A Novel Pipeline for Virus Integration Sites Detection in Tumor Genomes Using Deep Learning

Lorrayya Williams
School of Applied Computational Sciences
Department of Computer Science & Data Science
Meharry Medical College
Nashville, TN, USA
lwilliams24@mmc.edu

Vibhuti Gupta
School of Applied Computational Sciences
Department of Computer Science & Data Science
Meharry Medical College
Nashville, TN, USA
vgupta@mmc.edu

***Abstract*—Cancer is one of the leading causes of death worldwide. Pathogenic viruses are estimated to be responsible for 15% of all human cancers globally and pose significant threats to public health. Viruses integrate their genetic material into the host genome, increasing the risk of cancer promoting changes in it. To understand the molecular mechanisms of virus-mediated cancers, it is crucial to identify viral insertion sites in cancer genomes. However, this effort is hindered by the rapidly increasing volume of tumor sequencing data, along with the challenges of accurate data analysis caused by high viral mutation rates and the difficulty of aligning short reads to the reference genome. Thus it is crucial to develop an efficient method for virus integration site detection in tumor genomes. This paper proposes a novel pipeline to identify viral integration sites leveraging deep Convolutional Neural Networks (CNN). Our contributions are twofold: (i) We propose and integrate three novel matrix generation methods into the pipeline, developed after aligning the host and viral genomes with their respective reference genomes.; (ii) We employ one-hot encoded images with reduced computational complexity to represent viral integration sites and harness the capabilities of Deep CNN networks for detection. The paper illustrates our proposed approach and presents experiments conducted using both synthetic and real sequencing data. Our preliminary experimental results are promising, showcasing the effectiveness of the proposed methods in detecting viral integration sites.**

***Index Terms*—CNN, sequencing, NGS, matrix**

I. Introduction

Cancer poses a significant threat to global health, with the incidence of the disease steadily increasing. By the end of 2024, it is projected that around 2 million new cancer cases will be diagnosed, and approximately 600,000 people will die from the disease in the United States alone [1]. Pathogenic viruses pose significant threats to public health throughout the world and are estimated to be responsible for 15% of all human cancers globally [2] [3]. For example, human papillomavirus (HPV) causes 91 percent of cases of cervical cancer, the fourth most common cancer in women globally [4]. To effectively diagnose and treat cancer, it is essential to deepen our understanding of oncogenesis.

Viruses are a significant cause of oncogenesis. Some common viruses that contribute to oncogenesis include Human papillomavirus (HPV), which is linked to reproductive cancers; Epstein-Barr virus, often associated with lymphoma; and Hepatitis B and C, which are related to liver cancers. In cervical cancer and some other viral mediated cancers, viruses can integrate their genetic material into host cell genome [5]–[10]. The process of viral integration damages the host cell's DNA and elevates the risk of cancer-promoting changes in the host genome [7], [8]. Insertion of viral DNA can be particularly devastating at proto-oncogenes where cell proliferation is controlled. Therefore, to understand the molecular mechanisms of viral mediated cancers, a necessary step is to detect viruses and their insertion sites in cancer genomes. The initial step in this process is the precise identification of viral integration sites within the host genomes. Identifying viral insertions in the host genome will facilitate the recognition of patterns associated with viral integrations and help pinpoint pathways involved in cancer development.

Over the past two decades, rapid advances in next-generation sequencing (NGS) technologies [11]–[13] have led to their widespread use in hospitals. Consequently, many NGS-based tools have been developed to detect viruses and their insertion sites. However, due to the challenges associated with accurate detection, the sensitivity of current tools remains unsatisfactory, falling short compared to established quantitative technologies [14], [15]. Virus insertions in human genomes contribute to genomic instability, resulting in increased mutation rates. These fusion-induced mutations complicate the alignment of short reads to reference genomes, making it difficult to detect virus integration sites. Additionally, the high mutation rates of viruses lead to sequence divergence, which negatively affects detection. As a result, NGS reads sampled from actual virus genomes are less likely to align with commonly used virus reference sequences.

