Deep Learning for Better Variant Calling for Cancer Diagnosis and Treatment

Anand Ramachandran[1], Huiren Li[1], Eric Klee[2], Steven S. Lumetta[1], and Deming Chen[1]

[1]Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign
[2]Biomedical Informatics, Mayo Clinic, Rochester, Minnesota

Abstract— High-throughput techniques have revolutionized the study of genomics and molecular biology in recent years. These methods provide a large quantity of sequence data, and have applications in different areas of bioinformatics. One can sequence parts or whole of an organism's DNA to determine genetic information about an individual or a population, measure expression levels of different genes under different conditions, and determine binding affinity of proteins to DNA segments revealing details regarding gene regulation, at a higher resolution than before. However, different high-throughput methods that target even a single application have different underlying error models. Robust analytic pipelines are necessary to extract necessary information from the raw data. In this paper, we discuss future research directions for developing such analytics using techniques from Machine Learning and Deep Neural Networks. We focus on two applications that will affect the diagnosis and treatment of cancer.

I. Introduction

The Deoxyribonucleic Acid (DNA) molecule encodes the basic instructions for synthesis of proteins. Proteins are the essential agents in most cellular processes. The process by which a segment of DNA is read and synthesized into a protein is hence of great interest in many biological and medical analyses. This process can be influenced at different stages, but the root of the process is the organism's DNA, which encodes information as a specific sequence of components (bases). We refer to the entire sequence of DNA in an organism's cell, the genome. There are four types of bases in the DNA - Adenine(A), Guanine(G), Thymine(T), and Cytosine(C). The genome may be considered to be a sequence defined on the alphabet {A, C, G, T}.

A genome's sequence can be read through the process of DNA sequencing. In recent years, new sequencing technologies, called Next Generation and Third Generation sequencing (NGS, TGS), have been developed that provide high-throughput at low cost. These advances have led to the availability of data at very high resolution, allowing every base in the DNA to be sequenced multiple times providing redundancy, and higher signal to noise ratio. Various techniques have been developed to analyze data obtained through sequencing to determine variations (also called mutations or variants) in the sequence compared to a known standard reference sequence. This allows characterizing an individual's (or a population's) genome sequence(s) in terms of the variations. Variations can be of different types, from changes in single positions, to differences in the structure of arrangements of thousands of bases. Discovering variants (or variant calling) has applications in many fields of bioinformatics, from analyzing the study of diseases in humans [6] to development of breeding strategies for crops [20].

Among the different types of mutations, point substitutions (also called Single Nucleotide Polymorphisms or SNPs), where a single base is replaced with another, and localized insertions and deletions (called indels) account for the vast majority of genomic variations [39]. While NGS platforms produce reads of short length (50-250 bases), they have low error-rates and are cost-effective compared to TGS platforms, and are hence the current standard for obtaining sequencing data for calling SNPs and indels. Many tools have been developed for uncovering SNP and indel variants from NGS data [23][29][33]. While metrics of performance of these tools are relatively high when averaged over a whole genome, there are specific regions of the genome in which most tools consistently underperform [27]. There is a need to develop techniques that set and meet performance goals in such difficult-to-call regions of the DNA. This will involve reexamining the simplifying assumptions and heuristic rules built into the tools, and constructing machinery that can provide a more faithful representation of sequencing data as well as determine patterns in that data, for the purposes of variant calling.

A variant calling workflow attempts to delineate errors associated with sequencing and data preprocessing, and hence is impacted by the characteristics of the underlying sequencing technology. As sequencing platforms continue to evolve, future variant calling frameworks should be able to take advantage of such improvements and aim to provide more robust results, perhaps even combine information from multiple sequencing technologies. For example, though it has higher error rates than NGS, TGS produces longer reads (1-50 kilo bases). Longer reads can help disambiguate long repetitive regions of the genome and have potential to call large structural variants with accuracy. As the accuracy of sequencing using longer reads improve, it will become increasingly attractive to use this technology for calling SNPs and indels. In addition, variants called using both NGS and TGS reads can increase call confidence.

While variant calling reveals putative sources of modulations to cellular tasks, additional analysis is needed to identify the exact nature of these modulations. If a variant occurs in a portion of a DNA that codes for a protein (called a gene), then the variant can manifest itself as a change in the protein being synthesized from that gene (this process is termed "gene expression"). The effect of a variant can realize in other ways too. Variants can affect the ability of certain proteins, called Transcription Factors (TF) to bind to the DNA. TFs regulate the expression of various genes in the DNA. TFs bind to segments of the DNA that have a specific pattern. If a variant causes the TF-binding DNA segment to change its pattern, then the affinity of the protein to that segment will be affected. Multiple high-throughput technologies are available, which can provide measurements indicating the TF-binding affinity of different segments of DNA. Downstream analysis is performed on

this data to determine models of binding affinity of different TFs [5][17][45]. These models can be used to determine the binding affinity of a TF to DNA segments not encountered in the training set, computationally. For example, it is possible to determine the effect of a variant on a known TF binding site using these models.

