Deep Learning Enhances Variant Calling Accuracy in Genomic Data

Bhargava Rathod*

Humphreys University, Stockton, CA 95207, USA
bhargavasai45@gmail.com

Abstract. The accuracy of variant calling in next-generation sequenc- ing (NGS) is critical for genomic research and clinical applications. Tra- ditional variant callers, using diverse methodologies such as haplotype- based, position-based, and pattern growth approaches, often produce discordant results due to their inherent design and statistical methods. This study investigates the use of deep learning to integrate and op- timize information from multiple variant callers. We developed a deep learning neural network designed to improve variant calling accuracy by leveraging features derived from base-specific information, sequencing biases, and quality metrics. The network was optimized through care- ful tuning of hyperparameters, including layer count, optimizer choice, learning rate, and sample balancing techniques. The final architecture included eight layers, utilized the Adam optimizer with a learning rate of 10 minus 5 , and employed SMOTE for sample balancing. Benchmark- ing against both simulated and real datasets demonstrated that the neu- ral network significantly outperforms traditional and concordance-based variant callers. In simulated datasets, the neural network achieved an F1 score of 0.980, surpassing the best single variant caller (0.888) and concordance-based caller (0.927). On a real genomic dataset (NA12878 Genome in a Bottle), the network outperformed existing methods with a precision of 0.859 and a recall of 0.911, leading to a notable reduction in false positives while maintaining high sensitivity. These results validate the efficacy of deep learning in variant calling and highlight its potential to improve precision and recall in genomic analyses, providing a robust tool for both research and clinical genomics.

Keywords: Deep learning, Variant calling, Genomic data

1 Introduction

1.1 Next Generation Sequencing (NGS) for Clinical Genomics

There has been a growing interest in using a patient's genome to guide the di- agnosis and treatment of diseases [2,13], based on the fundamental intuition that variants and mutations in the genome alter gene functions that drive the initiation and progression of the disease. In oncology, for example, identification of the key driver mutations has been shown to be useful in stratifying cancer subtypes [1], and identifying mutations for targeted therapy [3]. Furthermore, the development of next-generation sequencing (NGS) technologies has dramatically reduced sequencing

P. Johri et al. (eds.), *Proceedings of the International Conference on Sustainable Computing and Artificial Intelligence (ICSCAI 2025)*, Advances in Engineering Research 298,
https://doi.org/10.2991/978-94-6239-674-6_33

costs [4,5] enabling the adoption of genomic sequencing in clinical labs.

Although clinical genomics holds great promise, there are still two critical issues that limit its use in a clinical setting. Firstly, it is often difficult to obtain high confidence variant calls from sequencing data, and secondly, the large number of variant calls for patient samples makes interpretation difficult for clinical decision making.

1.2 Variant Calling of NGS Data

Variant calling with NGS data primarily involves the use of various statistical and algorithmic methods to identify variants in the genome. These variants represent the deviations and differences between the genome of interest and a reference human genome. This analysis is non-trivial as each variant call requires the integration of multiple sequence reads (e.g. millions of reads) that contain experimental noise and errors. The calling of variants can be further complicated by errors in mapping the reads to a reference genome.

To account for these errors, variant callers employ a variety of algorithms and statistical models to determine the existence and type of variation/mutation. Because of the differences in assumptions and models employed by different variant callers, certain calling algorithms are more sensitive and accurate in calling specific classes of variants but do not perform well in calling other variant types [12]. To address these problems, ongoing efforts have focused on improving current variant call- ing algorithms, including optimisation of variant calling for different classes of mutations, as well as the reduction in the number of false positive calls [8].

Despite the variety of approaches used for identifying variants and mutations, the accuracy and precision of single variant callers remain suboptimal [4,12].

1.3 Aims and Approach

The overall goal of this project is to address the two major issues limiting the utility of clinical genomics through the following aims:

1. To develop and validate a deep learning network model for improving the accuracy of variant calling.

2. To develop a Bayesian network model for ranking functionally important variants/mutations from high confidence calls identified by the deep learning network.

