Analyzing Genomics using Deep Learning

Introduction

Genomics aims to characterize the function of all genomic elements of an organism[1]. Applications for functional genomics ranges from finding biomarkers[2], discovering association between genotype and phenotype[3] and predicting function of genes and genomic elements, such as enhancers[4]. Since the time of next-generation sequencing, sequencing methods and transcriptomics profiling techniques have become more elaborate and inexpensive to use, causing the amount of data available on genomics to have grown exponentially in the last years[5]. Additionally, newer sequencing technologies such as Chromatin Immunoprecipitation sequencing (ChIP-seq)[6], Assay for Transposase-Assessible Chromatin sequencing (ATAC-seq)[7] and Chromosome Conformation Capture (Hi-C)[8] have shed light on the epigenomic landscape, DNA-protein interactions and 3D DNA structure. Together with conventional sequencing, these methods have generated an abundance of information for the investigation of functional genomics.

The scale of genomics data highlights the importance of automated data mining methods, as they are too large to investigate by hand. Machine learning algorithms excel at finding patterns in large datasets and have therefore been broadly applied in genomics for the annotation and interpretation of genomic sequence elements. For instance, machine learning algorithms can be trained to recognize many different elements, such as splice sites[9], enhancers[10], and transcription start sites[11]. However, conventional machine learning algorithms strongly rely on a priori determined variables in the data, known as features. For example, finding enhancers is mainly achieved by searching for transcription factor binding sites, which are a set of consecutive nucleotides also known as a motif on which transcription factors can bind. In traditional machine learning, these motifs, which are the features of the data, have to be manually defined and extracted from the input before passing these extracted features to the machine learning algorithm. Deep learning models, which are a subset of machine learning algorithms, can learn these features themselves directly from the input data, creating an 'end-to-end model'[12]. Furthermore, these algorithms are able to improve feature complexity. In the example above, deep learning models would be able to identify improved motifs and discover new features for a specific dataset.

Currently, deep learning approaches are outperforming conventional algorithms in many fields such as computer vision, natural language processing and information retrieval[13]. Similarly, deep learning has successfully been used in healthcare and bioinformatic research. For instance, clinicians can expect deep learning applications for the identification of brain- and breast tumors from respectively MRI and mammogram in the near future[14,15]. Likewise, the first neural networks for motif analysis were already presented in 2015[16–18]. Since then, the applications for deep learning models in bioinformatics and genomics have grown immensely[19]. As of now, applications in genomics for deep learning models are present in base calling[20], small variant calling[21], DNA motif analysis[17], RNA analysis[22], DNA accessibility and chromatin[23], DNA methylation[24], DNA-protein interaction and pathogenic variant scoring[25].

In this review, we will describe the current state-of-the-art deep learning modeling techniques deployed in genomics. First, we start by summarizing both supervised and non-supervised deep learning models as well as explain the key concepts in deep learning. Next, we will discuss different areas in genomics and their applicability for deep learning approaches. Thereafter, we discuss the main areas of genomics we consider as promising fields for deep learning applications, which are variant calling, predicting regulatory signatures, pathogenic non-coding variant scoring, denoising and data augmentation. This review will mainly focus on functional genomics, in previous work Rang et al., 2018, have discussed the application of deep learning models in nanopore sequencing base calling[20]. For a more in-depth overview and guide on deep learning, we would recommend reading a primer and several reviews, in which deep learning models is more technically discussed[26–29].

Machine learning and deep learning modeling

Machine learning algorithms can be roughly divided between supervised and unsupervised methods. In supervised machine learning, labeled data is used to predict the classification or regression of data points. Examples of conventional supervised machine learning algorithms are linear and logistic regression, the random forest classifier and support vector machines (SVM). On the other hand, in unsupervised machine learning, patterns in data are learned without relying on predefined labels. Two examples of unsupervised machine learning are clustering and principal component analysis.

