiO, a gradient boosting classifier for reliable detection of true indels, trained with gold-standard dataset from genome-in-a-bottle (GIAB) consortium. Our filtering scheme significantly improves the performance of each variant calling pipeline used in GIAB and beyond. SICaRiO uses genomic features which can be computed from publicly available resources, hence, we can apply it on any indel callsets not having sequencing pipeline-specific information (e.g., read depth). This study also sheds lights on prior genomic contexts responsible for indel calling error made by sequencing platforms. We have compared prediction difficulty for three indel categories over different sequencing pipelines. We have also ranked genomic features according to their predictivity in determining false indel calls.

Introduction

Reliable detection of short genomic insertion or deletion (Indel) still remains challenging for standard alignment-based variant calling methods (Wala et al. 2018). Indel detection has recently attained more focus from research community due to the advancement of next-generation sequencing (NGS) technologies. Short indels are genomic variants defined by insertion or deletion of one or more base pairs (defined as <50bp in (Alkan et al. 2011)) at a particular locus within the DNA. Although rarer than SNPs (Single Nucleotide Polymorphisms), they comprised 16% to 25% of all genetic variations, the second most abundant form of polymorphism (Mills et al. 2006; Mullaney et al. 2010).

Several genetic disorders are linked to deleterious indels such as cystic fibrosis, fragile X syndrome, trinucleotide repeat disorders, Mendelian disorders and Bloom syndrome (Kanehisa et al. 2015) and NGS has become the standard tool for disease variant discovery (Koboldt et al. 2013; Wang et al. 2013). Short indels are also assumed to cause some cancers, e.g., acute myeloid leukaemia, and lung cancer. Besides, indels can also modify promoter structures and affect gene expressions (Cheung and Spielman 2009). Indels are also used as genetic markers for populations (Vali et al. 2008). Due to their importance in population genetics and clinical

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

genomics, accurate indel detection is of prime importance; more so in the case of germline de-novo and somatic mutations which are rare but mostly account for aberrant genomic alterations. NGS technologies (Quail et al. 2012) have continued to mature, become cheaper and are gradually getting adopted into clinical applications. Accuracy of SNP detection has reached at the level of 99th percentile[1]. However, indel detection accuracy is not at the par, despite the advent of a large number of indel calling tools (Hasan et al. 2015). Highly used indel callers have an accuracy around 60% over whole genome (containing both high confidence and non-high confidence regions) (Cornish and Guda 2015; Hasan et al. 2015), which is in fact validated in this work. Performance studies over only high-confidence regions (regions for which we have a complete map of indels) report high precision. For example, on PrecisionFDA truth challenge[1], DeepVariant (Poplin et al. 2018) reports a precision greater than 99% for indels only from high-confidence regions. When measured over whole genome, its precision drops to 63%. When we have taken concordance among multiple callers into account, performance drops even further. Indel calling error depends on sequencing platform characteristics, aligner and caller attributes as well as genomic context of the region (high vs. non-high confidence regions as mentioned in (Zook et al. 2016). Variant calling pipelines differ among their employed techniques and error landscapes. The techniques for one platform are hard to adapt across other platforms. The exact model of error distribution is also unknown (Poplin et al. 2018).

Over the years, lots of indel detection tools have appeared in the arena of NGS technologies. A comprehensive survey is out of the scope of this paper. Among the mainstreams, there are Samtools (Li 2011), GATK (McKenna et al. 2010a), Pindel (Ye et al. 2009), SOAPIndel (Li et al. 2013), VarScan (Koboldt et al. 2009), SplazerS (Emde et al. 2012), Dindel (Albers et al. 2011), SVM-M (Yang et al. 2016), IndelSeek (Au et al. 2017), Gindel (Chu et al. 2014), Sclapel (Fang et al. 2016), VarDict (Lai et al. 2016), LoFreq (Wilm et al. 2012) and more. A performance evaluation for some of them can be found in (Sandmann et al. 2017). All of them take aligned sequencing reads and make indel calls. To improve accuracy, ensemble of different methods has also been employed, e.g., HugeSeq [25] and BaySiC [26]. Data-centric approaches have also been explored, such as, SVM2 (Chiara et al. 2012), ForestSV (Michaelson and Sebat 2012), (Hwang et al. 2014) and Platypus (Rimmer et al. 2014). To address more difficult scenarios (e.g., de-novo and somatic indels) there exist tools like DeNovoGear (Ramu et al. 2013), SomaticSeq (Fang et al. 2015), and DNMFilter (Liu et al. 2014) among others. To validate these tools, projects have been launched to establish gold-standard datasets, e.g., Genome in a bottle (Zook et al. 2016, 2014, 2018) as well as SVclassify (Parikh et al. 2016). Recently deep learning techniques have made their way into genomic data analysis (Telenti et

[1] https://precision.fda.gov/challenges/truth/results

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

al. 2018; Zou et al. 2019), particularly in variant calling, for both short-read data (Illumina, Complete Genomics), e.g., DeepVariant (Poplin et al. 2018), GARFIELD-NGS (Ravasio et al. 2018) and long-read data (Pacific Bioscience, Oxford Nanopore), e.g., Clairvoyante (Luo et al. 2019).

In this paper, we have shown that, training of machine learning-based variant filters with the gold-standard datasets, using genomic contexts improves the performance of indel callers while retaining high sensitivity. Most, if not all, indel-callers (including above mentioned ones) rely on actual platform specific sequence reads which might be unavailable for large public datasets, e.g., exome aggregation consortium (ExAC) (Lek et al. 2016) and haplotype reference consortium (HRC) (McCarthy et al. 2015). Our filter can post-prune potential false indel calls to deliver a more reliable variant set.

Our specific contributions in this paper are summarized below:

1. We have developed a machine learning-based probabilistic filtering scheme to reliably identify false short indel calls. This improves state-of-the-art indel detection performance significantly, particularly when we consider whole-genome callsets (including non-high confidence regions or NHCR). We plan to make SICaRiO publicly available which can be trained and seamlessly integrated with any variant calling pipeline. For now, we will provide the tool and annotation data on request.
2. We have demonstrated that genomic contexts capture important prior information useful for reliable indel detection. To the best of our knowledge, no indel callers currently use this kind of features, e.g., GERP scores (Davydov et al. 2010), local DNA structures etc. We have also ranked genomic features according to their predictivity in determining false indel calls.
3. We have shown that different sequencing platforms make analogous calling mistakes albeit with different degree of similarity. This enables us to apply models trained with the callset generated from one platform over the callset generated from another platform. We have further estimated the performance for this cross-platform scenario.

Results

Performance Improvement After Variant Filtering

We have demonstrated the performance of our scheme by comparing the precisions before and after applying the variant filter. In ***Figure 1***, we have shown precision improvement of five different variant calling pipelines (each consisting a combination of sequenced base caller, read aligner and variant caller) over three different indel categories (see ***Supplementary Table 3***).

The definitions of these three different indel categories are as follows:

1. Homopolymers: In this category, the inserted/deleted nucleotides as well as the previous or following nucleotide (the flanking prefix or suffix context) consist of only one type of nucleotides. Example: [A → AA], [AA → A]

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

2. Tandem repeats: The variant in this category is not a homopolymer; the number of inserted/deleted nucleotides is more than one and an inserted/deleted sequence is followed or preceded by the same sequence. Example: [AT → ATAT]
3. Aperiodic indels: Any other insertion/deletion which is not a homopolymer or a tandem repeat belongs to this category. Example: [A → AGT]

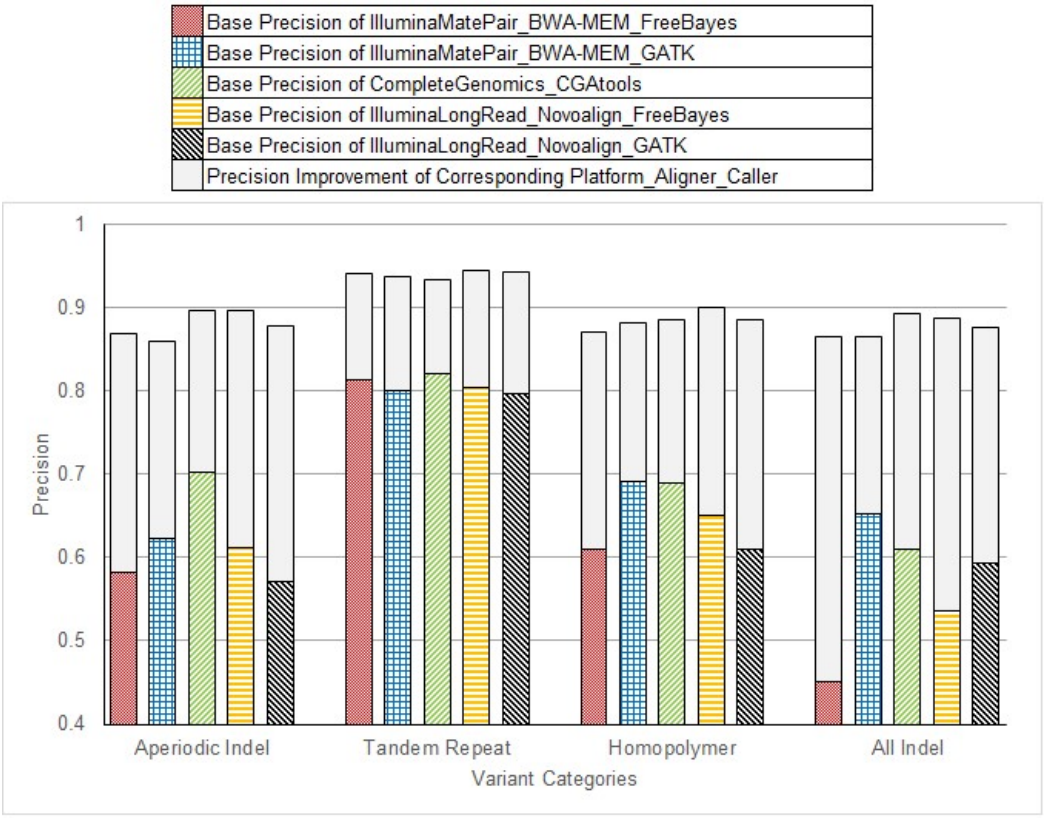


Figure 1: Precision Improvement across Five Different Platforms and Variant Categories

We have also evaluated all indels together, when our training and test data contain all variant categories.