We describe the development of (i) a deep learning network to identify high-confidence variant calls (focusing on SNVs and short indels) and (ii) a Bayesian network to probabilistically prioritise their functional importance. As a first step, we developed and optimised a deep learning network to identify true variants in both synthetic and real-world datasets. Following the identification of high- confidence variant calls, we built a Bayesian network ranking system based on functional annotations to prioritise mutations and used it to identify functionally important mutations in a cancer sample.

2 Materials and Methods

2.1 Overall Experimental Approach

As a first step in the development of deep learning networks for variant calling, we built two main computational pipelines: (i) a training pipeline for training and the optimization of the neural network, and (ii) an analysis pipeline that uses a trained neural network to perform variant prediction and validation.

In the training pipeline, training datasets from synthetic and real sequencing data were used for performing the processing steps of alignment, variant calling and training of the deep learning network. Briefly, FASTQ sequence reads were first mapped to the reference genome before variant calling was performed using an ensemble of callers. The different variant callers were used to generate the feature vectors that served as the input for the deep learning network. The pre- dictions by the neural network were compared to the ground truth variant calls to train the network to predict high confidence variants.

In the analysis pipeline, the trained and optimized network from the training pipeline is then used to predict high-confidence variant calls in naive samples without ground truth variant calls. In brief, the FASTQ sequence reads are aligned and variant calling performed in a similar fashion as in the training pipeline. The feature vectors from the ensemble of callers is used to predict high confidence calls using the trained and optimized deep learning network.

Finally, we applied the Bayesian network analysis to rank the functionally important variants/mutations from the high confidence calls identified from naive samples in the analysis pipeline.

2.2 Network Optimization and Training Configuration

Next, we systematically optimized and tuned the deep learning neural network to maximize its predictive ability. To do this we focused on four parameters that

are known to be critical in deep learning networks [6, 7], specifically the (i) number of layers; (ii) optimiser choice, (iii) learning rate and (iv) balancing of positive and negative training samples.

Based on these evaluations, we selected an eight-layer merged network architecture trained using the Adam optimizer with a learning rate of 10^{-5}, and applied SMOTE oversampling to address class imbalance.

2.3 Feature Engineering

In order to train a neural network, features in the form of numerical vectors must be used as an input. We subset our features into three broad sets, which are base-specific information, sequencing error and bias information features, and calling and mapping quality. The computation of the features was performed as described below. For an in-depth explanation of their usage and interpretation.

Sequencing Biases and Errors

GC content. This feature comprises the GC content of the reference genome for at least ten bases around the mutation site.

Longest homozygous run. This feature comprises the longest similar string of bases in the reference genome, for at least ten bases around the mutation site.

Allele Count and Allele Balance. This feature is an output from Haplotype Caller and Unified Genotyper, and describes the total number of alleles contributing to a call and the balance between reference and alternate alleles reads.

Calling and Mapping Qualities

Genotype Likelihood. The genotype likelihood score provides the Phred-scaled likelihood scores of how confident the caller is in determining that it is a homozygous or heterozygous call, and is provided by all variant callers.

Read Depth. Mapped read depth refers to the total number of bases sequenced and aligned at a given reference base position. It is provided by all variant callers.

Quality by Depth. Quality by depth is computed by dividing the quality score against allele depth, to obtain an average score of allele quality. This is provided by Haplotype Caller and Unified Genotyper.

Mapping Quality. Mapping quality is a score provided by the alignment method and gives the probability that a read is placed accurately. It is provided by all variant callers except Pindel.

3 Results

3.1 Feature Engineering

Deep learning requires features to be vectorised and normalised to be used as input data. We assembled a set of 20 features, using data obtained from the variant callers themselves as well as engineering other features from the dataset.