Knowledge regarding variants and gene regulation are important from the point of view of cancer. The primary cause of cancer is a defect in the genome, either inherited [19] or developed during the lifetime of an individual (somatic) [40]. It is possible that defects associated with cancer affect the production of TFs directly [8] or affect the DNA segments modulating TF-binding [35][5]. It is hence of great importance to study these genomic defects, which can be characterized through variants in an individual's DNA, as well as the mechanisms behind how they manifest through expression, and regulation driven by TFs.

Both variant calling and TF-binding studies handle DNA sequences arising from multiple techologies having different characteristics and error types. Some of the state-of-the-art methods used to tackle these problems rely on expert knowledge and heuristics. Machine Learning and Deep Neural Networks (DNNs) have been greatly successful in recent years in complex classification tasks [24], with the ability to bypass the need for explicit implementation of expert knowledge. DNNs provide a method to specify a problem in terms of data belonging to the problem's domain. Training a sophisticated enough DNN on a sufficiently large dataset allows it to determine the right set of patterns and assumptions relevant to the problem. Sophisticated DNN kernels are available today that have great potential to handle generic sequence type data. These methods, with their great expressive power, and generlizability can be used to adapt to the different types of underlying data in these areas. In the next section we examine some background material on variant calling, and TF binding, as well as Machine Learning and DNN methods that can handle genome sequence data. In the following section we examine some recent works that have successfully applied Deep Learning to these areas. Then we discuss some potential future directions for research in these areas.

II. Background

A. Variant calling

Sequencing data from the predominant NGS platform, Illumina, comes in the form of reads or short sequences of up to a few hundred bases in length [3]. These reads are extremely short compared to the length of the human genome, which is approximately 3 billion bases long. Before the reads can be used for variant calling, they need to be mapped to a reference sequence, which is a standard repository of an assembled human genome [2]. There are many tools such as bwa [28], and bowtie2 [26] that provide such services. It is assumed that SNPs and indels are limited in their span, so the mapping positions of reads to the reference is not affected by them. However, the exact manner in which the mapping is done locally can be affected by them. Once the reads are mapped to the reference, it is possible to list, for every position in the reference sequence, all the reads that map to that position, and specifically, all the read positions and the base from each such read that maps to that position. A consensus among these bases can be assumed to be the actual base at that position in the sequenced organism, and a difference between this consensus and the reference leads to a variant call. This summary is, of course, a simplified picture of variant calling. In reality, more nuanced statistical models and heuristic filters are introduced into the call procedure [11]. Figure 1 illustrates the variant calling task in some detail.

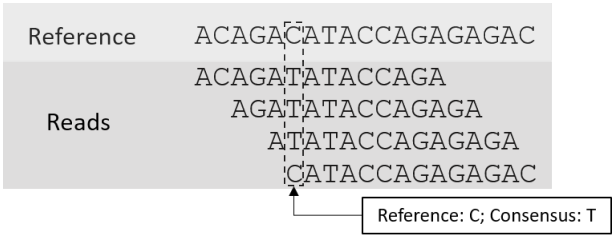


Fig. 1.: A simplified picture of variant calling

Germline variants are those that are inherited from parental DNA, and are found throughout an organism's body. Somatic variants develop during the lifetime of an organism and can be limited to specific parts of the body, such as a malignant tumor. Somatic variant calling is complicated by the fact that the composition of the sample used for sequencing can be heterogeneous with respect to the genome sequences in the cells in the sample, and this heterogeneity makes the data more complex to analyze. While germline variant callers depend on only one set of sequenced data, somatic variant callers sometimes depend on two, one from the affected tissue, and one from healthy tissue, so as to be able to differentiate between the two types of variants.