For aperiodic indels, before applying the filter, the maximum base precision of 70.3% is achieved by Complete Genomics (Peters et al. 2012). Illumina (mate pair[2]) with BWA-MEM (Li 2013) and GATK (McKenna et al. 2010b) ranks right after that with a base precision of around 62.22%. Illumina (long read[3]) with NovoAlign[4] and GATK resides at the bottom of the table with a base precision of 57.16%. Application of our filter improves performances across all platforms/callers among which Illumina (long read) with NovoAlign and GATK shows the highest margin of improvement (30%). Complete Genomics shows the least albeit significant

[2] https://www.illumina.com/science/technology/next-generation-sequencing/mate-pair-sequencing.html
[3] https://www.illumina.com/science/technology/next-generation-sequencing/long-read-sequencing.html
[4] http://www.novocraft.com/products/novoalign/

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

improvement (19%). The highest precision (89.70%) is achieved by Illumina (long read) with NovoAlign and FreeBayes (Garrison and Marth 2012) at a recall rate of 89.32%. The lowest precision (85.95%), not too far from the maximum, is achieved by Illumina (mate pair) with BWA-MEM and GATK at a recall rate of 88.44% (see ***Supplementary Table 3***).

For homopolymers and combined indel calls, most of the measures, i.e., base precision, filtered precision and platform recall show similar results like those of aperiodic indels. Here improvement is even bigger in some cases. For example, Illumina (mate pair) with BWA-MEM and GATK shows 41.5% precision improvement at a recall rate of 79.5% for combined indel calls. For tandem repeats, we have observed higher precision (93%) across all platforms at a superior recall rate (97%).

In ***Supplementary Table 4***, we have compared the precision improvement for combined indel callset of five more platforms including DeepVariant for which precision improvement is greater than 18.44% while keeping the platform recall at 86.25%. For details see ***Supplementary Table 4***.

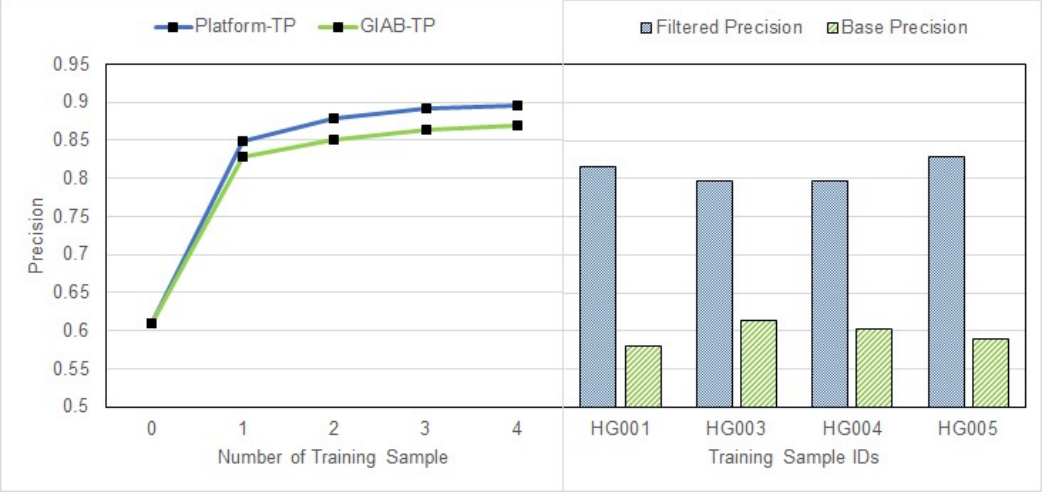


Figure 2: [*Left panel*] Precision vs. Training Sample Size for Complete Genomics Platform. True Positives Come from either True Platform Calls (Blue) or Genome-in-a-Bottle Gold Standard (Green). [*Right panel*] Filtered and Base Precision for Different Test Samples.

We have investigated the effect of training sample size over precision to demonstrate the efficacy and extensibility of our filter in case of the availability of more samples having gold standard callset. We have used training sample size from one to four. All four training samples and the fixed test sample (HG002) are sequenced by Complete Genomics platform. The first data point in ***Figure 2*** [Left panel] is the base precision (no filter applied). This shows that the first training sample gives the largest improvement. Subsequently, the improvement slows down and after adding the fourth sample, tends to saturate. We have also investigated the effect using all gold-standard indel calls instead of only those true indels called by Complete

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Genomics as the true positive training callset. We can see both approaches produce similar curves but training with only Complete Genomics indel callset always performs better.

If training and test samples are different, there may exist a set containing indels common to both training and test callset. If the samples are from same population, this set will be larger and boost the classifier performance positively. To examine the interchangeability of the samples without common indel effect, we have used a single sample for both training and testing. From each sample we have trained with 80% indels and tested over the remaining non-overlapping 20% indels. We have used four unrelated samples, coming from three different population and observed small (3%) precision variability (see ***Figure 2***[Right panel]).

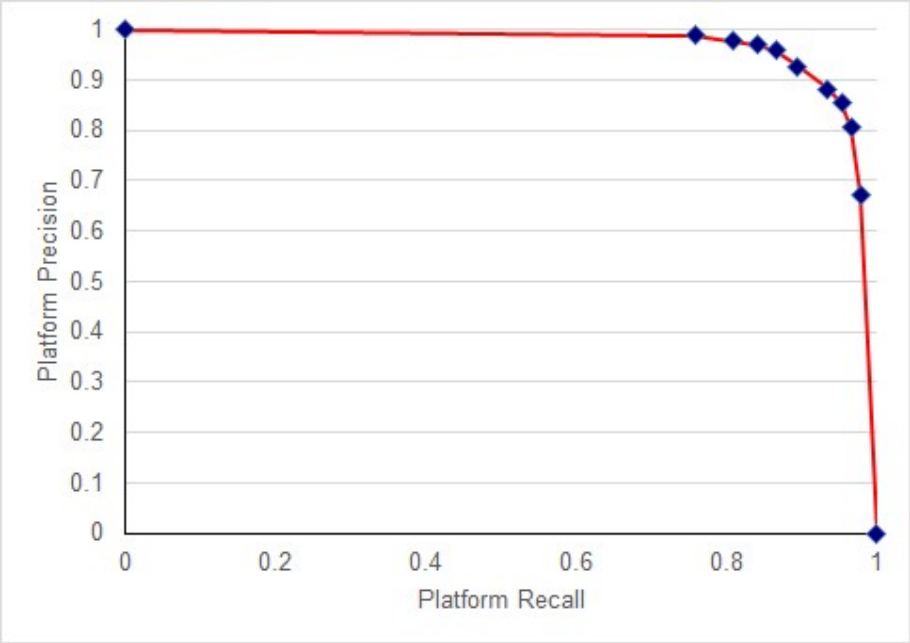


Figure 3: A Precision-Recall Curve for Complete Genomics Platform Trained over Sample HG001, HG003, HG004, HG005 and Tested over Sample HG002

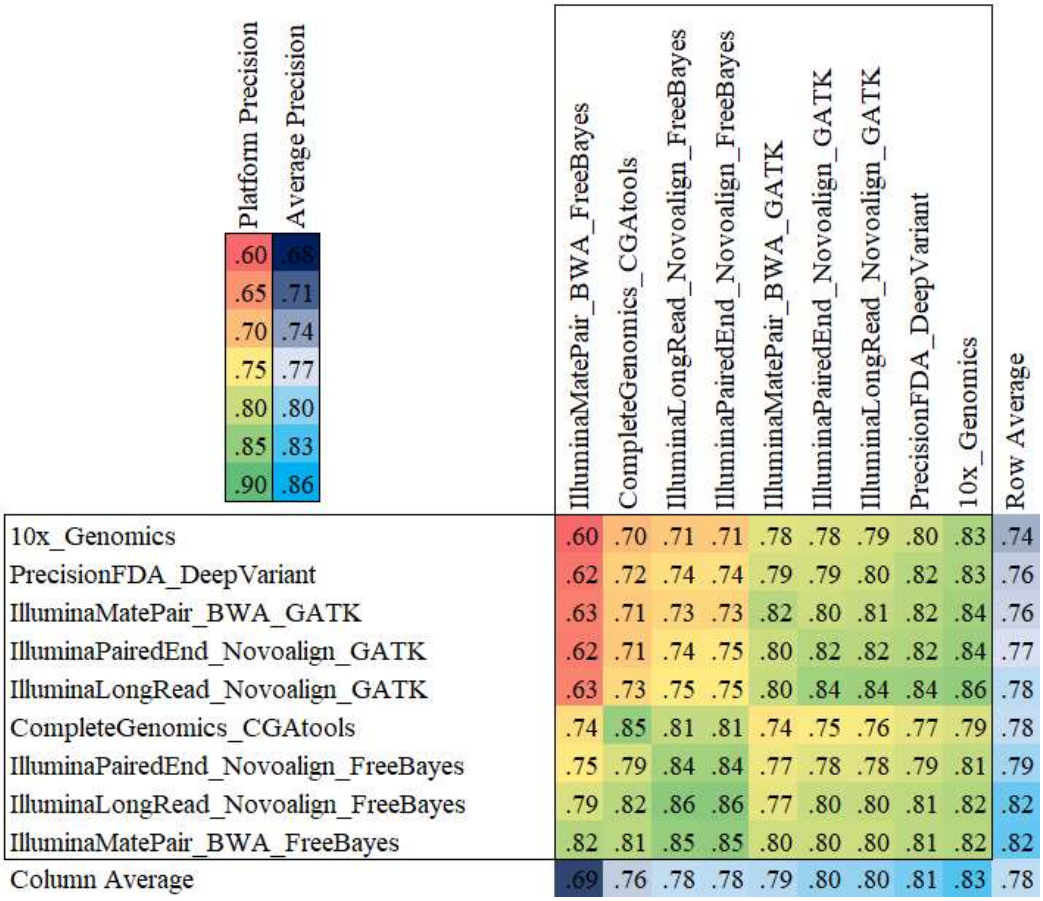
Although we have chosen precision as our performance measure (reason explained in *Methods* and *Discussion* section), the precision-recall curve of ***Figure 3*** shows that our filtering scheme has demonstrated high AUC or area under curve (0.965) for Complete Genomics platform. Since, AUC is a measure of ability to discriminate true and false indel calls, our filtering scheme can be used as an effective classifier. ***Supplementary Table 7*** corroborates this assumption as we can see good F1 scores (harmonic mean of precision and recall) over a range of threshold. We have used 0.5 as our threshold score which gives the highest F1 score (0.91). We can adjust the threshold according to our preference of precision and recall.