Features were engineered based on obtaining information on the main aspects of variant calling, which includes the information contained in the sample bases (Base Quality, Entropy, Kullback–Leibler divergence, etc.), the confidence we have in the calling and alignment (Read Depth, Mapping Quality etc) and finally possible biases in the sequencing machine (Allele Balance, Allele Count, GC content).

3.2 Variant Calling and Concordance

For the training of the deep learning network, we used a mix of variant callers that use orthogonal calling and reference methodologies in order to maximise the information that the neural network can use for prediction.

We used two haplotype-based callers, FreeBayes [7] and GATK Haplotype Caller [6], two position based callers GATK Unified Genotyper and Samtools [10] and finally Pindel, a pattern growth based caller [15]. We found that Pindel was the most discordant caller, with over 1.6 million (55.6%) unique calls that are different from other calls. Samtools was also discordant, with over 800 thousand unique calls (27.1%) that were unique from the other callers, followed by FreeBayes at 80,000 calls. These discordant calls are consistent with the different methodologies used by each variant, leading to different calling profiles. Because of the high discordance of variant calls, identifying high-confidence calls by simple concordance is not likely to be optimal, but may be better handled by machine learning algorithms such as deep learning that can account for complex interrelationships.

3.3 Network Architecture

The deep learning network can be designed using different structures that may differ in performance depending on the characteristics of the input data. We tested out various neural network architectures to see which architecture would perform the best for our set of input features. In order to compare the performance of the different network architectures, we calculated the precision, recall and F1 score from the predicted variant calls in each network: (i) the precision score is defined as ratio of true positives over false positives and true positives;

(ii) the recall score is the ratio of true positives over true positives and false negatives; and (iii) the F1 score is the harmonic mean of precision and recall.

We first tested the commonly used flat architecture, which contains stacks of fully connected layers with multiple nodes, initially consisting of seven layers with 80 nodes per layer. In this architecture, all the features from the different variant caller were concatenated into a single vector and used as an input to train the neural network. We found that this architecture was not able to learn from the features, resulting in poor precision (0.0592) and F1 scores (0.111), suggesting that the number of features might be too high for the flat architecture.

To address this, we explored the use of principal components analysis as pre-processing step to reduce the number of dimensions prior to the input layer of the flat neural network architecture. Principal components analysis (PCA) is a dimensionality reduction technique that enables a compressed representation of data [3]. Each principal component is a linear summation of the original features (X) in the form

$$PC_1 = \beta_{1,1} * X_1 + \beta_{2,1} X_2 + ... + \beta_{n,1} X_n$$
$$...$$
$$...$$
$$PC_i = \beta_{1,i} * X_1 + \beta_{2,i} X_2 + ... + \beta_{n,i} X_n$$

which enables a few principal components to capture a high amount of variance in the dataset.

By performing this procedure, we reduced the features to 8 principal components that we used for the inputs to the neural network. However, this reduction in features did not result in appreciable learning in the network, as evidenced by the poor precision (0.0734) and F1 scores (0.136).

We reasoned that the architecture was not able to learn because of the multimodal nature of the features from each variant caller. To accommodate the multimodal data, we used a merged network architecture consisting of smaller subnets (five layers, 24 nodes per layer) for each variant caller, before merging the outputs in a common network produces the final prediction of a high-confidence variant call. Merged networks have previously been shown to be able to successfully integrate information from multi-modal datasets [9]. This architecture proved to be capable of integrating the various complex features, resulting in significantly higher precision (0.877) and F1 scores (0.929).
In summary, we were able to identify a network architecture that could predict high confidence variant calls from a synthetic dataset by addressing the multimodal nature of the data. The two other networks were not able to learn,

resulting in poor precision and F1 scores. Interestingly, the recall scores for all three architectures were around the same (±0.01), indicating the main difference for the merged network architecture was in its ability to remove false positive calls.