Most of the state-of-the-art methods for variant calling use a combination of statistical models to measure the signal strength supporting a variant, but also depend on heuristic filters developed from domain knowledge to locate patterns that indicate misalignment or sequencing errors, not captured by the statistical models. Overall, the performance metrics of these tools are high, but they consistently underperform in specific areas of the genome. The next innovation in variant calling needs to address these issues. For example, it is simple to call variants in regions of the genome which have close to uniform distribution of the four bases in the genome, A, C, G and T. However, there are interesting regions of the genome where the GC-content is quite high or very low, and the read coverage (the average number of reads covering a single base in the sequenced genome) in such regions is affected [7]. Since variant callers depend on the coverage of reads to identify different scenarios, this deficiency can cause problems. In addition, Heng Li [27] discovered that most tools cannot efficiently analyze low-complexity regions, regions which contain subsequences that have small entropy, such as localized repeats of the same base (e.g., "AAAAA").

B. Transcription Factor binding models

Transcription Factors are known to bind to segments of DNA with a specific pattern. A pattern or motif associated with a TF is described using what is called a Position Weight Matrix (PWM). A PWM has four rows, and as many columns as the length of the pattern. The magnitude of the entries in a column reflect the proportionality of the four bases in a corresponding column in a DNA segment that matches the pattern. A specific segment of DNA may be scored using the PWM by choosing the entry in each column of the PWM that matches the base in the corresponding position of the DNA, and multiplying them together (Table I).

There are various models that can assign a PWM to a probability of binding. These may simply assume that the entries in the PWM are themselves probabilities and use them in scoring different "alignments" of the PWMs to the given sequence segment [37], or depend on modeling the binding energies of a protein to a given DNA segment as a function of its similarity with the TF's motif [45]. It is possible to use this modeling

infrastructure for different purposes. For example, it is possible to obtain profiles of the binding affinity of specific TFs to different parts of the genome from high-throughput technologies such as ChiP-seq [22], or PBM [32]. It is then possible to estimate the PWMs (originally unknown) that can predict the binding profiles. The estimated PWMs maybe stored for further analysis. Another interesting problem is, given the PWMs, to predict analytically, the binding affinity of a TF to different segments of a newly sequenced genome.

TABLE I
: Scoring a sequence segment using a PWM. Example sequence segment ACGCT; Score = 0.2x0.8x0.8x0.1x0.7

A	**0.2**	0.1	0.1	0.9	0.1
C	0.7	**0.8**	0.02	**0.1**	0.1
G	0.05	0.05	**0.8**	0.0	0.1
T	0.05	0.05	0.08	0.0	**0.7**
seq	A	C	G	C	T

C. Relevance to cancer and approach

NGS techniques are becoming of increasing significance in the understanding and treatment of cancer. Genetic predisposition to certain types of cancers may be determined from the variants in an individual's genome [21][40]. It is possible to improve the accuracy of diagnosis of certain cancer types using small biopsies, when using sequence data in the analysis [21]. The number of mutations in the genome in the affected tissue can provide methods to estimate progression of the disease [40]. In many cancer types, the mechanism of regulation through TFs can be affected due to mutations affecting the production of the TFs [8]. It is also possible that DNA-segments exhibit TF-binding characteristics that lead to unfavourable regulation [35]. A recent work [5] attempted characterizing the effects of somatic mutations collected in the COSMIC [1] database on TF-binding.

To measure the impact of these phenomena, it is necessary to develop methods that are robust across multiple sources of sequencing data. It is important that tool performance be benchmarked and improved in different types of scenarios, and overall metrics averaged over the entire problem-space may mask potential problem areas. For example, the next set of innovations in variant calling should concentrate on improving call confidence in difficult-to-call regions of the genome, as well as overall confidence level in the calls. To break through to the next level of improvement and fidelity in this manner, it may be necessary to question explicit or implicit assumptions involved in modeling these problems in classical approaches.

Neural Networks, being universal function approximators [25], have the ability to represent complex problems in classification and regression. Deep Neural Networks have been successful recently, delivering state-of-the-art performance in tasks in image and speech applications [15][18][24]. They have the potential to learn the underlying patterns of a problem faithfully, from a large number of examples of the problem, without the need for expert intervention or simplifying assumptions. To unlock the potential of DNNs in analyzing genomic data, efforts need to be set up to develop models that are native to genome sequence data, as well as specifying the problem using datasets that faithfully represent the statistics of real data in the field.