Current NGS tools employ statistical models to detect viral integration events [16]–[21]. However, due to noise in sequencing data and uncertainties in read alignment, these tools only keep reads that meet specific quality criteria for analysis. The thresholds for these filters are primarily set empirically to manage false positive rates. This filtering compromises the ability of current tools to detect cryptic viral insertions. Some tools, like VirusFinder [11], [12], have been instrumental in identifying integration sites of diverse, previously undiagnosed viruses from sequencing data. Due to its precision and unique methodology, VirusFinder has been extensively used in inves-

tigating various types of cancer [15], [22]. However, as the volume of genomic data grows exponentially and due to the limitations of detecting cryptic viral insertions, further research is needed to address these challenges.

With recent advances in machine learning, deep neural networks are now widely applied in various fields, including image recognition, genomic analysis, and COVID-19 detection. Deep Convolutional neural networks (CNNs) are highly effective in visual recognition tasks, as they efficiently capture the spatial and temporal dependencies within the input [23]. Recently Deep CNNs are used in the genomics field of research [24], [25]. This is because after sequencing reads from a sample are aligned to the reference genome (or transcriptome), they effectively create an image. In contrast to traditional methods, deep CNNs consist of multiple layers of processing, allowing them to hierarchically learn complex features from imaging data. This capability makes them well-suited for addressing the complexities involved in virus integration detection [26].

In this paper, we propose a deep convolutional neural network (CNN) based approach to detect virus integration sites in tumor genomes to improve NGS-based detection of virus integration. Our major contributions in this paper can be summarized as follows:

1) We propose a novel pipeline to identify viral integration sites in tumor genomes leveraging Deep Convolutional Neural networks (CNNs).
2) We propose and integrate three novel matrix generation methods into the pipeline, developed after aligning the host and viral genomes with their respective reference genomes as an image to apply Deep CNNs.
3) We employ one-hot encoded images with reduced computational complexity to represent viral integration sites and harness the capabilities of Deep CNN networks for detection.
4) We experiment on both synthetic and real sequencing data and evaluate the performance using various metrics.

To the best of our knowledge, this paper represents the first effort to use different image matrix representations to characterize virus integration sites from NGS data that is efficient, and accurate. This will not only aid cancer researchers in exploring the etiological relationship between viruses and cancer but also create a cutting-edge tool for the scientific community.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 describes the dataset followed by the data description and novel matrix generation algorithms. Section 4 describes the experiments and results and Section 5 concludes the paper.

II. Related Work

Recent research on viral integration site detection methods includes statistical methods [16]–[21] and deep learning methods [24], [25], [27]–[31].

GENE-IS [16] is the Genome Integration Site Analysis Pipeline which is developed to provide efficient and accurate detection of NGS-based viral integration sites in gene therapy data. GENE-IS used many traditional statistical approaches to detect the viral integration sites. [17] developed VirTect to detect viral integration sites from multiple related tumor sequencing data from the same patient. Their algorithm examined the short reads after mapping to the reference genome, cluster them, and then applied local realignment procedure to detect the exact breakpoint of the integration sites. [18] proposed a tool HGT-ID to detect viral integration sites through multiple steps including preprocessing of unaligned read, viral site detection using soft clipping and discordant approach, and finally arranging them using a scoring function. [19], [20] developed an approach to utilize single breakends and correcting the read alignment for accurate viral integration site detection. All the statistical methods and tools have shortcomings to keep reads that satisfy specific quality criteria based on thresholds which limits the application to specific use cases.

There are some works using Deep CNNs for virus integration sites detection. [24] developed a deep learning framework to detect human T-cell leukemia virus type 1 (HTLV-1) integration sites and leveraged it for multiple applications such as motif discovery, and cis-regulatory factor identification. Deep-HINT [25] employs a CNN combined with an attention module to capture the contextual sequence features of HIV integration, allowing it to predict HIV integration sites from primary DNA sequences. In addition to HIV, similar frameworks have been employed to investigate the local genomic environments of integration sites for other virus types, such as HBV [28] and HPV [27]. Deep CNNs have also gained considerable popularity for detecting genomic variants from NGS data. DeepVariant, a tool that led this approach, converts aligned reads indicative of candidate variants into images, subsequently using CNNs to identify small variants [29]. Another tool, NeuSomatic, utilizes CNNs to detect somatic variants rather than germline variants [30]. Deep CNNs have also demonstrated strong performance in detecting complex structural variations (SVs) [31]. Unlike the above approaches for viral integration site detection, our proposed approach also used Deep CNN's however we proposed novel matrix generation approaches which was not explored in any of the aforementioned approaches.