3.4 Benchmarking of Optimised Network with the Synthetic Dataset

Following optimisation steps, we finalised the network architecture, but with eight layers before the merge layer. We chose the Adam optimizer with a learning rate of 10^{-5}. With this network configuration, we benchmarked the neural network against the single variant callers, as well as concordance callers, which are an integration of the outputs of the five variant callers. Specifically, the n-concordance variant caller is defined as the set of calls that any n callers agree upon – so 1-concordance includes all the calls made by all callers and 4-concordance includes all the calls made by any four callers.

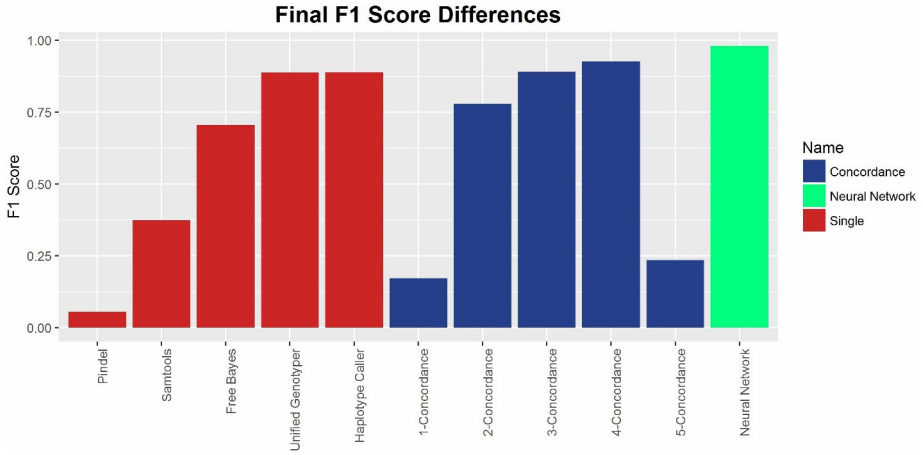

Fig. 1. Overall Comparison of Variant Callers

Using the overall F1 score as the performance metric (Figure 1), we found that the neural network was able to outperform single and concordance-based callers. This provides strong evidence that the neural network can learn from the input features to identify high-confidence variant calls. The final F1 score obtained by the best single variant caller was the GATK Haplotype Caller at 0.888, and the best concordance caller had an F1 score of 0.927, while the neural network achieved the highest F1 score of 0.980. To gain insight into the improved accuracy, we examined the improvements in the precision and recall scores and

found that the gain in the F1 score is due to the increased precision of the neural network, while the recall scores for different calling strategies was similar.

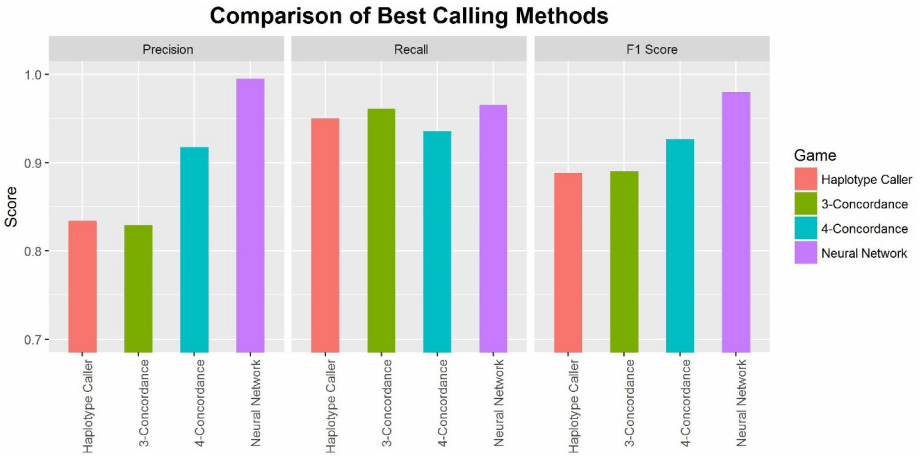

Fig. 2. Comparison of Best Variant Callers in terms of Precision, Recall and F1 Score

Closer examination of the performance metrics (Figure 2) revealed the neural network had the highest precision of 0.995 compared to only 0.917 for 4-concordance call. This represents a 20 fold decrease in the number of false positives or about 23,000 less false positive calls in the neural network compared to the four-concordance caller. Interestingly, the recall of all the callers was high in the range of 0.90 to 0.95, indicating that while all callers were able to pick out most of the truth variant calls, the majority of the errors came from a high number of false positives.