Deep learning, which is a subclass of machine learning, is best described as a series of algorithms to uncover underlying relations in data, also referred to as features, which in traditional machine learning have to be defined beforehand. In general, genomics is well suited for deep learning approaches as many areas in genomics harness high dimensional, noisy data with nonlinear relationships, which deep neural network excel at in comparison to conventional machine learning approaches. Furthermore, unlike statistical models, deep learning is highly generalizable, indicating that same models are applicable for many problems in genomics. For instance, variant calling algorithms are devised for a specific sequencing method, while a single deep learning model could infer variants for all sequencing methods equally. Another example is a classification problem with data which cannot be linearly separated and is therefore unsolvable with conventional machine learning such as logistic regression (Figure 1A). To solve this problem, we can assess a simple neural network, known as a multilayer perceptron. In this model, every neuron of one layer i are connected to all the neurons of the following layer $i + 1$, with each connection having a different trained weights or parameters $w_{i,j}$. Each neuron computes the weighted sum of its input and applies a non-linear activation function to calculate its outcome. Coming back to the example, using these series of nonlinear transformations, we can distinguish the two different labels (Figure 1A).

Supervised learning

In order to train supervised neural networks, labeled data first has to be partitioned into three parts: A training set, validation set and a test set. The training set is used to learn the model weights. Next, the validation is assessed to test the model and tune hyperparameters, discussed later. The ultimate goal of a machine learning algorithm is to achieve high accuracy on predicting labels in the test set. Notable, the hold-out test set should be a separate dataset that is used for training and validation. The accuracy of the model is measured by calculating the difference between the predicted labels and true labels in a loss function. In order to increase accuracy, this loss function has to be minimized. In our classification example, minimizing the loss function would be assigning data points to right groups. The two main types of loss functions used in neural networks are cross-entropy and mean squared error. Cross-entropy calculates the difference between the predicted distribution and the true distribution, which is used for binary and multi-class classification problems[30]. Mean squared error calculates the average squares difference between the estimated and true values,

applied for regression problems[30]. Data partitioning is an important part of machine learning and should be carefully selected to account for class imbalance and sample representativeness biases[31].

Training a neural network focusses on finding the 'best' weights for predicting the labels of the test data. To train a neural network, first weights between each neuron are initialized with random values. Next, these weights are iteratively updated using gradient descent, an optimization algorithm used for finding the minimum of the loss function. This minimum is achieved when the model can predict the training data as best as possible. In order to reach this minimum the weights are updated based on the gradient in the network which is calculated using backward propagation algorithm, based on the uses of the chain rule for derivatives[30]. Loss function minimization is difficult as this function is often high-dimensional and non-convex, best explained as a landscape with hills and valleys. This can result in the algorithm to get stuck in local minima (i.e. a valley which is not the lowest point in the landscape), and unable to converge to the global minimum, also referred to as 'trapping'. There are several types of gradient descent algorithms: batch, stochastic and mini-batch[12]. Firstly, batch gradient descent iteratively uses a whole dataset before calculating the loss function and converging its weights. Secondly, stochastic gradient descent uses this optimization after each data point. Lastly and most widely applied, mini-batch gradient descent uses small subsets of the dataset, referred to as batches, in order to convergence to the minimum of the loss function, the state in which the model can best predict the labels. Unlike other algorithms mini-batch gradient descent suffers less from overshooting and trapping in local minima[32]. Neural network training is monitored by evaluating the loss function on the validation set. In order to improve neural network accuracy, several hyperparameters can be tuned, such as number of neurons and layers, type of activation functions and learning rates for gradient descent[33]. Briefly, several activation functions, such as sigmoid, tanh and ReLU, can be used to transform the output of each neuron differently[34]. Learning rates determine how much to update the weights based on the estimated error, with larger values for the learning rate, being faster but possibly over-shooting the global minima and smaller values for the learning rate, being more carful but slower and possibly getting stuck in local minima[35].

Aside from multilayer perceptron, numerous different neural network structures can be assessed for supervised learning. A convolutional neural network (CNN), mostly known for

its applications in computer vision, scans a filter on every position, thereby accounting for spatial dependencies in the data (Figure 1B). This is often the case in genomics, for example with DNA motifs for transcription factors or interactions in Hi-C data. Recurrent neural networks (RNN) process time-series or sequential data and feed the signal through the next layer of the RNN (Figure 1C), similar to a feed-forward network. Neurons in a recurrent network function as memory that retain information from the previous state, for instance the previous nucleotide of a sequence, and update this memory state to the next neuron. Similar to CNN, RNN function well where sequential dependencies in data consists.