III. Methodology

In this section, we briefly describe our overall pipeline, proposed matrix generation methods, and CNN approach used to build the predictive models for viral integration site prediction.

A. Overall Framework

Earlier methods extracted viral integration sequences from their surrounding sequence context. In this work, we employ three methods that progressively incorporate the sequence context at increasing levels. Figure 1 shows the proposed pipeline for virus integration site detection in tumor genomes using Deep CNN. Our approach begins with Fastq files that contain viral integration sites. These files are aligned to the hg19

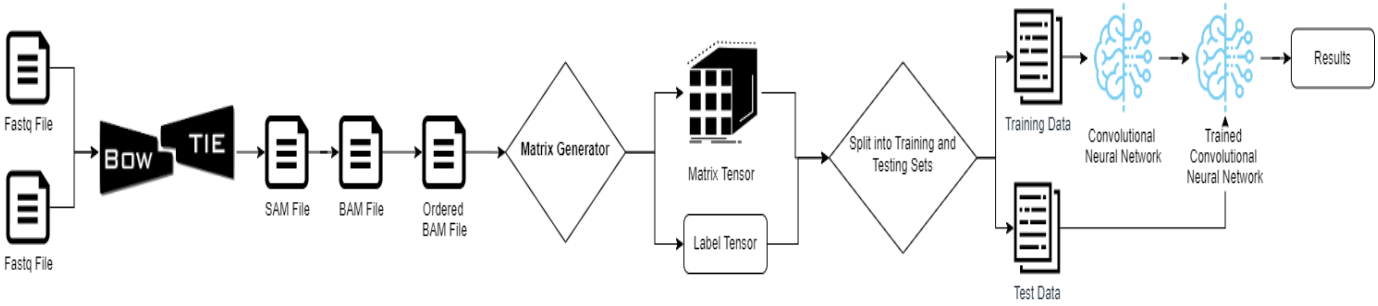

Fig. 1. Pipeline for Virus integration site detection in Tumor genomes using Deep CNN

reference genome utilizing the Bowtie2 alignment program as shown in Fig 1. We used hg19 reference genome as the collected data was based on hg19 reference genome so to maintain consistency across samples, hg19 reference genome is used. The Bowtie2 program is wrapped in a novel python wrapper that implements the Bowtie2 program. Bowtie2 generates a Sequence alignment/map format (SAM) file. This SAM file is stored as a Binary alignment Map (BAM) file which is then ordered before the Matrix generation portion. As noted earlier, significant attention was devoted to developing and comparing three methods for generating matrices. The generated SAM files are organized in such a way that the resulting DNA sequence matrix accurately reflects the order of the genes.

The novelty of our approach is that DNA sequences with viral integrations are analyzed in relation to the complementary DNA strand and their arrangement within the chromosome. These factors are taken into account at varying levels, depending on the matrix generation method used. We proposed three novel algorithms for generating these matrices in this paper. The matrices are represented as a tensor and equivalent to images of genomic regions. These matrices are labeled. The data in this format is split into training and testing sets as shown in Fig 1. Finally, a CNN is trained using the training set and applied to the test set to predict the viral integration sites in the given tumor genome sequences.

B. Matrix Generation Method 1

We employ one-hot encoded images to represent viral integration sites because they lead to a simplified CNN architecture with lower computational complexity for both training and prediction. One-hot encoded images are 3-dimensional images with many channels, each recording a certain alignment signal. These one-hot encoded images are represented as a 3-dimensional tensor, corresponding to the three channels in an image. These images are generated from the matrices. The matrix is created by aligning the reads from the input genomic sequence with the reference genome sequence. If there is a match among the characters *A, C, G, T*, a value of 1 is assigned; otherwise, a value of 0 is assigned for no match.

We proposed algorithm 1 as the first matrix generation method. This idea of this method is based on the matrix generation performed in the DeepHBV [28] data pipeline. As shown in Algorithm 1, the input consists of a DNA sequence that has to be aligned with the reference genome sequence (Line 1). However, the output is a 3-dimensional tensor representing the matrix generated (Line 2). For each sequence in S and for each base pair in the sequence, if there is a match with the reference genome, a value of 1 is assigned; otherwise, a value of 0 is assigned for no match (Lines 2-6). This process is repeated until there is no sequence left. Finally, it returns a matrix T (3-dimensional tensor) of zeroes and ones. This matrix generation has sequences with viral integration and without viral integration. Each row contains a sequence with or without a viral integration site.