D. Models for analyzing genomic data

D.1 Hidden Markov Models

A Hidden Markov Model (HMM) is a directed graph representing a joint distribution of two sequences, one of which is a symbol sequence, and the other a state sequence, where a state is a node in the graph. The symbol sequence is usually associated with some observed quantity in real life, such as a speech signal or a sequence of bases from a genome. The state sequence is unobserved in reality, but is usually attributed to some underlying process that generates the symbol sequence. Using the parameters of the HMM, it is usually possible to infer the most likely state sequence that produced any given symbol sequence, which allows labeling different parts of the sequence. It is possible to use the HMM as a generative model, to produce data simulating the distribution of the real data that it attempts to model. During the generative process, one starts at some state j with probability π_j. At state j, a symbol is generated following an "emission probability distribution" attached to that state, which is a distribution over the alphabet of the symbols in the symbol sequence. Then a transition is made from state j to another state i, which is picked following a transition probability distribution attached to state j. Repeating this process generates a sequence of states (or nodes) and sequence of symbols. This description corresponds to a first order HMM, in which the symbol generated and the next state depend only on the current state. In a zero-th order HMM, the probability of the next state doesn't depend on the current state, but follows a fixed global probability distribution.

HMMs have been the extremely popular in many sequence data analytics in bioinfomatics and genomics. HMMs are used in gene modeling [41], motif detection [37], read mapping and multiple sequence alignment [12], to name a few. In each of these cases, an unobserved property is assigned to a part of an observed genomic sequence. Even though HMMs do not explicitly assume a Markov relationship between consecutive symbols in the observed symbol sequence, they have been found to be unable to model long term dependences across the length of the observed sequence [13].

D.2 Deep Neural Networks

A DNN is a cascade of mathematical functions that process, sequentially, an input data item to produce an output conclusion. To be considered deep, this architecture needs to have more than two functions applied sequentially on the data. Each such function is called a layer, and in many cases a layer itself is a composition of a linear and non-linear function. The most popular applications of DNNs are regression and classification, where an input X is used to predict an output Y. During the training phase, ground-truth values for the output, $\hat{Y}$, are provided and the error between Y and $\hat{Y}$ is minimized, by changing the parameters of the DNN appropriately. During the testing or inference phase, the DNN is used to predict Y from unseen examples of X.

Mathematically, a DNN maps an input to an output using a complex composition of many, usually nonlinear, functions.

$$Y = f_1\left(f_2\left(f_3\left(f_4 \cdots f_n\left(X\right)\right)\right)\right) \tag{1}$$

Here $f_1, f_2, \cdots f_n$ are the layers of the DNN. These can be functions with scalar, vector, matrix or tensor inputs and outputs. In the simplest case, the layer takes the form

$$f(\vec{x}) = \sigma(\vec{x}^T W + \vec{b}) \tag{2}$$

where the input, $\vec{x}$, is a column-vector, σ is a nonlinear function such as the sigmoid function, the tanh function, or rectifi-

cation, W is a matrix, and $\vec{b}$ is a bias vector. The layer described by Equation 2 is usually referred to as a densely-connected layer, since the matrix W defines some relationship between every component in $\vec{x}$ and every component in the output vector. The densely-connected layers usually form the final layers in a DNN, providing the output vector. There are many different types of popular DNN layers available today, each suited for specific tasks or sub-tasks within a DNN architecture. We will briefly review a few of the successful DNN layer types next.

Convolutional Neural Networks Convolutional Neural Networks (CNN) contain convolutional layers, which have a similar form as the densely-connected layers in that they apply a linear function followed by a non-linear function to the input. If X is a set of inputs $\{x_1, x_2, \cdots x_n\}$, then the convolutional layer produces an output set $\{y_1, y_2, \cdots y_m\}$ using the following operation:

$$y_i = \sum_{j=1}^{n} \sigma(x_j * W_{ij} + b_i) \tag{3}$$

where $*$ is the convolution operation, σ is a non-linear operation and W_{ij} is called the convolutional kernel. Both x_j and W_{ij} have the same number of dimensions, that is, if x_j is an image, W_{ij} is a two-dimensional matrix, and if x_j is a vector, W_{ij} is also a vector. W_{ij} is usually smaller than the input x_j.

If x_j is an image, the convolution operation slides W_{ij} along the length and height of x_j and at each step, produces an output element, by multiplying W_{ij} element-wise with the slice of x_j currently lined-up against W_{ij}, and summing up. This computation implies that if W_{ij} has the capacity to uncover a specific pattern, the exact position of the pattern in the image is not important (this property is more formally referred to as linear shift invariance).

CNNs also usually contain what are called pooling layers between different convolutional layers. A pooling layer, when applied to a 2D image, shrinks the image using local operations on neighboring elements in the image (e.g., average of neighboring elements). This structure allows CNNs to build intermediate outputs (or features) that are successively higher-level patterns that help the classification task.