Unsupervised learning

In additional to supervised deep learning, several unsupervised deep learning approaches are as well deployed in genomics. These models, rather than training to predict the outcome value from the input, try to model hidden structures without instructions on required outcome data. For example, Autoencoders are neural networks designed for dimensionality reduction techniques (Figure 1D). In contrast to principal component analysis, autoencoders are trained to find the input representation in low-dimensional space which is still able to reconstruct the original information. Data dimension reduction is achieved by creating a 'bottleneck' in the neural network, in which less neurons are available in comparison to the input and output. Aside from autoencoders application for dimensionality reduction, it can also be helpful for denoising, as only important features are retrained through the network. Another unsupervised deep learning model is a Generative adversarial network (GAN), which consists of two neural networks, one trying to distinguish real data from artificially created data and another network which creates artificial data from random noise (Figure 1E). The competition between the two networks, enables the construction of artificial data from random noise which is highly representative to the original dataset.

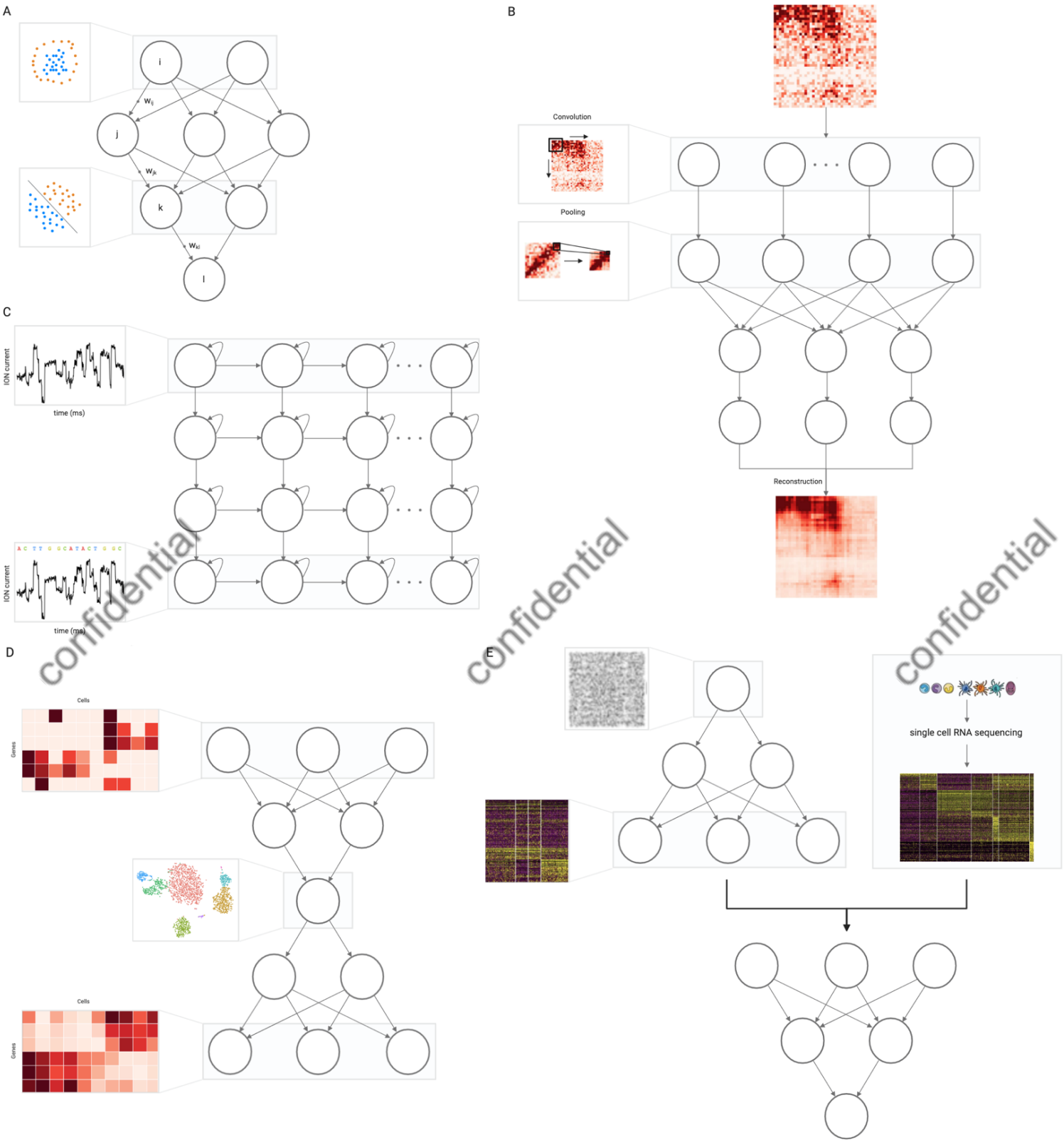

Figure 1: Deep learning approaches applied to genomics. *A-C) Supervised deep learning. A) Multilayer perceptron, a series of non-linear transformations to create linear separable data. The multilayer perceptron consists of input layer (i), two hidden layers (j,k) and an output layer (l). In these layers the weighted sum of the previous neurons is calculated based on the learned parameters (w_{ij}, w_{jk}, w_{kl}), transformed with an activation function and passed on to the next layer. The last layer outputs the predicted labels of the data points. In classification problems, the last activation is a sigmoid function (2 groups) or softmax (>2 groups). B) Convolutional network, each neuron in the first layer depicts a filter run over the picture, in this example a part of a Hi-C map[75]. Second layer is a maxpool layer, which takes the maximum value in the square. Pooling layers can have several functions, such as maxpool, minpool or averagepool. The following layers, which can be as deep as possible, represent convolutional and sequentially pool layers. In this example, Hi-C data is reconstructed with higher accuracy[75]. C) Recurrent neural network can handle time series data such as nanopore raw data to predict base calling[88,89]. Memory state of the neurons are passed on to following time series, to handle sequent information. There are different types of recurrent cells, such as long short-term memory, which is most commonly used[45]. Note, that often hybrid models are used, which have both convolutions, recurrent as well as fully connected layers. D-E) Unsupervised