Cross-platform Performance

Our platform-specific filtering scheme can be extended to the cross-platform scenario as well. We have trained nine different models from training callsets generated from nine different

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

platforms (see ***Figure 4***). We have evaluated each of these trained models over nine different test callsets generated from the same nine platforms. The resulting matrix is shown as a heatmap in ***Figure 4***. Our first observation is that a model trained from a callset generated from a platform does not always perform the best on the test callset generated from the same platform.



	IlluminaMatePair_BWA_FreeBayes	CompleteGenomics_CGAtools	IlluminaLongRead_Novoalign_FreeBayes	IlluminaPairedEnd_Novoalign_FreeBayes	IlluminaMatePair_BWA_GATK	IlluminaPairedEnd_Novoalign_GATK	IlluminaLongRead_Novoalign_GATK	PrecisionFDA_DeepVariant	10x_Genomics	Row Average
10x_Genomics	.60	.70	.71	.71	.78	.78	.79	.80	.83	.74
PrecisionFDA_DeepVariant	.62	.72	.74	.74	.79	.79	.80	.82	.83	.76
IlluminaMatePair_BWA_GATK	.63	.71	.73	.73	.82	.80	.81	.82	.84	.76
IlluminaPairedEnd_Novoalign_GATK	.62	.71	.74	.75	.80	.82	.82	.82	.84	.77
IlluminaLongRead_Novoalign_GATK	.63	.73	.75	.75	.80	.84	.84	.84	.86	.78
CompleteGenomics_CGAtools	.74	.85	.81	.81	.74	.75	.76	.77	.79	.78
IlluminaPairedEnd_Novoalign_FreeBayes	.75	.79	.84	.84	.77	.78	.78	.79	.81	.79
IlluminaLongRead_Novoalign_FreeBayes	.79	.82	.86	.86	.77	.80	.80	.81	.82	.82
IlluminaMatePair_BWA_FreeBayes	.82	.81	.85	.85	.80	.80	.80	.81	.82	.82
Column Average	.69	.76	.78	.78	.79	.80	.80	.81	.83	.78

Figure 4: Cross Platform Precision Heatmap. Rows are Platforms Generating Training Indel Calls for Sample HG001/HG005 and Columns are Platforms Generating Test Indel Calls for Sample HG002

For example, the model trained using callset generated by 10x_Genomics achieves 83% precision over the test callset generated by 10x_Genomics, whereas for the same test callset, model trained from data generated by Illumina (long read) with GATK achieves 86% precision. On the contrary, the best precision (85%) over Complete Genomics test callset is achieved by the model trained with Complete Genomics training callset.

Our second observation is that there are platforms whose generated callsets are hard to predict unless the model is trained with callsets generated from similar platform, e.g., Illumina (mate pair) with FreeBayes caller. On the other hand, there are platforms, whose generated callsets can be used to train models that can perform well over callsets produced from diverse platforms, e.g., Illumina (long read) with FreeBayes caller. So, these universal filters show the promise to be used in case we don't have information about the data generating platform. An interesting

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

observation is that if the train and test callsets share a common variant caller, e.g. FreeBayes, GATK, the corresponding model is likely to perform well. The greenish rectangle at the bottom-left corner of the matrix is created due to FreeBayes. Our cross-platform study demonstrates that using filter trained from any model will improve precision. Although some models will perform better than others, it is always better to use a filter than no filter.

Feature Importance

We have grouped our selected genomic features used for classification into 10 feature groups according to their categorical similarities. XGBoost provides a score for each feature which indicates how useful the feature has been in the construction of the boosted decision trees within the model. By dividing each score with the maximum score, we have defined a measure named *relative importance*. For each group, we have calculated mean and standard deviation of the *relative importance* from the individual *relative importance* of member features (see ***Figure 5***).

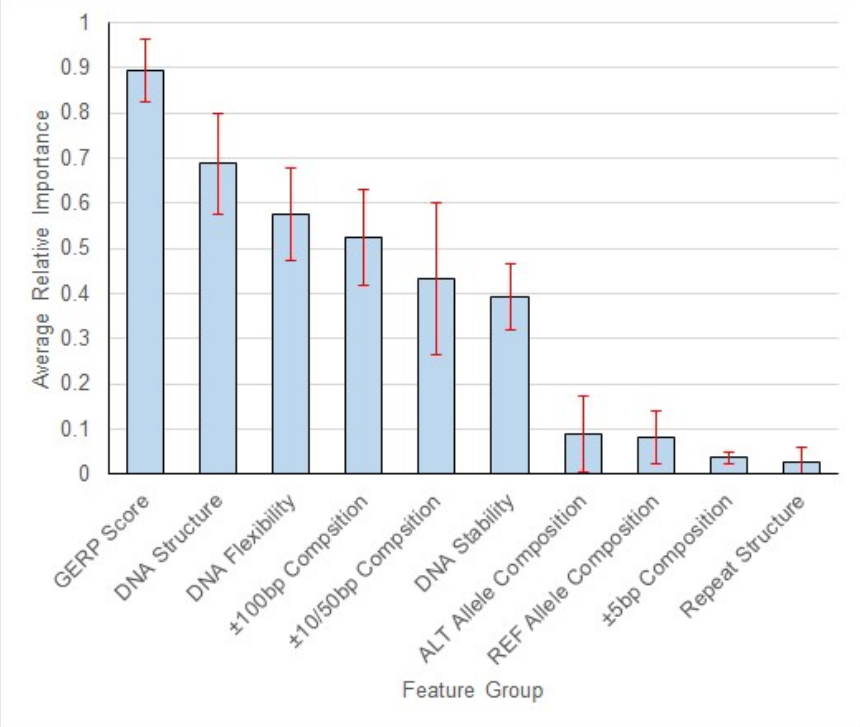


Figure 5: Mean and Standard Deviation (Red Error Bars) of Relative Feature Importance for Different Feature Groups.

We can see that the level of conservation (±10bp GERP score) is one of the best indicators of false indel calls. A feature is particularly useful in discriminating if its distribution differs between true and false positives. We have demonstrated that for 1bp downstream GERP score (see ***Figure 6***), true positives have a heavier-tailed distribution than false positives. Local

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

physical DNA structure and flexibility are also good predictors. Longer genomic contexts, e.g., ±100bp distribution have better predictivity than shorter ones, e.g., ±5bp composition. An interesting observation is that the composition of reference or alternative allele has less importance than the composition of surrounding contexts in discrimination of true and false indel calls. A description of features is given in ***Supplementary Table 2***. and the ranking of features according to their importance is given in ***Supplementary Figure 4***.

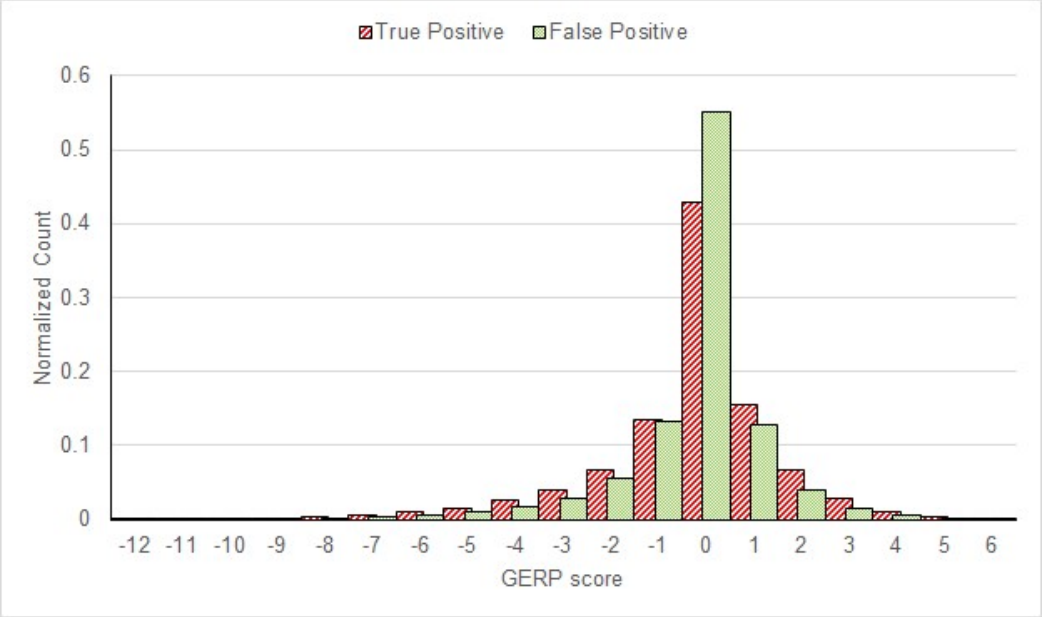


Figure 6: Histogram of 1bp Downstream GERP Scores of True Indel Calls (Red) and False Indel Calls (Green) from Complete Genomics Platform for Sample HG002

Discussion

Identification of true indel calls is of prime importance in many personal genomics applications. Due to the proliferation of next-generation sequencing, we have large repositories of indel calls at our disposals. With a few exceptions, these datasets do not come with detailed publicly available sequencing read information. To extract reliable indel calls from these large-scale data, our sequencing-read-independent filter should be advantageous. Our selected features can be computed from publicly available genome annotations.