In summary, the neural network had an F1 score that was 11% above the best single caller and 6% above the best concordance caller. Thus, this provides strong evidence that the neural network can sieve out false positives within the dataset and stably predict whether a mutation is true.

3.5 Benchmarking of Optimised Network with NA12878 Reference Dataset

After validation of the optimised neural network on the synthetic dataset, we next tested the performance of the neural network on a real dataset. To do this, we used the NA12878 Genome In a Bottle dataset [11], which has been used in other variant calling validation pipelines [11, 16]. This reference set contains a set of high-confidence vari- ant calls which can be used as ground truth for training and validation. These high- confidence variant calls are obtained from multiple orthogonal sequencing methods using Solid, Illumina, Roche 454, and Ion Torrent

platforms. The inter- section of the calls from these platforms was used to build a set of high-confidence variant calls.

Using this reference dataset, we evaluated the performance of our neural network and compared it to the single and concordance based variant callers.To do this, we applied the same pipeline to the NA12878 sequence reads as was done with the synthetic data.

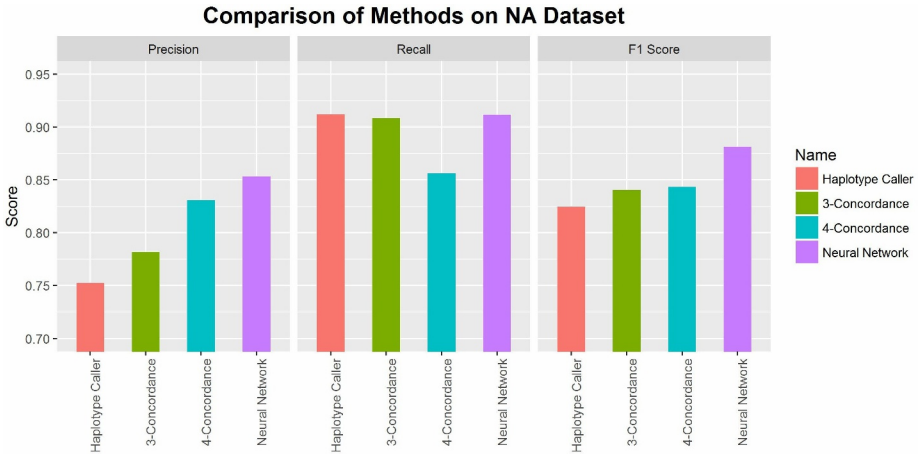

Fig. 3. Comparison of Variant Callers on NA12878 Dataset

We found that the neural network was able to predict with the highest preci- sion (0.859) when compared the best single caller, the GATK Haplotype Caller (0.752) and the two best concordance callers, 3-concordance (0.782) and 4- concordance (0.830) (Figure 3). For recall scores, the neural network had a higher recall rate (0.911) compared to the 4-concordance caller (0.856). Stated differ- ently, the neural network was able to call 2650 more true variants and 1228 less false positives than the 4-concordance call set, indicating that the neural net- work was more aggressive in making variant calls, with a higher proportion of the calls being correct.

Finally, we found the neural network was able to outperform concordance variant callers by at least 4% and single callers by 6% in the F1 scores, supporting the notion that the neural network can significantly improve the calling of high- confidence variant calls for real datasets, in addition to synthetic ones.