Recurrent Neural Networks Recurrent Neural Networks (RNNs) attempt to find patterns in time, or in a sequence of data that have possible relationships among themselves. Towards this end RNNs generate and store state information after every element in the sequence is processed, and use this state information in the computation of the output corresponding to the next element in the sequence. For example, if the input sequence is a sequence of vectors, $\vec{x_i}$ is the i-th item in the sequence, W, $\vec{b}$ are parameters of the RNN layer, and $\vec{s}_{i-1}$ is the state information after processing the first $i-1$ items in the sequence, the RNN computes the output corresponding to the i-th input as

$$\vec{y_i} = \sigma((\vec{x_i}, \vec{s}_{i-1})^T W + b) \tag{4}$$

where $(\vec{a}, \vec{b})$ concatenates $\vec{a}, \vec{b}$. In the simplest case, the state information $\vec{s}_{i-1} = \vec{y}_{i-1}$.

The simplest RNNs do not function well when attempting to capture patterns set far apart in time. To do so, the state information must be computed and incorporated in a more sophisticated manner. Long-Short Term Memory (LSTM) and Gated Recurrent Units (GRU) are variations of the RNN that incorporate such extended state information. They include what are called "gates" to determine when the state information is updated, and whether to use the state information in computing the current output.

E. Accelerating workflows

Since many analytics in genomics have relevance to medicine, it is important to have fast execution. FPGAs [31] and GPUs [10][30] have been quite successful in obtaining higher performance and higher power. While FPGAs may be harder to program, there are techniques such as High Level Synthesis that can boost productivity [36]. Recent works in FPGA design [43][44] aim to accelerate Deep Neural Network workflows and these may be relevant to areas with high potential to use advanced machine learning methods such as genomics. Modern machine learning frameworks such as Tensorflow [4] provide highly programmer-friendly interfaces to unlock the power of optimized CPU and GPU libraries for training and testing Deep Neural Networks. Advances in Machine Learning theory as well as efficient implementation techniques for Deep Neural Networks are both highly relevant to big data analytics such as those in genomics.

III. Machine learning in Genomics

A. Machine learning methods in variant calling

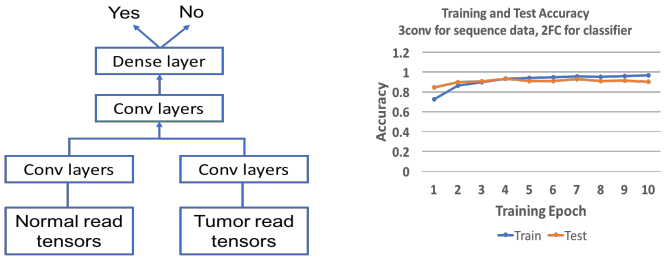
There have been recently some machine learning-based approaches that can handle variant calling of point substitutions and indels. SomaticSeq [14] merges calls from multiple somatic variant callers to determine putative locations of variants. For these locations, features regarding mapping and quality of reads are extracted from the sequencing data, which combined with the call results of the multiple tools, are used as inputs for a boosted decision tree type classifier. SNooPer [38] is another somatic variant caller that determines features necessary for variant calling and uses them as inputs for a Random Forest classifier. VariantMetaCaller [16] combines information from multiple germline variant callers using a Support Vector Machine to produce a variant call list. While these methods use Machine Learning techniques, they do not fall under the deep learning paradigm.

In contrast to these methods, DeepVariant [34] is a variant caller that applies Deep Learning to the problem of variant calling. DeepVariant collects read mapping data from a putative site with variation. It then obtains a pictorial representation of the local alignment between reads mapping to the site and the segment of the reference sequence surrounding the site as printed out visually (also referred to as pileup), and then encodes each base in the pileup with a different colour. This color-coded image is then passed through a CNN which predicts whether there is a variant at the site. DeepVariant is a marked deviation from the conventional approaches in which domain expertise is used to craft explicit tests or devise features to be fed to a classifier that are likely relevant to the problem domain. Instead, DeepVariant relies on the ability of the DNN to learn the necessary patterns for calling variants.

We propose to extend this approach further. Instead of posing the variant calling problem as an image recognition problem, we attempt to develop specialized representations for alignment data that can expose detailed similarity measures between the read sequence and the associated segment of the reference sequence. This representation will then be examined by a DNN to determine patterns that can help it classify the site as having a variant or not. The use of a detailed representation deviates from DeepVariant's pileup representation in that it doesn't provide the DNN classifier with "baked-in" alignments to look at. Instead it provides similarity measures that the DNN

can re-interpret internally as necessary. This difference is important since in many regions of the genome, the local mapping of a read to the reference segment, as output by a mapping tool, can be wrong and in many cases, a local remapping tool may not be able to correct for these mistakes [27].