Since, we are more interested in filtering false positives from called indels rather than identification of every rare indel call, we have reported precision as our primary performance measurement. However, our training algorithm minimize a loss function for binary logistic regression which does not prioritize precision in any way. We have provided all standard classification performance measures, i.e., recall, F1-score and area under precision-recall curve where appropriate. We have demonstrated that our filtering scheme has done well in all criteria; our choice of threshold has achieved the highest F1-score (.91) with an AUC > 0.965 for Complete Genomics platform. Our reported recall estimates the fraction of platform-supported

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

true indels (only those called by platform) passed by our filter as positive; this does not estimate the fraction of all GIAB-supported true indels (the gold standard, some of which are not called by the platform) passed by our filter as positives. This makes sense, as our filter will not get a chance to evaluate an indel not called by the platform. Thus, we cannot know the true recall of our filter, and this is another reason to use precision as our metric.

We have demonstrated the efficacy of GERP scores in recognition of true indel calls. ***Figure 6*** shows a histogram of 1bp downstream GERP score for both true and false indel calls. Indel calls around the region with neutral GERP scores exhibit a strong bias for false indel calls. One possible explanation of this phenomenon is that regions under no selective pressure can easily accumulate repetitive elements (e.g., homopolymer, tandem repeats). Sequencing of these regions are inherently difficult and error-prone.

Our training and validation data consist of gold-standard indel calls from both high and non-high confidence region (NHCR). Since we do not have a complete map of NHCR indels for gold-standard samples, most of the performance evaluation tasks reported in the literature have been conducted using only indels from high confidence regions (e.g., PrecisionFDA truth challenge[5]). In our opinion, this approach has two drawbacks. First, density of false indels calls are much higher (10% of the genome harbours greater than 50% of false indel calls) in NHCR, despite the fact that a fraction of these NHCR indel calls (in our estimate around 3%, see ***Supplementary Table 9***) is expected to be true. Second, for a new sample we do not know exactly which portions of the genome are high confidence regions. Thus, discarding NHCR indel calls beforehand is not a feasible option. We argue that the inclusion of NHCR indel calls in our training set improves our model's capability of false indel call detection without putting us in any advantageous situation with respect to precision, our principal performance metric. Rather our reported precision can be seen as an underestimation of true precision (which could be calculated if we would know the complete NHCR indel map), since our training set is more biased towards type II error (from the viewpoint of statistical hypothesis testing) or false negative (missing true indel calls).

We have observed that the performance of our classifier is consistently better in the context of different train-test sample than in the context of same sample train-test splits. We hypothesize that in case of a different train-test samples, a set of indel calls are common in both samples. In actual scenario, we are most likely to apply the classifier across samples and get that additional performance boost.

One of the current tides in genomic data analysis and in the broader area of machine learning is the extensive application of deep neural network architecture. We have previously mentioned DeepVariant which shows promising results for variant calling. The biggest downside of deep

[5] https://precision.fda.gov/challenges/truth/results

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

architectures is the difficulty of model interpretation within the black-box, although researchers are actively working on this issue (Shrikumar et al. 2017). Unlike other areas, interpretability is very important in biology, as this lay the foundation of the broader understanding about living systems. Another downside is the existence of adversarial examples, a sample with marginal noise which will be disproportionately misclassified by the model. Requirement of large training data (with no clearly reliable techniques for genomic data augmentation), high-end GPU platform, complicated hyperparameter tuning add up to the complexity. In our context, our use of gradient tree boosting directly gives feature importance. Moreover, our annotation data is mostly static (as we don't use read data and model particular error pattern) and will not vary with more samples. Rather, more training data will saturate our understanding about unreliable genomic contexts corresponding to a particular sequencing platform.

A potential limitation of our filter is that it may have less sensitivity towards rare or de-novo variants. Discrimination between a sequencing error and a rare variant is an inherently complicated task and our filter also suffers in this context, more so since we don't use sequencing reads. However, at a bare minimum, it can provide a prior probability of being a true indel to be used with other evidences for a hard to validate indel.

Methods

We have formalized the identification of false indel calls as a supervised binary classification problem. Our basic assumption is that the genomic context of a true indel callset and that of a false indel callset differs from each other. Decision tree is an effective model for classification problems. Ensembles of decision trees are some of the most powerful off-the-shelf classifiers available to data analysts (Friedman et al. 2000). We have particularly used XGBoost (Chen and Guestrin 2016), an implementation of decision tree-based gradient boosting classifier. XGBoost provides a verdict on being a true indel as well as an estimate of uncertainty with a probability score. For training and testing, we have used genome in a bottle (GIAB) data set. To train a binary classifier we need instances from both classes, i.e., true and false indel calls, annotated with relevant genomic contexts captured within a feature matrix. Our positive instances have come from the indel callset generated by the corresponding platform and supported by GIAB high-quality gold standard dataset. The generated indel callset not present in the gold standard dataset are our negative instances for corresponding platform. We have annotated both positive and negative instances using publicly available genomic features, not related to any specific platform or caller or population.

Dataset

We have used both platform-specific indel calls and the gold-standard indel calls from GIAB (Genome-in-a-bottle) consortium. In our primary evaluation, for training purposes we have used indel callsets of three samples, i.e., HG003, HG004 and HG005, generated from five

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

different sequencing pipelines (see ***Figure 1***). For some platforms, indel callset of only one training sample (HG001) is available which we have used to evaluate them (see ***Supplementary Table 4***). For testing purposes, we have always used sample HG002, as this was sequenced by each platform. To assess the effect of the sample size over performance, for training purposes we have added indel callsets from four samples generated from Complete Genomics, namely, HG005, HG004, HG003 and HG001, in the given order. For annotation purposes, we have used RepeatMasker (Tempel 2012) for identifying various repeat regions, e.g., tandem, interspersed etc.. To measure the level of conservation, we have used GERP scores (Davydov et al. 2010). All DNA sequence contexts are calculated with hg19 reference genome (Rosenbloom et al. 2015).

Feature Selection

Extracting representative features is a crucial step for building any machine learning classifier. Here the goal is to select features which are relevant predictors for sample class, e.g., true vs false calls. We have tried a number of genome annotations as well as their combinations to find a good set of platform-independent genomic features. We have not used any platform or population specific features, such as, read depth, quality, alignment, allele frequency, genotype calls, etc. All of the features can be generated with publicly available genome sequences and annotations.

We have found that functional features, e.g., gene annotation from GENCODE (Harrow et al. 2012), open chromatin, binding sites, histone modification etc. from ENCODE (Bernstein et al. 2012) are less indicative of indel calling accuracy; instead physical properties (Thomas Abeel 2011), nucleotide composition (distribution of monomer, dimer and trimer), level of conservation (GERP scores) (Davydov et al. 2010) and repeat structure of the indel and its surrounding DNA are more suggestive. We have calculated these features for four corresponding contexts of different scales, namely, ±5bp, ±10bp, ±50bp and ±100bp. In total, we have kept 219 features which are found to discriminate true indels from false ones.

The Model

Boosting is an ensemble technique in machine learning that combines several weak learners to build a strong learner (Friedman et al. 2000). We can formulate boosting using a general class of mathematical model, i.e., the adaptive basis function model. Here, for a given dataset $\mathbf{x}$, a target function $f(\mathbf{x})$ is expressed as a linear combination of d basis functions as in the following equation:

$$f(\mathbf{x}) = \sum_{k=0}^{d} w_k \phi_k(\mathbf{x}) \quad (1)$$

Here, $\phi_k(\mathbf{x})$ is the k^{th} basis function, i.e., weak learner/classifier, and w_k is the assigned weight of $\phi_k(\mathbf{x})$ in decision making. In gradient tree boosting, each $\phi_k(\mathbf{x})$ is a decision tree classifier.

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Behind most supervised learning algorithm, the assumption is that optimal model parameters will minimize the risk of misclassification. True risk of misclassification is approximated through the introduction of some specified per-sample loss function $\ell\big(y_i, f(\mathbf{x}_i)\big)$ for the i^{th} sample, where y_i is the true class label. A general stage-wise boosting algorithm defines a recursive relation between the ensemble learner from stage k and $k-1$ for the i^{th} sample as follows:

$$f(\mathbf{x}_i)^k = f(\mathbf{x}_i)^{(k-1)} + \beta_k \phi_k(\mathbf{x}_i) \tag{2}$$

Learning (β_k, ϕ_k) can be cast as an optimization problem which tries to minimize the risk according to following equation:

$$(\beta_k, \phi_k) = \arg\min_{\beta,\phi} \sum_{i=1}^{n} \ell\big(y_i, f(\mathbf{x}_i)^{(k-1)} + \beta\phi(\mathbf{x}_i)\big)$$

Gradient boosting splits this single step optimization into two parts. First it fits $\phi_k(\mathbf{x}_i)$ to a new quantity called pseudo-residuals r_{ik} where,

$$r_{ik} = -\frac{\partial}{\partial f(\mathbf{x}_i)^{(k-1)}} \ell\big(y_i, f(\mathbf{x}_i)^{(k-1)}\big) \tag{3}$$

In fact, r_{ik} is the negative of the gradient of loss function for the i^{th} sample with respect to $f(\mathbf{x}_i)^{(k-1)}$ from stage $k-1$. This is an example of functional gradient descent giving the direction of steepest descent in the space of loss function. The amount of total descent or the weight, β_k of the basis function $\phi_k(\mathbf{x}_i)$ is calculated using the following equation:

$$\beta_k = \arg\min_{\beta} \sum_{i=1}^{n} \ell\left(y_i, f(\mathbf{x}_i)^{(k-1)} + \beta\phi_k(\mathbf{x}_i)\right) \tag{4}$$

Sometimes, instead of Equation 1 we use the following recursive equation:

$$f(\mathbf{x})^{(k)} = f(\mathbf{x})^{(k-1)} + \nu\beta_k\phi_k(\mathbf{x}) \tag{5}$$

Here, ν is called a shrinkage factor which is used to slow down stage wise model building to avoid overfitting.