Algorithm 1 Matrix generation method 1

Input: $S \leftarrow$ a DNA sequence list
Output: $T \leftarrow$ a 3-Dimensional Tensor

1: **procedure** MATRIX-GENERATOR-FIRST(S)
2: **for** each $s \in S$ **do**
3: **for** each base $b \in s$ **do**
4: $T \leftarrow 1$ at Base location
5: **end for**
6: **end for**
7: **end procedure**

C. Matrix Generation Method 2

The second matrix generation method takes account of both strands in order of appearance in the genome. Each line represents a strand of DNA with opposing strands being in the following row. This allows for the matrix to be created within the context of the strand itself rather than just as a single strand. For each line in the BAM file, there are three values extracted from it, *CIGAR (Compact Idiosyncratic Gapped Alignment Report) value, Sequence, and position*. The position of the insert is also input for labeling purposes. Each base in the sequence has a 1 inserted into the tensor that corresponds to the column of that base.

We proposed algorithm 2 as the second matrix generation method. As shown in Algorithm 2, the input consists of a DNA sequence that has to be aligned with the reference genome sequence and a BAM file (Line 1). However, the output is a 3-dimensional tensor representing the matrix generated (Line

Algorithm 2 Matrix generation method 2

Input: $S \leftarrow$ a DNA sequence list, $B \leftarrow$ BAM File
Output: $T \leftarrow$ a 3-Dimensional Tensor

1: **procedure** MATRIX-GENERATOR-SECOND(S,B)
2: $Len \leftarrow 0$
3: $Ind \leftarrow 0$
4: **for** each $b \in B$ **do**
5: $Cigar\_list \leftarrow$ split(CIGAR)
6: **if** len($Cigar\_list$) == 2 **then**
7: **if** Cigar == 'M' **then**
8: **for** each base $b \in S$ **do**
9: $T \leftarrow 1$ at Base location
10: **end for**
11: **else**
12: $Len \leftarrow 0$
13: $Ind \leftarrow 0$
14: **for** time in range(len($Cigar\_list$)/2) **do**
15: **if** Cigar == 'M' **then**
16: **for** $j \leftarrow Len$ to $Len + Cigar\_list[time + 2] - 1$ **do**
17: $T \leftarrow 1$ `at Base location in Tensor`
18: **end for**
19: **end if**
20: $Len \leftarrow Len + Cigar\_list[time + 2]$
21: **end for**
22: **end if**
23: **end if**
24: **end for**
25: **end procedure**

2). We initialized the length and index as zero (Lines 2-3) For each base pair in the BAM file, we split the CIGAR value, if the length of CIGAR list value is 2 and then if value is a match then a value of 1 is assigned (Lines 1-10). Otherwise, if the length of CIGAR list value is greater than 2, then we loop through every 2 elements of the CIGAR value. If there is a match "M" in the cigar value for the set of indexes in the sequence, so for all those indices, 1 is inserted into the tensor that corresponds to the column of that base (Lines 12-24). This process is repeated until there is no sequence left in the BAM file. Finally, it returns a matrix T (3-dimensional tensor) of zeroes and ones. In this method, we have used the CIGAR list value as 2 as CIGAR is a column generated in a BAM/SAM file that shows the matches, deletions, and insertions for the read maps between the DNA sequence with the reference genome. We divided the CIGAR column into a list where the odd-indexed elements represent the number of bases, and the even-indexed elements indicate whether it was a match or a deletion. If the length of the Cigar_list exceeds 2, we iterate through it, performing different actions depending on the values it contains.

D. Matrix Generation Method 3

The third matrix generation method is the most context-dependent. It takes into account whether or not the bases match with the corresponding strand at the base pair location. Previous matrix generation methods do not take into account whether or not the base pair strands match. This adds to the novelty of the matrix generation methods. The algorithm starts with creating a dataset with a row for each base in the sequence that includes direction, location, and whether it matches with reference. Then separate that data by direction 3'à5' one way 5'à 3' the other way. We loop through the data to check with there is a match from reference genome and insert 1 to tensor. Then we check for the matching opposite strand. If matches then insert 3 in the strand.