deep learning. D) Autoencoder, makes use of a bottleneck in order to learn to compress data and still be able to recover data integrity. In this example, single cell RNA sequencing is compressed in a lower dimension to enable efficient cell clustering, similar to principal component analysis. E) Generative adversarial networks make use of a generator, to create artificial data from noise and a decoder which tries to distinguish, in this example, real single cell RNA data from artificially created data. Competition between the two networks drive the likeliness of the generator created artificial data to real world data[90].*

Single nucleotide polymorphism and small variants calling

To identify single nucleotide polymorphisms (SNPs) and indels several toolkits can be deployed, such as GATK and Strelka[36,37]. These toolkits use a set of statistical models to identify both germline and somatic small variants. For instance, the commonly used GATK makes use of logistic regression and hidden Markov model to recalibrate base quality score, Bayesian classification algorithm for variant identification and Bayes Gaussian mixture model (GMM) for variant quality score recalibration in order to filter false positives[38]. Altogether, these algorithms achieve high accuracy on Illumina sequencing platforms[36]. However, several benchmark papers highlight imperfections in these methods, such as regions which there is no genotype caller for, sensitivity issues with detecting somatic variants and high false positive rates even when used in an ensemble, which is an combination of methods including merging and filtering[39–41]. Additionally, generalizing these algorithms for other sequencing technologies has proven to be difficult.

Several deep learning approaches have been developed to tackle small variant identification. DeepVariant uses the pre-trained google inception CNN, which has been widely applied in computer vision software, to investigate candidate variant positions (Figure 2B)[42]. Originally, variant call image snapshots were encoded in a red-green-blue (RBG) pileup image, consisting of reference and read bases, quality score and other read features. However, as described by Ryan Poplin et al., 2018, their method of data processing is suboptimal as they were unable to encode all available read information in these three layers. Therefore, the open source software aside from using the newer inception_v3 network now uses a

multichannel tensor representation, which encodes all the read information in several additional layers, to overcome this limitation. DeepVariants final output layer is a three-class softmax layer to infer genotype likelihood (Hom-ref, Het, Hom-alt). In general, DeepVariant is able to achieve similar results to conventional variant calling algorithms with high sensitivity but still low specificity. Additionally, the neural network is broadly applicable for several sequencing technologies, such as Illumina, SOLiD, PacBio and Ion Torrent.