In our experiments, we have used XGBoost (Chen and Guestrin 2016), an efficient, scalable, sparsity-aware, cache-oblivious, distributed implementation of gradient tree boosting. It also uses second-order Taylor series approximation of a regularized loss function for fast computation.

Decision tree-based learning models provide a natural framework for feature importance estimation. Gradient tree boosting extends the framework for an ensemble of decision trees. Since the importance of each feature is calculated separately without being entangled with other features, we can sort the relative importance of features and provide a global ranking. At every non-leaf node of a decision tree, a feature with an appropriate value (splitting point) is selected which will reduce the node impurity (mixture of samples from different classes) maximally. For each feature, the total weighted reduction across all the nodes represent its importance for

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

corresponding decision tree. For an ensemble of trees, we can estimate the actual feature importance by averaging the feature importance from individual decision trees (Friedman 2001). We have used the feature score provided by XGBoost (Friedman et al. 2000) to assess the relative importance of used features.

Performance Evaluation

The stakes associated with decision errors by classifiers are not always symmetric across possible class labels. Using our tool, later we have plan to build a repository of high quality indels enriched with true positives (TP). To achieve this, we have been more interested on the reduction of false positive (FP) calls than that of false negative (FN) calls. Thus, we have reported precision or positive predictive value (PPV) as our main performance metric which is defined as follows: $PPV = \frac{TP}{TP+F}$

For every sequencing pipeline of a particular platform, an aligner and a variant caller mentioned in ***Supplementary Table 1***, we have calculated a base precision, intersection (using RTG tools (Cleary et al. 2015)) between indel callset generated by the sequencing pipeline and the gold standard GIAB indel callset. We have applied our classifier in two different settings. First, the classifier is trained using the gold standard indel calls (all or only platform generated) from one sample and tested on the indel callset of another sample produced by that particular pipeline. Unless otherwise specified, this is the default settings. Second, we have also performed train-test split using the pipeline generated indel callsets for each sample to discard the effect of common instances between the training and test samples. This was only done (c.f., the analysis of ***Figure 2***) to understand the classifier performance under completely randomized and controlled scenario. But this is not likely to be an actual application scenario.

Tuning Hyperparameters

We have performed a grid search in the hyperparameters' space for the fine tuning of our classifier. Three most important hyperparameters are:

1. Number of trees: this is the number of boosting rounds.
2. Tree depth: the maximum number of conditions checking from the root to a leaf
3. Learning rates: shrinkage factor for a new tree

Number of trees or boosting rounds for our gradient boosting machine is one of the most important predefined hyperparameters. ***Supplementary Figure 2*** shows the effect of increasing number of trees over performance. Evidently, initial boosting rounds improve performance drastically whereas later rounds show performance saturation.

Other two significant hyperparameters are learning rate and tree depth. ***Supplementary Figure 3*** explores their mutual effect over performance. Increasing tree depth generally boosts performance. But similar to number of trees, the performance increment saturates with increasing tree depth. ***Supplementary Figure 3*** shows that influence of learning rates and tree

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

depth on performance is intertwined. Low learning rates slow down the performance saturation. So, attaining a high-performance mark becomes possible. Higher learning rates tend to saturate performance quickly.

Chosen values for our experiments are as follows, boosting round = 200, tree depth = 16 and learning rate = 0.25.

Conclusions

Our findings in this paper should help researchers to identify a more reliable set of indels. This study also sheds lights on the genomic contexts likely to be responsible for interesting mutation events. We have future plans to use our variant filter to extract more complicated class of variants, e.g., multi-allelic indels. This study should also help to elucidate mutation history.

References

Albers CA, Lunter G, MacArthur DG, McVean G, Ouwehand WH, Durbin R. 2011. Dindel: accurate indel calls from short-read data. *Genome Res* **21**: 961–73. http://www.ncbi.nlm.nih.gov/pubmed/20980555 (Accessed June 13, 2017).

Alkan C, Coe BP, Eichler EE. 2011. Genome structural variation discovery and genotyping. *Nat Rev Genet* **12**: 363–376. http://www.ncbi.nlm.nih.gov/pubmed/21358748 (Accessed June 13, 2017).

Au CH, Leung AYH, Kwong A, Chan TL, Ma ESK. 2017. INDELseek: detection of complex insertions and deletions from next-generation sequencing data. *BMC Genomics* **18**: 16. http://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-016-3449-9 (Accessed June 13, 2017).

Bernstein BE, Birney E, Dunham I, Green ED, Gunter C, Snyder M. 2012. An integrated encyclopedia of DNA elements in the human genome. *Nature* **489**: 57–74. http://dx.doi.org/10.1038/nature11247 (Accessed July 9, 2014).

Chen T, Guestrin C. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pp. 785–794, ACM Press, New York, New York, USA http://dl.acm.org/citation.cfm?doid=2939672.2939785 (Accessed June 14, 2017).

Cheung VG, Spielman RS. 2009. Genetics of human gene expression: mapping DNA variants that influence gene expression. *Nat Rev Genet* **10**: 595–604. http://www.ncbi.nlm.nih.gov/pubmed/19636342 (Accessed June 14, 2017).

Chiara M, Pesole G, Horner DS. 2012. SVM^2: an improved paired-end-based tool for the detection of small genomic structural variations using high-throughput single-genome resequencing data. *Nucleic Acids Res* **40**: e145. http://www.ncbi.nlm.nih.gov/pubmed/22735696 (Accessed June 14, 2017).

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Chu C, Zhang J, Wu Y. 2014. GINDEL: Accurate Genotype Calling of Insertions and Deletions from Low Coverage Population Sequence Reads ed. R. Belshaw. *PLoS One* **9**: e113324. http://www.ncbi.nlm.nih.gov/pubmed/25423315 (Accessed June 13, 2017).

Cleary JG, Braithwaite R, Gaastra K, Hilbush BS, Inglis S, Irvine SA, Jackson A, Littin R, Rathod M, Ware D, et al. 2015. Comparing Variant Call Files for Performance Benchmarking of Next-Generation Sequencing Variant Calling Pipelines. *bioRxiv* 023754. https://www.biorxiv.org/content/10.1101/023754v2 (Accessed April 5, 2019).

Cornish A, Guda C. 2015. A Comparison of Variant Calling Pipelines Using Genome in a Bottle as a Reference. *Biomed Res Int* **2015**: 1–11. http://www.hindawi.com/journals/bmri/2015/456479/ (Accessed June 14, 2017).

Davydov E V, Goode DL, Sirota M, Cooper GM, Sidow A, Batzoglou S. 2010. Identifying a high fraction of the human genome to be under selective constraint using GERP++. *PLoS Comput Biol* **6**: e1001025. http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1001025 (Accessed September 27, 2015).

Emde A-K, Schulz MH, Weese D, Sun R, Vingron M, Kalscheuer VM, Haas SA, Reinert K. 2012. Detecting genomic indel variants with exact breakpoints in single- and paired-end sequencing data using SplazerS. *Bioinformatics* **28**: 619–627. http://www.ncbi.nlm.nih.gov/pubmed/22238266 (Accessed June 13, 2017).

Fang H, Bergmann EA, Arora K, Vacic V, Zody MC, Iossifov I, O'Rawe JA, Wu Y, Jimenez Barron LT, Rosenbaum J, et al. 2016. Indel variant analysis of short-read sequencing data with Scalpel. *Nat Protoc* **11**: 2529–2548. http://www.ncbi.nlm.nih.gov/pubmed/27854363 (Accessed June 13, 2017).

Fang LT, Afshar PT, Chhibber A, Mohiyuddin M, Fan Y, Mu JC, Gibeling G, Barr S, Asadi NB, Gerstein MB, et al. 2015. An ensemble approach to accurately detect somatic mutations using SomaticSeq. *Genome Biol* **16**: 197. http://www.ncbi.nlm.nih.gov/pubmed/26381235 (Accessed June 13, 2017).

Friedman J, Hastie T, Tibshirani R. 2000. Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors). *Ann Stat* **28**: 337–407. http://projecteuclid.org/euclid.aos/1016218223 (Accessed June 15, 2017).

Friedman JH. 2001. Greedy function approximation: A gradient boosting machine. *Ann Stat* **29**: 1189–1232. http://projecteuclid.org/euclid.aos/1013203451 (Accessed July 27, 2017).

Garrison E, Marth G. 2012. Haplotype-based variant detection from short-read sequencing. http://arxiv.org/abs/1207.3907 (Accessed March 6, 2019).

Harrow J, Frankish A, Gonzalez JM, Tapanari E, Diekhans M, Kokocinski F, Aken BL, Barrell D, Zadissa A, Searle S, et al. 2012. GENCODE: The reference human genome

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

annotation for the ENCODE project. *Genome Res* **22**: 1760–1774.

Hasan MS, Wu X, Zhang L. 2015. Performance evaluation of indel calling tools using real short-read data. *Hum Genomics* **9**: 20. http://www.ncbi.nlm.nih.gov/pubmed/26286629 (Accessed June 13, 2017).