Algorithm 3 Matrix generation method 3

Input: $F \leftarrow$ BAM file
Output: $M \leftarrow$ Mapping List

1: **procedure** MATRIX-GENERATOR-THIRD(F)
2: **for** each line l in F **do**
3: **if** base == match **then**
4: $M \leftarrow$ seq, mat, pos, dir, count
5: **end if**
6: **end for**
7: $M \leftarrow M$ sorted by values of pos
8: Three_five $\leftarrow M$ in 3' $\rightarrow$ 5' direction
9: Five_three $\leftarrow M$ in 5' $\rightarrow$ 3' direction
10: **for** each row in M **do**
11: **if** dir == 3' $\rightarrow$ 5' **then**
12: **if** b aligned genome **then**
13: $T \leftarrow 1$
14: **if** b matches with opposing strand **then**
15: $T \leftarrow 3$
16: **end if**
17: **else if** b match with opposing strand **then**
18: $T \leftarrow 2$
19: **end if**
20: **else if** dir == 5' $\rightarrow$ 3' **then**
21: **if** b aligned genome **then**
22: $T \leftarrow 1$
23: **if** b matches with opposing strand **then**
24: $T \leftarrow 3$
25: **end if**
26: **else if** b match with opposing strand **then**
27: $T \leftarrow 2$
28: **end if**
29: **end if**
30: **end for**
31: **end procedure**

We proposed algorithm 3 as the third matrix generation method. As shown in Algorithm 3, the input consists of a BAM file containing all the sequences with viral integration sites and with no integration sites (Line 1). However, the output is a mapping list (Line 2). For each line in the BAM file (F), if

there is a match for the base pairs between reference genome and the input sequence then a mapping list is generated with the sequence, position, and count attributes (Lines 2-6).

E. Convolutional Neural Network

Recent advances in artificial intelligence have made deep CNN the primary model for virtually every image related problem. Deep CNN, as a class of deep learning algorithms, is composed of stacks of processing layers, allowing it to learn complex features hierarchically from imaging data. The CNN networks typically utilize multiple convolution-pooling modules that are built on top of each other to learn from input-output pairs. Input images will be fed into the first convolution-pooling module of the CNN networks to perform a series of convolution operations followed by rectified linear activation (ReLU) and max-pooling to extract linear features from the input image. The output of the final convolution-pooling module of the CNNs will be fed to a fully connected module, which will be trained to perform predictions. Model training is fully automated, thereby removing the need of feature engineering and human intervention. This makes deep CNNs suitable for handling the complexity of virus insertion site characterization, and thus, effectively avoiding the limitations of today's tools.

IV. Experiments and Results

This section provides the experimental results to evaluate the effectiveness of our proposed approaches in the real and synthetic sequencing datasets.

A. Dataset and experimental setup

We applied our proposed matrix generation methods with CNN approach for viral integration site detection on a set of real sequencing data and synthetic data. The synthetic data is generated with 5 viral integration sites for the experimental purpose. We generated the synthetic dataset by following the distribution used in the paper [11]. Aligning samples to a reference genome can be computationally expensive, so to reduce the time while alignment, we used only Chromosome 1 viral insertions for our experiments. For preliminary testing, we utilized simulation data with a single viral insertion. After that, we web scrapped viral integration site data from VISDB which is Viral Integration Site Database that offers viral integration site sequences and location. The data we used from this site was focused on Chromosome 1 HBV viral integration sites. Some of these viral integration sites are overlapping. Each viral integration was looked at including 10,000 bases both upstream and downstream from the viral integration site. [28]. For this project, Google Colab was utilized for all steps of development. Google Colab's basic computing offers 12.7 GB of System RAM, 15 GB of GPU, and 112.6 GB of Disk space.

We have used the simple convolutional neural network (CNN) for our experiments. Our CNN architecture consists of 3 convolutional layers, 3 one-dimensional max-pooling layers of pool size 2, and one dense layer followed by the final classification layer. One convolutional layer has 32 filters with 3 kernels and the other 2 convolutional layers have 64 filters with 3 kernels. The dense layer has 64 units with the activation function RELU and the classification layer has 1 unit with the activation function as sigmoid.