We tested our assumptions in a somatic variant calling setting, where reads from normal and tumor tissues are available. Initially, the data is analyzed by weak callers that are highly sensitive to variant signals, but not specific in that the resultant callers have a large number of false positive calls. It is fairly simple to create such weak callers. Next, we create representations of reads from the filtered sites and pass them through a DNN. Figure 2a gives a high-level overview of our DNN architecture. At every site, CNNs analyze normal and tumor sequencing data to isolate patterns related to variant calling. The conclusions from the CNNs are combined and processed further by additional layers to determine the variant call confidence at the site. The initial results in Figure 2b show that our representation method can expose patterns related to variant calls and generalize them to unseen test examples. We are currently working on scaling our experiments up, and improving our call efficiency.



(a) DNN architecture (b) Train and test accuracy

Fig. 2.: Experimental implementation of DNN-based somatic variant caller

B. Machine learning methods in TF binding

Recently, there have been a few CNN-based approaches that can obtain higher performance than the traditional models in analyzing data related to TF-binding studies.

DeepBind [5] treats the input sequence as a vector. It then uses a convolutional layer where each convolutional kernel mimics a PWM. Each convolutional kernel is slid across the input sequence, and a measure of match between every segment of the input sequence and each PWM is generated. If there are N motifs (PWMs) being searched for in the input sequence, then there are N convolutional kernels used, and N outputs of the convolutional operations. The results of the N convolutions are each pooled along the length of the sequences (using a pooling layer), and fed into a densely connected layer which determines the binding affinity of the input sequence. Variations of the DeepBind architecture have been studied [42], and it has been found that other convolutional architectures can outperform the DeepBind model. Another work [17] uses a combination of convolutional layers and LSTM layers to determine TF binding affinity. In this case, the results of convolution are examined by an LSTM network. The LSTM network can extract patterns related to the arrangement of the motifs along the input sequence, in addition to determining whether the motifs are present in the input sequence.

As mentioned before, HMMs have been used for determining TF-binding sites in genome segments, as well as to determine the PWMs in a set of known binding locations. An HMM can perform a similar task as an LSTM in that it can

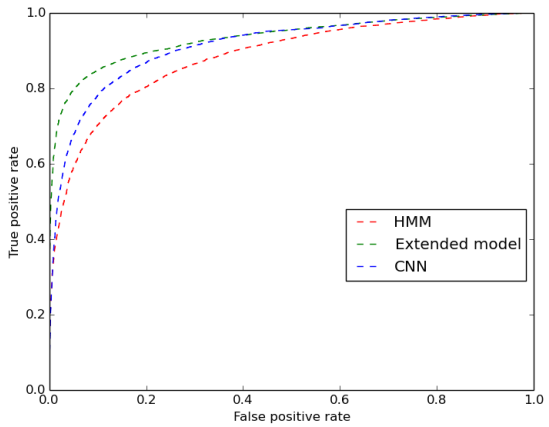


Fig. 3.: ROC curve comparing an HMM, our model, and a CNN

examine all types of arrangements of motifs along the input sequence. Previously, neural network architectures were explored in which an ensemble of HMMs draws features from input sequences, which are then used in a classification task using a densely-connected layer [9]. The densely-connected layers and the HMMs can be trained jointly in this framework. Using an HMM-based formulation has some advantages over using only advanced DNN layers. The HMM formulation places a definitive probabilistic meaning on the output of the model. This meaning allows for the model to be used to draw inferences outside of the classification framework. Layers in a CNN, or LSTM networks do not naturally lend themselves to stand-alone and rigorous interpretations, even though in the case of DeepBind, parallels may be drawn between PWMs and the convolutional kernels. In addition, a pure CNN-based approach may not generalize to all sequence-based problems, especially when the sequences have variable length, which is the case with a lot of the data in genomics (in such cases, RNNs are usually preferable).

The disadvantage of the HMM compared to an LSTM is that it may not be able to capture patterns set far apart in time. We are currently working to extend the HMM formulation to produce a model that is probabilistically meaningful and interpretable like an HMM, but is also efficient in examining patterns that have significant separation in time. Our model may be trained within the DNN framework, but can also be later used to perform other types of inferences, stand-alone. We set up an initial experiment comparing our extended model, and an HMM in a task classifying a segment of genomic sequence as having affinity for TF-binding or not. Both models are zero-th order models, implying that the transition probabilities do not depend on the current state. The experiment uses one of the datasets published in a previous work [42] for the motif discovery task. Figure 3 compares the performance of the HMM, our extended model, and a DeepBind-like CNN. It may be seen that our model significantly outperforms the HMM, while performing similar to the CNN. We plan to improve the scale of our model as well as that of the experiments in the future.