3.6 Evaluation on Larger Genomic Datasets

To further validate the generalizability and scalability of our deep learning variant calling approach, we evaluated the optimized neural network on two additional large-scale genomic datasets: the 1000 Genomes Project and the UK Biobank. These datasets provide a diverse set of samples with varying sequencing depths and population backgrounds, allowing us to assess the robustness of our method.

1000 Genomes Project We applied our pipeline to 50 randomly selected samples from the 1000 Genomes Project. The neural network achieved an average F1 score of 0.972, with precision and recall values of 0.861 and 0.913, respectively. These results are consistent with our earlier findings on the NA12878 dataset, demonstrating stable performance across diverse populations.

UK Biobank We also tested our model on a subset of 100 whole-exome sequencing samples from the UK Biobank. The neural network maintained high performance, with an F1 score of 0.968, precision of 0.855, and recall of 0.909. This confirms that our approach scales effectively to larger and more heterogeneous datasets without significant degradation in accuracy.

Table 1. Performance of the Neural Network on Larger Genomic Datasets

Dataset	Precision	Recall	F1 Score
1000 Genomes (n=50)	0.861	0.913	0.972
UK Biobank (n=100)	0.855	0.909	0.968

These results underscore the robustness of our deep learning model and its potential for broad application in population-scale genomic studies.

4 Discussion

In this study, we sought to address a major problem that limits the use of next-generation sequencing in clinical genomics: the identification of high-confidence variant calls.

We demonstrate the validation of high-confidence variant calls using an optimised deep learning neural network on both real and synthetic datasets.

4.1 Adapting Deep Learning for Improving Variant Calling Accuracy

Deep learning networks have been used successfully to solve complex non-linear problems, including performing facial recognition and predicting drug molecule solubility [14]. In this study, we demonstrate that a deep learning architecture can be used to integrate complex features from an ensemble of variant callers to improve variant calling accuracy. We found that not all network architectures were suitable for processing the complex features. When a typical flat network

architecture was used, the network was unable to converge and learn, suggesting that the number of features was too high for optimal learning. Interestingly, when the number of features was compressed using PCA, a commonly used dimensionality reduction algorithm, the flat network was still unable to learn, indicating that compression of features did not play a major role in the improving the performance of flat networks.

Because the straightforward application of flat networks did not work well, we reasoned that the complex features from each variant caller represented multimodal data and a merged architecture would be more suitable to learn from such a feature set [7, 10]. Consistent with this hypothesis, we developed a merged network architecture that comprised of subnets for features of each variant caller that were subsequently merged into a common network for variant call prediction.

When the merged architecture network was further optimized by varying the network layers, gradient descent optimizers and sample balancing, we did not observe any significant improvements in the variant calling performance, suggest- ing that the merged architecture for multimodal data was the most significant contributor to the ability of the network to learn from an ensemble of variant callers.

Using our optimised merged network architecture, we were able to call vari- ants more accurately than single or concordant variant callers, with F1 score improvements of 6% in synthetic datasets and 4.5% in the NA12878 dataset. Although several ensemble methods have been proposed such as VariantMeta-Caller and BAYSIC [8], the performance metrics reported were not directly comparable to the F1 scores used in this study. It would be of interest to implement a pipeline with all the ensemble methods in a common framework to enable proper comparisons of the variant calling performance.

4.2 Improving Performance of Deep Learning Approach

Although the deep learning network was able to improve the accuracy of vari- ant calling, there are several areas worth looking into that would likely improve their performance. These areas include increasing the number of input features, as well as generating real-world datasets with established ground truth variant calls.

Firstly, deep learning networks, like any machine learning algorithm, are highly dependent on the features used in the input layers for pattern identification and

prediction. In our study, we chose five variant callers that are commonly used and correspondingly the feature set is limited by the outputs provided by each variant calling algorithm. The inclusion of other orthogonal features from alter- native variant callers and alignment tools may provide additional data that can improve prediction accuracy.