Hwang K-B, Lee I-H, Park J-H, Hambuch T, Choe Y, Kim M, Lee K, Song T, Neu MB, Gupta N, et al. 2014. Reducing False-Positive Incidental Findings with Ensemble Genotyping and Logistic Regression Based Variant Filtering Methods. *Hum Mutat* **35**: 936–944. http://doi.wiley.com/10.1002/humu.22587 (Accessed March 4, 2019).

Kanehisa M, Sato Y, Kawashima M, Furumichi M, Tanabe M. 2015. KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Res*. http://www.ncbi.nlm.nih.gov/pubmed/26476454 (Accessed October 21, 2015).

Koboldt DC, Chen K, Wylie T, Larson DE, McLellan MD, Mardis ER, Weinstock GM, Wilson RK, Ding L. 2009. VarScan: variant detection in massively parallel sequencing of individual and pooled samples. *Bioinformatics* **25**: 2283–2285. http://www.ncbi.nlm.nih.gov/pubmed/19542151 (Accessed June 13, 2017).

Koboldt DC, Steinberg KM, Larson DE, Wilson RK, Mardis ER. 2013. The next-generation sequencing revolution and its impact on genomics. *Cell* **155**: 27–38. http://www.ncbi.nlm.nih.gov/pubmed/24074859 (Accessed March 4, 2019).

Lai Z, Markovets A, Ahdesmaki M, Chapman B, Hofmann O, McEwen R, Johnson J, Dougherty B, Barrett JC, Dry JR. 2016. VarDict: a novel and versatile variant caller for next-generation sequencing in cancer research. *Nucleic Acids Res* **44**: e108. http://www.ncbi.nlm.nih.gov/pubmed/27060149 (Accessed March 26, 2019).

Lek M, Karczewski KJ, Minikel E V., Samocha KE, Banks E, Fennell T, O'Donnell-Luria AH, Ware JS, Hill AJ, Cummings BB, et al. 2016. Analysis of protein-coding genetic variation in 60,706 humans. *Nature* **536**: 285–291. http://www.nature.com/doifinder/10.1038/nature19057 (Accessed June 15, 2017).

Li H. 2011. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics* **27**: 2987–93. http://www.ncbi.nlm.nih.gov/pubmed/21903627 (Accessed June 13, 2017).

Li H. 2013. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. http://arxiv.org/abs/1303.3997 (Accessed March 6, 2019).

Li S, Li R, Li H, Lu J, Li Y, Bolund L, Schierup MH, Wang J. 2013. SOAPindel: Efficient identification of indels from short paired reads. *Genome Res* **23**: 195–200. http://www.ncbi.nlm.nih.gov/pubmed/22972939 (Accessed June 13, 2017).

Liu Y, Li B, Tan R, Zhu X, Wang Y. 2014. A gradient-boosting approach for filtering de novo mutations in parent-offspring trios. *Bioinformatics* **30**: 1830–6. http://www.ncbi.nlm.nih.gov/pubmed/24618463 (Accessed June 13, 2017).

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Luo R, Sedlazeck FJ, Lam T-W, Schatz MC. 2019. A multi-task convolutional deep neural network for variant calling in single molecule sequencing. *Nat Commun* **10**: 998. http://www.nature.com/articles/s41467-019-09025-z (Accessed March 26, 2019).

McCarthy S, Das S, Kretzschmar W, Durbin R, Abecasis G, Marchini J. 2015. A reference panel of 64,976 haplotypes for genotype imputation. *bioRxiv*. http://biorxiv.org/content/early/2015/12/23/035170.abstract.

McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M, et al. 2010a. The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res* **20**: 1297–303. http://www.ncbi.nlm.nih.gov/pubmed/20644199 (Accessed June 13, 2017).

McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M, et al. 2010b. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res* **20**: 1297–1303. http://www.ncbi.nlm.nih.gov/pubmed/20644199 (Accessed March 6, 2019).

Michaelson JJ, Sebat J. 2012. forestSV: structural variant discovery through statistical learning. *Nat Methods* **9**: 819–821. http://www.ncbi.nlm.nih.gov/pubmed/22751202 (Accessed June 14, 2017).

Mills RE, Luttig CT, Larkins CE, Beauchamp A, Tsui C, Pittard WS, Devine SE. 2006. An initial map of insertion and deletion (INDEL) variation in the human genome. *Genome Res* **16**: 1182–1190. http://www.ncbi.nlm.nih.gov/pubmed/16902084 (Accessed June 14, 2017).

Mullaney JM, Mills RE, Stephen Pittard W, Devine SE. 2010. Small insertions and deletions (INDELs) in human genomes. *Hum Mol Genet* **19**.

Parikh H, Mohiyuddin M, Lam HYK, Iyer H, Chen D, Pratt M, Bartha G, Spies N, Losert W, Zook JM, et al. 2016. svclassify: a method to establish benchmark structural variant calls. *BMC Genomics* **17**: 64. http://www.ncbi.nlm.nih.gov/pubmed/26772178 (Accessed June 14, 2017).

Peters BA, Kermani BG, Sparks AB, Alferov O, Hong P, Alexeev A, Jiang Y, Dahl F, Tang YT, Haas J, et al. 2012. Accurate whole-genome sequencing and haplotyping from 10 to 20 human cells. *Nature* **487**: 190–195. http://www.nature.com/articles/nature11236 (Accessed March 6, 2019).

Poplin R, Chang P-C, Alexander D, Schwartz S, Colthurst T, Ku A, Newburger D, Dijamco J, Nguyen N, Afshar PT, et al. 2018. A universal SNP and small-indel variant caller using deep neural networks. *Nat Biotechnol* **36**: 983. http://www.nature.com/doifinder/10.1038/nbt.4235 (Accessed March 26, 2019).

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Quail MA, Smith M, Coupland P, Otto TD, Harris SR, Connor TR, Bertoni A, Swerdlow HP, Gu Y. 2012. A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers. *BMC Genomics* **13**: 341. http://www.biomedcentral.com/1471-2164/13/341 (Accessed July 9, 2014).

Ramu A, Noordam MJ, Schwartz RS, Wuster A, Hurles ME, Cartwright RA, Conrad DF. 2013. DeNovoGear: de novo indel and point mutation discovery and phasing. *Nat Methods* **10**: 985–987. http://www.ncbi.nlm.nih.gov/pubmed/23975140 (Accessed June 14, 2017).

Ravasio V, Ritelli M, Legati A, Giacopuzzi E. 2018. GARFIELD-NGS: Genomic vARiants FIltering by dEep Learning moDels in NGS ed. B. Berger. *Bioinformatics* **34**: 3038–3040. https://academic.oup.com/bioinformatics/article/34/17/3038/4970515 (Accessed March 3, 2019).

Rimmer A, Phan H, Mathieson I, Iqbal Z, Twigg SRF, WGS500 Consortium W, Wilkie AOM, McVean G, Lunter G. 2014. Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. *Nat Genet* **46**: 912–8. http://www.ncbi.nlm.nih.gov/pubmed/25017105 (Accessed June 14, 2017).

Rosenbloom KR, Armstrong J, Barber GP, Casper J, Clawson H, Diekhans M, Dreszer TR, Fujita PA, Guruvadoo L, Haeussler M, et al. 2015. The UCSC Genome Browser database: 2015 update. *Nucleic Acids Res* **43**: D670–D681.

Sandmann S, de Graaf AO, Karimi M, van der Reijden BA, Hellström-Lindberg E, Jansen JH, Dugas M. 2017. Evaluating Variant Calling Tools for Non-Matched Next-Generation Sequencing Data. *Sci Rep* **7**: 43169. http://www.ncbi.nlm.nih.gov/pubmed/28233799 (Accessed March 26, 2019).

Shrikumar A, Greenside P, Kundaje A. 2017. Learning Important Features Through Propagating Activation Differences. http://arxiv.org/abs/1704.02685 (Accessed April 5, 2019).

Telenti A, Lippert C, Chang P-C, DePristo M. 2018. Deep learning of genomic variation and regulatory network data. *Hum Mol Genet* **27**: R63–R71. http://www.ncbi.nlm.nih.gov/pubmed/29648622 (Accessed March 4, 2019).

Tempel S. 2012. Using and understanding repeatMasker. *Methods Mol Biol* **859**: 29–51.

Thomas Abeel. 2011. Abeel Java Toolkit. http://www.abeel.be/ajt.

Vali U, Brandstrom M, Johansson M, Ellegren H. 2008. Insertion-deletion polymorphisms (indels) as genetic markers in natural populations. *BMC Genet* **9**: 8. http://bmcgenet.biomedcentral.com/articles/10.1186/1471-2156-9-8 (Accessed June 14, 2017).

Wala JA, Bandopadhayay P, Greenwald NF, O'Rourke R, Sharpe T, Stewart C, Schumacher S, Li Y, Weischenfeldt J, Yao X, et al. 2018. SvABA: genome-wide detection of

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

structural variants and indels by local assembly. *Genome Res* **28**: 581–591. http://www.ncbi.nlm.nih.gov/pubmed/29535149 (Accessed April 4, 2019).

Wang Z, Liu X, Yang B-Z, Gelernter J. 2013. The role and challenges of exome sequencing in studies of human diseases. *Front Genet* **4**: 160. http://www.ncbi.nlm.nih.gov/pubmed/24032039 (Accessed March 4, 2019).

Wilm A, Aw PPK, Bertrand D, Yeo GHT, Ong SH, Wong CH, Khor CC, Petric R, Hibberd ML, Nagarajan N. 2012. LoFreq: a sequence-quality aware, ultra-sensitive variant caller for uncovering cell-population heterogeneity from high-throughput sequencing datasets. *Nucleic Acids Res* **40**: 11189. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3526318/ (Accessed March 26, 2019).