IV. Conclusions

High-throughput technologies are able to produce a large amount of genomic data at a low cost. The availability of large quantitites of data paves the way for using Deep Neural Networks and Deep Learning to analyze the data. Deep Neural

Networks have the capacity to learn complex patterns necessary for classification or regression problems, and allows specification of the problem through a representative sample of data from the problem domain. DNN formulations can replace expert knowledge, heuristics and simplifying assumptions, and this capability is important to obtain the next level of improvement in accuracy for genomics applications. These applications can have high impact in important areas of treatment and diagnosis such as in the case of cancer. Deep Learning approaches are emerging in some of these areas, but there is scope for further innovation.

REFERENCES

[1] COSMIC, http://cancer.sanger.ac.uk/cosmic.

[2] Genome Reference Consortium, https://www.ncbi.nlm.nih.gov/grc/human.

[3] Illumina sequencing platforms, https://www.illumina.com/systems.html.

[4] TensorFlow, https://www.tensorflow.org/.

[5] B. Alipanahi, A. Delong, M. T. Weirauch, and B. J. Frey. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nature Biotechnology*, 2015.

[6] M. J. Bamshad, S. B. Ng, A. W. Bigham, H. K. Tabor, M. J. Emond, D. A. Nickerson, and J. Shendure. Exome sequencing as a tool for Mendelian disease gene discovery. *Nature Reviews Genetics*, 2011.

[7] Y. Benjamini and T. P. Speed. Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic Acids Research*, 2012.

[8] A. S. Bhagwat and C. R. Vakoc. Targeting Transcription Factors in Cancer. *Trends in cancer*, 2015.

[9] S.-B. Cho and J. H. Kim. An HMM/MLP Architecture for Sequence Recognition. *Neural Computation*, 1995.

[10] Z. Cui, Y. Liang, K. Rupnow, and D. Chen. An Accurate GPU Performance Model for Effective Control Flow Divergence Optimization. In *2012 IEEE 26th International Parallel and Distributed Processing Symposium*, 2012.

[11] M. A. DePristo, E. Banks, R. Poplin, K. V. Garimella, J. R. Maguire, C. Hartl, A. A. Philippakis, G. del Angel, M. A. Rivas, M. Hanna, A. McKenna, T. J. Fennell, A. M. Kernytsky, A. Y. Sivachenko, K. Cibulskis, S. B. Gabriel, D. Altshuler, and M. J. Daly. A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nature Genetics*, 2011.

[12] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, 1998.

[13] S. R. Eddy. Profile hidden Markov models. *Bioinformatics Review*, 1998.

[14] L. T. Fang, P. T. Afshar, A. Chhibber, M. Mohiyuddin, Y. Fan, J. C. Mu, G. Gibeling, S. Barr, N. B. Asadi, M. B. Gerstein, D. C. Koboldt, W. Wang, W. H. Wong, and H. Y. Lam. An ensemble approach to accurately detect somatic mutations using SomaticSeq. *Genome Biology*, 2015.

[15] A. Graves, A. Mohamed, and G. E. Hinton. Speech recognition with deep recurrent neural networks. *CoRR*, 2013.

[16] A. Gézsi, B. Bolgár, P. Marx, P. Sarkozy, C. Szalai, and P. Antal. VariantMetaCaller: automated fusion of variant calling pipelines for quantitative, precision-based filtering. *BMC Genomics*, 2015.

[17] H. R. Hassanzadeh and M. D. Wang. DeeperBind: Enhancing prediction of sequence specificities of DNA binding proteins. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2016.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. *CoRR*, 2015.

[19] S. Hodgson. Mechanisms of inherited cancer susceptibility. *Journal of Zhejiang University Science B*, 2008.

[20] Y. Jiao, H. Zhao, L. Ren, W. Song, B. Zeng, J. Guo, B. Wang, Z. Liu, J. Chen, W. Li, M. Zhang, S. Xie, and J. Lai. Genome-wide genetic changes during modern breeding of maize. *Nature Genetics*, 2012.

[21] R. Kamps, R. D. Brandão, B. J. van den Bosch, A. D. C. Paulussen, S. Xanthoulea, M. J. Blok, and A. Romano. Next-Generation Sequencing in Oncology: Genetic Diagnosis, Risk Prediction and Cancer Classification. *International Journal of Molecular Sciences*, 2017.

[22] P. Kharchenko, M. Tolstorukov, and P. Park. Design and analysis of ChIP-seq experiments for DNA-binding proteins. *Nature Biotechnology*, 2008.