TABLE I
Deep CNN Results with all matrix generation methods

Proposed Methods	Testing Accuracy	Loss
Matrix-generation-Method1	0.99	0.08
Matrix-generation-Method2	0.99	0.06
Matrix-generation-Method3	1.0000	0.69

B. Experimental results

We experimented with all three proposed matrix generation algorithms on the real-data marking HBV site integrations within Chr1. Synthetic data was primarily used for preliminary testing to develop methods created for the data pipeline. For each set of experiments, we apply the matrix-generated method, followed by the CNN, and finally the evaluation performance. The data is split into training and testing sets with the training data used for training and testing data to evaluate the model performance. We used 70% data for training and 30% for testing in the experiments. As shown in Table 1, all the methods has good performance with Methods 1 and 2 as accuracy 0.99 and Method 3 has accuracy of 1. Method 3 has the best performance among all the methods. The accuracy of the matrix generation methods developed is higher. However, these are the preliminary results to demonstrate the feasibility and effectiveness of the proposed methods. There is still a need to broaden the scope of experiments across diverse clinical datasets to achieve more comprehensive insights. Additionally, it is necessary to adjust the contextual understanding of viral integration sites, considering their placement within the sequence and the number of upstream and downstream bases included.

V. Conclusion

We have presented our proposed pipeline for virus integration site detection using Deep CNN approach in this paper. We proposed three different matrix generation methods and evaluated them on the synthetic dataset containing viral integration sites. This approach has potential to improve the detection accuracy of virus integration sites which will further help in understanding the etiologic association of viruses with cancer and other diseases. We are applying the proposed approach in the real sequencing samples of various publicly available datasets. In our future work, we aim to further refine the CNN model and conduct testing on a broader dataset to enhance the accuracy and precision of our deep learning predictive models. Moreover, we will experiment with alternative CNN architectures such as VGG16 or VGG19 etc.

Acknowledgment

This project was supported, in part, by grants from the U.S. National Science Foundation (IIS2334391), and Diversity in cancer research institutional development (ACS-DICRIDG)

grant award number ACS DICRIDG-21-071- DICRIDGT. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not reflect the views of NSF or other funding agencies.

REFERENCES

[1] American Cancer Society. 2024 cancer facts figures. https://www.cancer.org/research/cancer-facts-statistics/all-cancer-facts-figures/2024-cancer-facts-figures.html, 2024. Accessed: [insert access date here].

[2] Donald Maxwell Parkin. The global health burden of infection-associated cancers in the year 2002. *International journal of cancer*, 118(12):3030–3044, 2006.

[3] H. Z. Hausen. Viruses in human cancers. *Science*, 254(5035):1167–1173, 1991.

[4] World Health Organization. Human papillomavirus (hpv) and cervical cancer. https://www.who.int/news-room/fact-sheets/detail/human-papillomavirus-(hpv)-and-cervical-cancer, 2021. Accessed: June 2022.

[5] N-PD Nguyen, V Deshpande, J Luebeck, PS Mischel, and V Bafna. Vifi: accurate detection of viral integration and mrna fusion reveals indiscriminate and unregulated transcription in proximal genomic regions in cervical cancer. *Nucleic Acids Research*, 46(7):3309–3325, 2018.

[6] Y. Wongworawat. Factors promoting human papillomavirus mediated cervical carcinogenesis. Master's thesis, Loma Linda University, 2016. Published online June 1, 2016.

[7] Z Jiang, S Jhunjhunwala, J Liu, and et al. The effects of hepatitis b virus integration into the genomes of hepatocellular carcinoma patients. *Genome Research*, 22(4):593–601, 2012.

[8] K Akagi, J Li, TR Broutian, and et al. Genome-wide analysis of hpv integration in human cancers reveals recurrent, focal genomic instability. *Genome Research*, 24(2):185–199, 2014.

[9] W-K Sung, H Zheng, S Li, and et al. Genome-wide survey of recurrent hbv integration in hepatocellular carcinoma. *Nature Genetics*, 44(7):765–769, 2012.

[10] A Khan, Q Liu, X Chen, and et al. Detection of human papillomavirus in cases of head and neck squamous cell carcinoma by rna-seq and virtect. *Molecular Oncology*, 13(4):829–839, 2019.

[11] Q. Wang, P. Jia, and Z. Zhao. Virusfinder: Software for efficient and accurate detection of viruses and their integration sites in host genomes through next generation sequencing data. *PLOS ONE*, 8(5):64465, 2013.