Yang J, Shi X, Hu L, Luo D, Peng J, Xiong S, Kong F, Liu B, Yuan X. 2016. InDel marker detection by integration of multiple softwares using machine learning techniques. *BMC Bioinformatics* 1–11. http://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-016-1312-2 (Accessed June 13, 2017).

Ye K, Schulz MH, Long Q, Apweiler R, Ning Z. 2009. Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics* **25**: 2865–2871. http://www.ncbi.nlm.nih.gov/pubmed/19561018 (Accessed June 13, 2017).

Zook J, McDaniel J, Parikh H, Heaton H, Irvine SA, Trigg L, Truty R, McLean CY, Vega FMD La, Xiao C, et al. 2018. Reproducible integration of multiple sequencing datasets to form high-confidence SNP, indel, and reference calls for five human genome reference materials. *bioRxiv* 281006. https://www.biorxiv.org/content/10.1101/281006v2 (Accessed March 4, 2019).

Zook JM, Catoe D, McDaniel J, Vang L, Spies N, Sidow A, Weng Z, Liu Y, Mason CE, Alexander N, et al. 2016. Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Sci Data* **3**: 160025. http://www.nature.com/articles/sdata201625 (Accessed January 5, 2017).

Zook JM, Chapman B, Wang J, Mittelman D, Hofmann O, Hide W, Salit M. 2014. Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nat Biotechnol* **32**: 246–251. http://www.nature.com/doifinder/10.1038/nbt.2835 (Accessed June 13, 2017).

Zou J, Huss M, Abid A, Mohammadi P, Torkamani A, Telenti A. 2019. A primer on deep learning in genomics. *Nat Genet* **51**: 12–18. http://www.nature.com/articles/s41588-018-0295-5 (Accessed March 4, 2019).

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Supplemental Materials

Supplementary Table 1: List of sequencing platform and variant caller combinations used in our experiment

Platform	**Caller**
GIAB Consensus	Gold-standard
Complete Genomics	CGATools
10X Genomics	GATK
Illumina Paired End	FreeBayes
Illumina Paired End	GATK
Illumina Mate Pair	FreeBayes
Illumina Mate Pair	GATK
Illumina Long Read	FreeBayes
Illumina Long Read	GATK

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Supplementary Table 2: Selected features used for sample annotation

Type	**Description**
Indel type	Balanced and unbalanced substitution, homopolymer, generalized indel, tandem repeat
Variant length	Distance between reference and variant, affected reference length, alternative allele length
Trimer distribution	The probability distribution of all possible trimers within 100 base pairs upstream and downstream. Reverse complements are considered equivalent which reduced the count by half.
Nucleotide content	A, C, G, T, AT, GC, Purine and Pyrimidine contents
Helical structure	DNA twist, tilt, roll, shift, slide, rise
Stability	Duplex stability free energy, Duplex stability disrupt energy, Denaturation, Stacking energy, Z-DNA, A-philicity
Twist structure	Propeller-twist, protein-DNA twist, b-DNA twist
Flexibility	Bendability, bending stiffness, protein deformation, nucleosome preference
Radical Cleavage Intensity	Dimer radical cleavage intensity, trimer radical cleavage intensity, tetramer radical cleavage intensity, pentamer radical cleavage intensity
Repeat structure	RepeatMasker annotations, e.g., LINE, SINE, LTR, RC, SATELLITE, LOW_COMPLEXITY, SIMPLE_REPEAT, OTHER_REPEAT, UNKNOWN, RNA, rRNA, scRNA, snRNA, srpRNA, tRNA
Conservation	Upstream GERP scores, downstream GERP scores, Average GERP score of affected reference allele

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Supplementary Table 3: Precision (both before and after Filtering) and Recall across Different Platforms and Variant Categories (Training with HG003, HG004, HG005 and Testing with HG002)

Variant	Platform	Filtered Precision	Base Precision	Improvement	Platform Recall
Indel	BWA_FB	0.868890026	0.583043444	0.285846582	0.883360069
	BWA_GATK	0.859497505	0.622203483	0.237294022	0.884426773
	CGAtools	0.896116064	0.702979592	0.193136472	0.938955828
	Novo_FB	0.897032378	0.612719866	0.284312511	0.893288653
	Novo_GATK	0.877508294	0.57158593	0.305922364	0.860690178
Tandem Repeat	BWA_FB	0.940764718	0.812712126	0.128052592	0.980806547
	BWA_GATK	0.937302191	0.800719808	0.136582383	0.973697353
	CGAtools	0.934402393	0.821558788	0.112843604	0.988311802
	Novo_FB	0.94401434	0.804878823	0.139135516	0.976835043
	Novo_GATK	0.942267346	0.796464258	0.145803088	0.97089365
Homopolymer	BWA_FB	0.870649246	0.609115045	0.261534201	0.882427908
	BWA_GATK	0.882210569	0.691475896	0.190734673	0.90676607
	CGAtools	0.885474293	0.689986058	0.195488235	0.919632425
	Novo_FB	0.899932157	0.650754154	0.249178004	0.889525386
	Novo_GATK	0.884976479	0.609146049	0.27583043	0.857741692
Combined	BWA_FB	0.865285106	0.450321038	0.414964068	0.794888386
	BWA_GATK	0.865642257	0.652947091	0.212695166	0.816405359
	CGAtools	0.892443364	0.610070535	0.28237283	0.931938682
	Novo_FB	0.888261042	0.536538358	0.351722684	0.79629738
	Novo_GATK	0.876234053	0.593069812	0.283164241	0.756913691

Supplementary Table 4: Precision (both before and after Filtering) and Recall across Different Platforms (Training with HG001 and Testing with HG002)

Platform	Filtered Precision	Base Precision	Improvement	Platform Recall
10x_Genomics	0.829839207	0.654981899	0.174857308	0.882151456
Complete_Genomics	0.848867008	0.610070535	0.238796474	0.890191933
DeepVariant_PrecisionFDA	0.818655218	0.634213021	0.184442197	0.862526114
IlluminaPairedEnd_FreeBayes	0.838743642	0.530593095	0.308150546	0.878192968
IlluminaPairedEnd_GATK	0.820186934	0.557212315	0.262974618	0.846778272

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Supplementary Table 5: Data for Figure 2 [Left panel]

No of Samples	Platform-TP	GIAB-TP
0	0.610070534578925	0.610070534578925
1	0.849362313633914	0.827872729325533
2	0.87795531017638	0.850080457715
3	0.892443364103996	0.86390142852011
4	0.894835812876568	0.868815875200175

Supplementary Table 6: Data for Figure 2 [Right panel]

Training Sample	Overlapping	Non-overlapping
HG001	0.849266647064217	0.822827813535822
HG003	0.867073369975678	0.836731004348079
HG004	0.871835541071362	0.83580797576253
HG005	0.849364939432526	0.825721263977402

Supplementary Table 7: Data for Figure 3

Threshold	TP	FP	TN	FN	Precision	Recall	F1-score
0.1	298280	95039	97840	3492	0.758366618	0.98842835	0.858247337
0.2	294788	69895	122984	6984	0.808340394	0.9768567	0.884644875
0.3	292501	54564	138315	9271	0.842784493	0.969278131	0.901616277
0.4	289449	44672	148207	12323	0.866299933	0.959164535	0.910370141
0.5	279808	32884	159995	21964	0.894835813	0.927216574	0.910738465
0.6	266042	18404	174475	35730	0.935298791	0.881599353	0.907655514
0.7	257484	12285	180594	44288	0.954461039	0.853240195	0.901016725
0.8	242742	8471	184408	59030	0.966279611	0.804388744	0.877933398
0.9	202766	4263	188616	99006	0.979408682	0.671917872	0.797034597

Supplementary Table 8: Data for Figure 5

Feature	Mean	STD
GERP Score	0.893857254	0.069916296
DNA Structure	0.687149784	0.112285741
DNA Flexibility	0.575843505	0.101948985
±100bp Composition	0.524790685	0.106620855
±10/50bp Composition	0.432841764	0.16987573
DNA Stability	0.392034154	0.073936636
ALT Allele Composition	0.089534154	0.083501268
REF Allele Composition	0.082080024	0.057725063
±5bp Composition	0.036398041	0.012786659
Repeat Structure	0.026875692	0.032858894

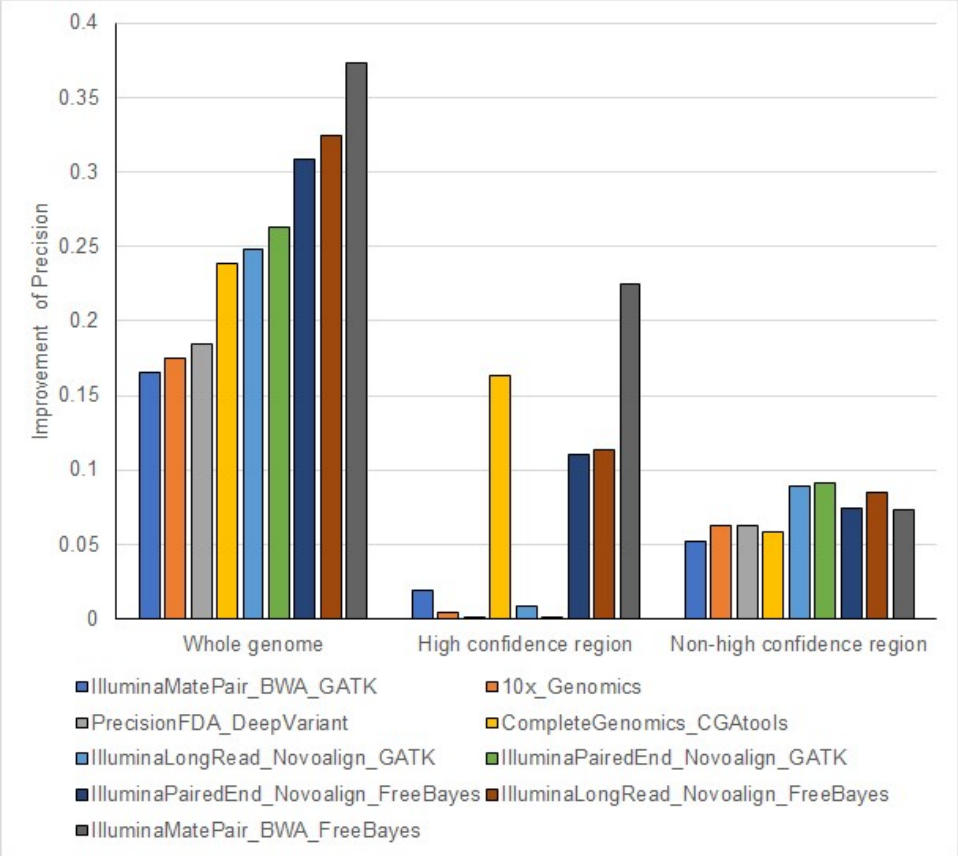
bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