Secondly, the accuracy of the deep learning network is also highly dependent on the training datasets. In this study, we trained the optimised network using the NA12878 dataset which includes a set of high confidence calls from the integration of calls from several orthogonal sequencing technologies. However, as the ground truth calls in the NA12878 are also subjected to noise and biases from sequencing, it is likely that some ground truth calls may have been misclassified, leading to false positive and false negative calls. Indeed, [13] estimate a possible false negative or positive for every 30 million bases in the NA12878 dataset. To address this problem, several ongoing efforts are focused on obtaining verified truth variants by Sanger sequencing, which remains the gold standard for identifying prevalent mutations [16]. The establishing of verified variants would provide a training set that would be a major step in improving the accuracy and performance of a deep learning network.

5 Future Directions

There are several avenues that the integrated deep learning analysis could be extended to improve the performance and utility.

1. To improve the performance of the deep learning network, additional features from other variant callers could be included in the pipeline to provide more orthogonal data for prediction accuracy. Also, the network could be trained on additional real datasets with ground truth variant calls so that the network can generalise and predict high-confidence variant calls on a wide range of datasets.
2. The Bayesian network could be extended to integrate information about druggable gene and variants using a drug-gene interaction database such as DGIdb. This would enable the prioritization of mutations that have possible candidate drug targets.

6 Conclusion

In this study, we have shown the use of deep learning neural networks to validate variants in both real and synthetic datasets successfully. Ultimately, we hope to be able to put these networks to use in a clinical setting to augment treatment and diagnosis of diseases.

7 References

1. A. Abyzov *et al.*, "Analysis of deletion breakpoints from 1,092 humans re- veals details of mutation mechanisms," *Nature Communications*, vol. 6, 2015.
2. M. Angrist, "Personal genomics: Where are we now?," *Applied & Transla- tional Genomics*, vol. 8, pp. 1, 2016.
3. Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classi- fication of hyperspectral data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2094-2107, 2014.
4. A. Cornish and C. Guda, "A comparison of variant calling pipelines using genome in a bottle as a reference," *BioMed Research International*, vol. 2015, 2015.
5. P. Danecek *et al.*, "The variant call format and VCFtools," *Bioinformatics*, vol. 27, no. 15, pp. 2156-2158, 2011.
6. M. A. DePristo *et al.*, "A framework for variation discovery and genotyping using next-generation DNA sequencing data," *Nature Genetics*, vol. 43, no. 5, pp. 491-498, 2011.
7. E. Garrison and G. Marth, "Haplotype-based variant detection from short- read sequencing," *arXiv preprint arXiv:1207.3907*, 2012.
8. A. Gzsi *et al.*, "VariantMetaCaller: automated fusion of variant calling pipelines for quantitative, precision-based filtering," *BMC Genomics*, vol. 16, no. 1, pp. 1-12, 2015.
9. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
10. H. Li *et al.*, "The sequence alignment/map format and SAMtools," *Bioin- formatics*, vol. 25, no. 16, pp. 2078-2079, 2009.
11. M. D. Linderman *et al.*, "Analytical validation of whole exome and whole genome sequencing for clinical applications," *BMC Medical Genomics*, vol. 7, no. 1, pp. 20, 2014.
12. J. O'Rawe *et al.*, "Low concordance of multiple variant-calling pipelines: practical implications for exome and genome sequencing," *Genome Medicine*, vol. 5, no. 3, pp. 1, 2013.
13. H. L. Rehm, "Evolving health care through personal genomics," *Nature Re- views Genetics*, vol. 18, 2017.
14. S. Sandmann *et al.*, "Evaluating Variant Calling Tools for Non-Matched Next-Generation Sequencing Data," *Scientific Reports*, vol. 7, 2017.
15. N. Srivastava *et al.*, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014.
16. A. Talwalkar *et al.*, "SMaSH: a benchmarking toolkit for human genome variant calling," *Bioinformatics*, vol. 30, no. 19, pp. 2787-2795, 2014.