[23] D. Koboldt, Q. Zhang, D. Larson, D. Shen, M. McLellan, L. Lin, C. Miller, E. Mardis, L. Ding, and R. Wilson. VarScan 2: Somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Research*, 2012.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. NIPS, 2012.

[25] Kurt Hornik. Multilayer Feedforward Networks are Universal Approximators. *Neural Networks*, 1989.

[26] B. Langmead and S. L. Salzberg. Fast gapped-read alignment with Bowtie2. *Nature Methods*, 2012.

[27] H. Li. Toward better understanding of artifacts in variant calling from high-coverage samples. *Bioinformatics*, 2014.

[28] H. Li and R. Durbin. Fast and accurate short read alignment with Burrows-Wheeler Transform. *Bioinformatics*, 2009.

[29] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, and 1000 Genome Project Data Processing Subgroup. The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 2009.

[30] Y. Liang, H. P. Huynh, K. Rupnow, R. S. M. Goh, and D. Chen. Efficient GPU Spatial-Temporal Multitasking. *IEEE Transactions on Parallel and Distributed Systems*, March 2015.

[31] S. Liu, A. Papakonstantinou, H. Wang, and D. Chen. Real-Time Object Tracking System on FPGAs. In *2011 Symposium on Application Accelerators in High-Performance Computing*, 2011.

[32] Martha L. Bulyk. Protein binding microarrays for the characterization of DNA-protein interactions. *Advances in biochemical engineering/biotechnology*, 2007.

[33] A. McKenna, M. Hanna, E. Banks, A. Sivachenko, K. Cibulskis, A. Kernytsky, K. Garimella, D. Altshuler, S. Gabriel, M. Daly, and M. A. DePristo. The Genome Analysis Toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Research*, 2010.

[34] R. Poplin, D. Newburger, J. Dijamco, N. Nguyen, D. Loy, S. Gross, C. Y. McLean, and M. DePristo. Creating a universal snp and small indel variant caller with deep neural networks. *bioRxiv*, 2016.

[35] C. S. Ross-Innes, R. Stark, A. E. Teschendorff, K. A. Holmes, H. R. Ali, M. J. Dunning, G. D. Brown, O. Gojis, I. O. Ellis, A. R. Green, S. Ali, S.-F. Chin, C. Palmieri, C. Caldas, and J. S. Carroll. Differential oestrogen receptor binding is associated with clinical outcome in breast cancer. *Nature*, 2012.

[36] K. Rupnow, Y. Liang, Y. Li, D. Min, M. Do, and D. Chen. High level synthesis of stereo matching: Productivity, performance, and software constraints. In *2011 International Conference on Field-Programmable Technology*, 2011.

[37] S. Sinha, E. van Nimwegen, and E. D. Siggia. A probabilistic method to detect regulatory modules. *Bioinformatics*, 2003.

[38] J.-F. Spinella, P. Mehanna, R. Vidal, V. Saillour, P. Cassart, C. Richer, M. Ouimet, J. Healy, and D. Sinnett. SNooPer: a machine learning-based method for somatic variant identification from low-pass next-generation sequencing. *BMC Genomics*, 2016.

[39] The 1000 Genomes Project Consortium. A global reference for human genetic variation. *Nature*, 2015.

[40] B. Vogelstein, N. Papadopoulos, V. E. Velculescu, S. Zhou, L. A. Diaz, and K. W. Kinzler. Cancer Genome Landscapes. *Science*, 2013.

[41] B.-J. Yoon. Hidden Markov Models and their Applications in Biological Sequence Analysis. *Current Genomics*, 2009.

[42] H. Zeng, M. D. Edwards, G. Liu, and D. K. Gifford. Convolutional neural network architectures for predicting DNA-protein binding. *Bioinformatics*, 2016.

[43] C. Zhang, P. Li, G. Sun, Y. Guan, B. Xiao, and J. Cong. Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks. In *Proceedings of the 2015 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, FPGA '15, New York, NY, USA, 2015. ACM.

[44] X. Zhang, X. Liu, A. Ramachandran, C. Zhuge, S. Tang, P. Ouyang, Z. Cheng, K. Rupnow, and D. Chen. High-performance video content recognition with long-term recurrent convolutional network for FPGA. In *2017 27th International Conference on Field Programmable Logic and Applications (FPL)*, 2017.

[45] Y. Zhao and G. D. Stormo. Quantitative analysis demonstrates most transcription factors require only simple models of specificity. *Nature Biotechnology*, 2011.