Supplementary Table 9: Calculation of the Fraction of False Indel Calls Made by Complete Genomics Platform in non-High Confidence Regions Expected to be True Figure 3

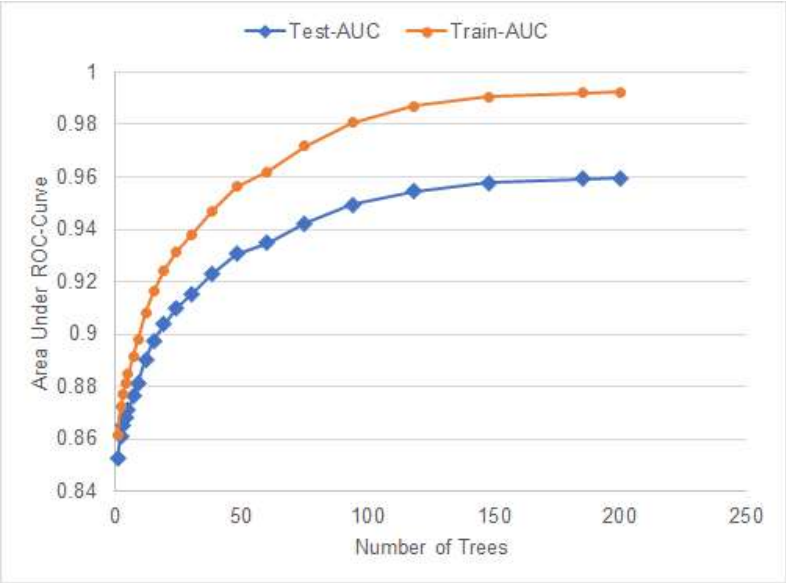
Sample	A	B	C	D	E	F	G	H
HG001	2437910000	432630	29125	14648	6208	3122	120124	0.026
HG002	2358060000	434622	35198	24983	6478	4598	118572	0.039
HG003	2346020000	431171	31470	30752	5928	5793	118051	0.049
HG004	2349000000	437931	32288	30274	6355	5959	125417	0.048
HG005	2376860000	369471	38285	9547	14872	3709	137480	0.027

X =	# of non-N base pairs in the whole reference genome excluding X chromosome = 2684573069
A =	# of non-N base pairs in high-confidence regions (HCR)
B =	# of GIAB gold-standard indels in HCR
C =	# of GIAB gold-standard indels in non-high confidence regions (NHCR)
D =	Expected # of missing indels in NHCR = [X*B/A]-[B+C]
E =	# of GIAB gold-standard indels in NHCR detected by Complete Genomics (CG) platform
F =	Expected # of indels in NHCR supposed to be detected by CG platform
G =	# of false indel calls made by CG platform in NHCR
H =	Fraction of false indel calls made by CG platform in NHCR expected to be true.

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.

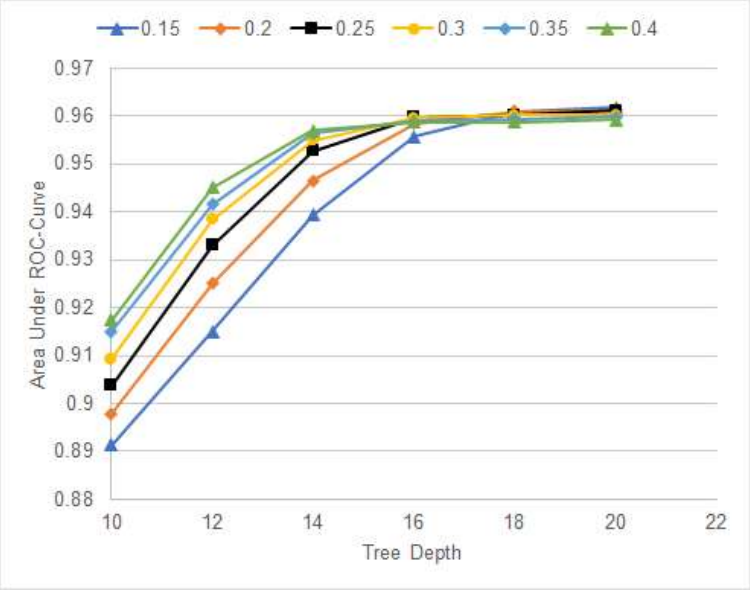


Supplementary Figure 1: Precision Improvement over both Whole Genome and High/non-High Confidence Regions Separately for Different Platforms



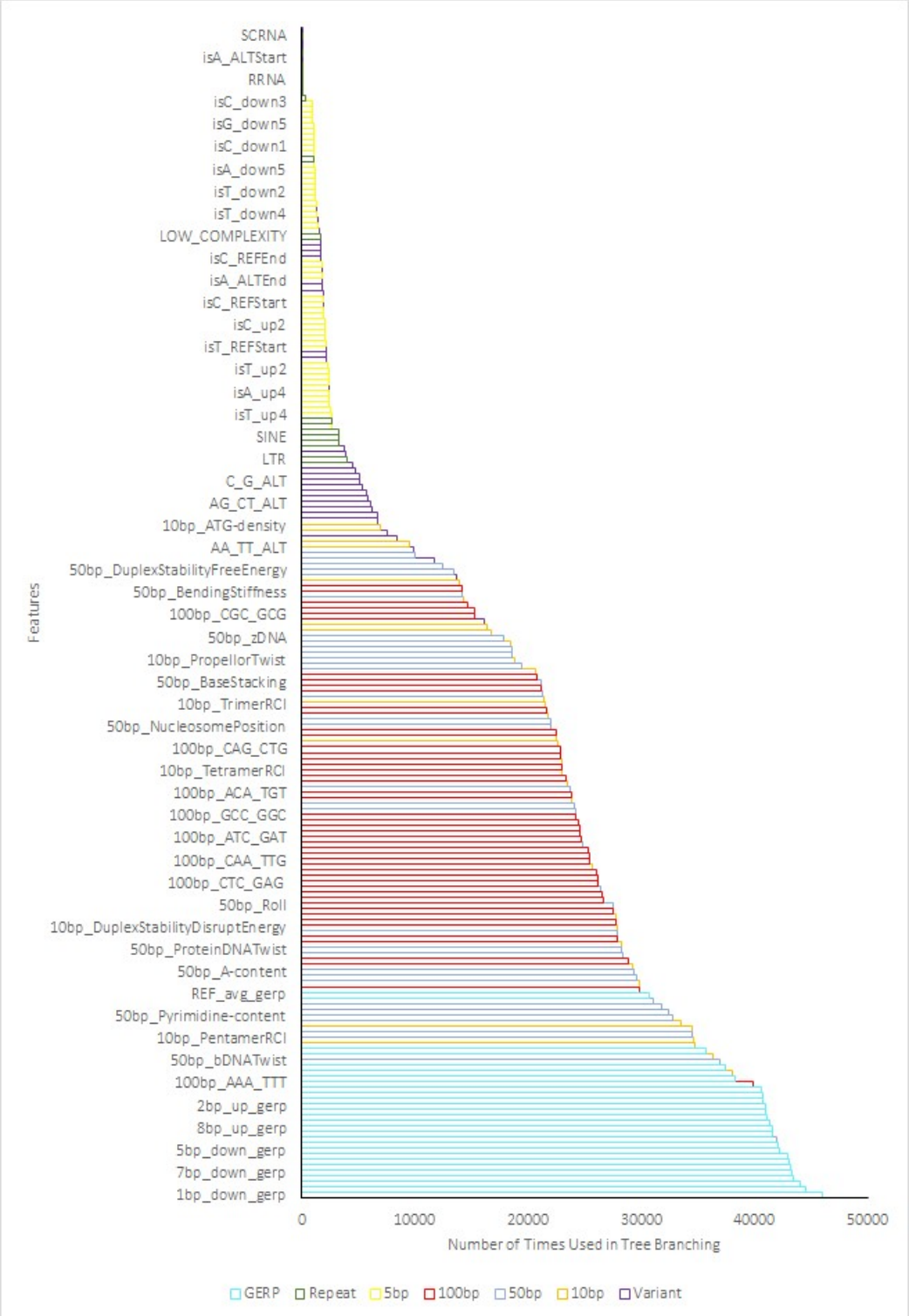
Supplementary Figure 2: Area under ROC curve vs. Number of Boosting Trees

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.



Supplementary Figure 3: Area under ROC curve vs. Tree Depth for Different Learning Rates

bioRxiv preprint doi: https://doi.org/10.1101/601450; this version posted April 7, 2019. The copyright holder for this preprint (which was not certified by peer review) is the author/funder, who has granted bioRxiv a license to display the preprint in perpetuity. It is made available under aCC-BY-NC-ND 4.0 International license.



Supplementary Figure 4: Usage Count of Classifier Features in Sorted Order