[12] Q. Wang, P. Jia, and Z. Zhao. Verse: a novel approach to detect virus integration in host genomes through reference genome customization. *Genome Medicine*, 7(1):2, 2015.

[13] Y. Chen, H. Yao, E. J. Thompson, N. M. Tannir, J. N. Weinstein, and X. Su. Virusseq: software to identify viruses and their integration sites using next-generation sequencing of human cancer tissue. *Bioinformatics*, 29(2):266–267, 2013.

[14] X. Chen, J. Kost, and D. Li. Comprehensive comparative analysis of methods and software for identifying viral integrations. *Briefings in Bioinformatics*, 20(6):2088–2097, 2019.

[15] M. Zapatka, I. Borozan, and D. S. Brewer. The landscape of viral associations in human cancers. *Nature Genetics*, 52(3):320–330, 2020.

[16] S. Afzal, S. Wilkening, C. V. Kalle, M. Schmidt, and R. Fronza. Gene-is: Time-efficient and accurate analysis of viral integration events in large-scale gene therapy data. *Molecular Therapy - Nucleic Acids*, 6:133–139, 2017.

[17] Y. Xia, Y. Liu, M. Deng, and R. Xi. Detecting virus integration sites based on multiple related sequencing data by virtect. *BMC Medical Genomics*, 12(1):19, 2019.

[18] S. Baheti, X. Tang, O. Brien, and D. R. Hgt-id: an efficient and sensitive workflow to detect human-viral insertion sites using next-generation sequencing data. *BMC Bioinformatics*, 19(1):271, 2018.

[19] D. L. Cameron and et al. Virusbreakend: Viral integration recognition using single breakends. *bioRxiv*, 2021.

[20] R. Rajaby, Y. Zhou, and Y. Meng. Survirus: a repeat-aware virus integration caller. *Nucleic Acids Research*, 51(D1):D1237, 2023.

[21] X. Zeng, L. Zhao, C. Shen, Y. Zhou, G. Li, and W. K. Sung. Hivid2: an accurate tool to detect virus integrations in the host genome. *Bioinformatics*, 37:2623–2625, 2021.

[22] Steven V. Bratman, James P. Bruce, and Brian O'Sullivan. Human papillomavirus genotype association with survival in head and neck squamous cell carcinoma. *JAMA Oncology*, 2(6):823–826, 2016.

[23] Osval A. Montesinos-López, Abelardo Montesinos-López, Paulino Pérez-Rodríguez, Juan A. Barrón-López, John W. Martini, Silvia B. Fajardo-Flores, and José Crossa. A review of deep learning applications for genomic selection. *BMC Genomics*, 22(1):1–23, 2021.

[24] Hao Xu, Jiyuan Jia, H. H. Jeong, and Zhiyuan Zhao. Deep learning for detecting and elucidating human t-cell leukemia virus type 1 integration in the human genome. *Patterns*, 4(2), 2023.

[25] Hao Hu, Anqi Xiao, and Sheng Zhang. Deephint: understanding hiv-1 integration via deep learning with attention. *Bioinformatics*, 35(10):1660–1667, 2019.

[26] Jian Wu. Introduction to convolutional neural networks. Technical Report 23, National Key Lab for Novel Software Technology, Nanjing University, China, 2017.

[27] Ruijie Tian, Ping Zhou, and Ming Li. Deephpv: a deep learning model to predict human papillomavirus integration sites. *Briefings in Bioinformatics*, 2020.

[28] Chao Wu, Xiang Guo, Ming Li, Jie Shen, Xiaogang Fu, Qing Xie, and Jing Liang. Deephbv: a deep learning model to predict hepatitis b virus (hbv) integration sites. *BMC Ecology and Evolution*, 21:1–10, 2021.

[29] Ryan Poplin, P. C. Chang, and A. D. A universal snp and small-indel variant caller using deep neural networks. *Nature Biotechnology*, 36(10):983–987, 2018.

[30] S. Sahraeian, R. Liu, B. Lau, K. Podesta, M. Mohiyuddin, and H. Lam. Deep convolutional neural networks for accurate somatic mutation detection. *Nature Communications*, 10, 2019.

[31] L. Cai, Y. Wu, and J. Gao. Deepsv: accurate calling of genomic deletions from high-throughput sequencing data using deep convolutional neural network. *BMC Bioinformatics*, 20(1):665, 2019.