Clairvoyante a multitask five-layer CNN specifically developed for variant calling for single molecule sequencing technologies also showed to achieve high accuracy on small variant identification[43]. Importantly, Clairvoyante was able to identify 3135 additional variants using independently PacBio and Oxford Nanopore Technology (ONT) sequencing missed by Illumina sequencing. Similar to DeepVariant, candidate positions were transformed into a multidimensional tensor. However, instead of using read images, read information is transformed into four one-hot encoded matrices in order to limit the input dimensions and required neural network depth. The multitask model outputs the zygosity of the variant, type of variant and length of Indel if applicable. Recently, Clair, the successor of Clairvoyante has been released, which uses a RNN to uncover SNPs and Indels (Figure 2C)[44]. Using bidirectional long short-term memory (bi-LSTM) layers, Clair is able to solve limitation from its predecessor including multiallelic variant calling and long indel calling. Briefly, LSTM's are a special kind of recurrent cells, which pass information from one state to the next, and are deployed as they solve issues with vanishing and exploding gradients, often displayed in larger recurrent networks[45]. In this case of read information, longer indels which span multiple nucleotides can be inferred as the information of previous position can flow through the network. Lastly, bi-LSTM memory cannot exclusively flow from start to end but also from end to start in comparison to LSTM layers. Additionally, the newer Clair model has around two and a half million parameters to learn which is nearly double that of Clairvoyante but still a tenth as many as DeepVariant. Clair uses the same one-hot encoding as input data to analyze genomic data with additional tensors for more sequence information. Although Clair is achieving near perfect precision and recall on ONT, it is still outperformed by DeepVariant on PacBio and Illumina.

Lastly, NeuSomatic uses a deep CNN inspired by ResNet to identify specifically somatic small variants (Figure 2D)[46–48]. NeuSomatic was able to dramatically outperform existing somatic variant callers at all different tumor purity scenarios and allelic frequencies as well as

different sequencing technologies, including Illumina and PacBio. Notable even for NeuSomatic, somatic variant calling remains difficult when tumor purity provided is low. Comparable to Clairvoyante, candidate positions are summarized in a single base frequency matrix in order to represent different types of small variants. Additionally, more than hundred features are incorporated in a multidimensional tensor to capture mutation signal effectively, including normal and tumor coverage. Although NeuSomatic performs well, we would still advise to do somatic variant calling with an ensemble of existing methods to achieve the highest possible accuracy.

In general, while SNP and indel calling can acquire high accuracy using one of the variants calling neural networks, these methods do not drastically increase the detection of variants in comparison to traditional variant calling methods. Furthermore, these deep learning models still require some features extraction beforehand, which are mapping and tensor buildup. As described before, one of the main advantages of deep learning should be the removal of this initial feature extraction. Nevertheless, deep learning methods for variant calling do benefit from their widely applicability on several sequencing platforms, making them more generalizable than traditional methods.

Aside from variant calling, deep learning approaches have been investigated for automated refinement of somatic variant calling[49]. Various different types of cancer sequencing data were used to a train logistic regression model, random forest model and feed-forward neural networks, consisting of four hidden layers with 20-nodes. In conclusion, machine learning in general could accurately recapitulate manual refinement, with both the random forest as well as the neural network to acquirer high accuracy.

Structural variants calling

Unlike small variants where accuracy of variant caller reaches near perfect standards already with conventional statistical and machine learning approaches, identifying structural variants (SVs) remains hard to this day[50]. Benchmarking of SV calling algorithms, such as Manta, Delly and Lumpy, depicts the inability of these algorithms to consistently call these variants[51,52]. Additionally, analyzing the SVs identified with these algorithms highlight the discrepancy between them, as more often than not, SVs are strictly identified by just one of the algorithms. Therefore, an ensemble of SV detection algorithms, in which multiple callers are combined, shows to be most promising for SV calling. However, this process provides to be

tedious and requires additional filtering and consensus generation using various statistical models[50]. Nevertheless, recently an SV caller ensemble has been devised for parallel SV calling and data post-processing[53].

To our knowledge, DeepSV is currently the only deep learning approach available for deletion calling (Figure 2E)[54]. Similar to previous deep learning approaches, candidate positions are selected using k-means clustering and processed into pileup images. Reads are encoded in a red-green-blue color map with 256 choices for each color. Aside from base information, these color maps can encode various different read information such as split reads, quality score and discordant read pair. Pileup images are partitioned into 50 bp non-overlapping windows and run through a CNN to predict specifically if the candidate location has a deletion. Notable, DeepSV looks throughout the whole deletions instead of border regions which could be computational costly as deletions could be thousands of bases long, resulting in an excessive number of images to analyze. DeepSV in comparison to conventional SV callers has increased accuracy especially with higher coverage sequencing data.

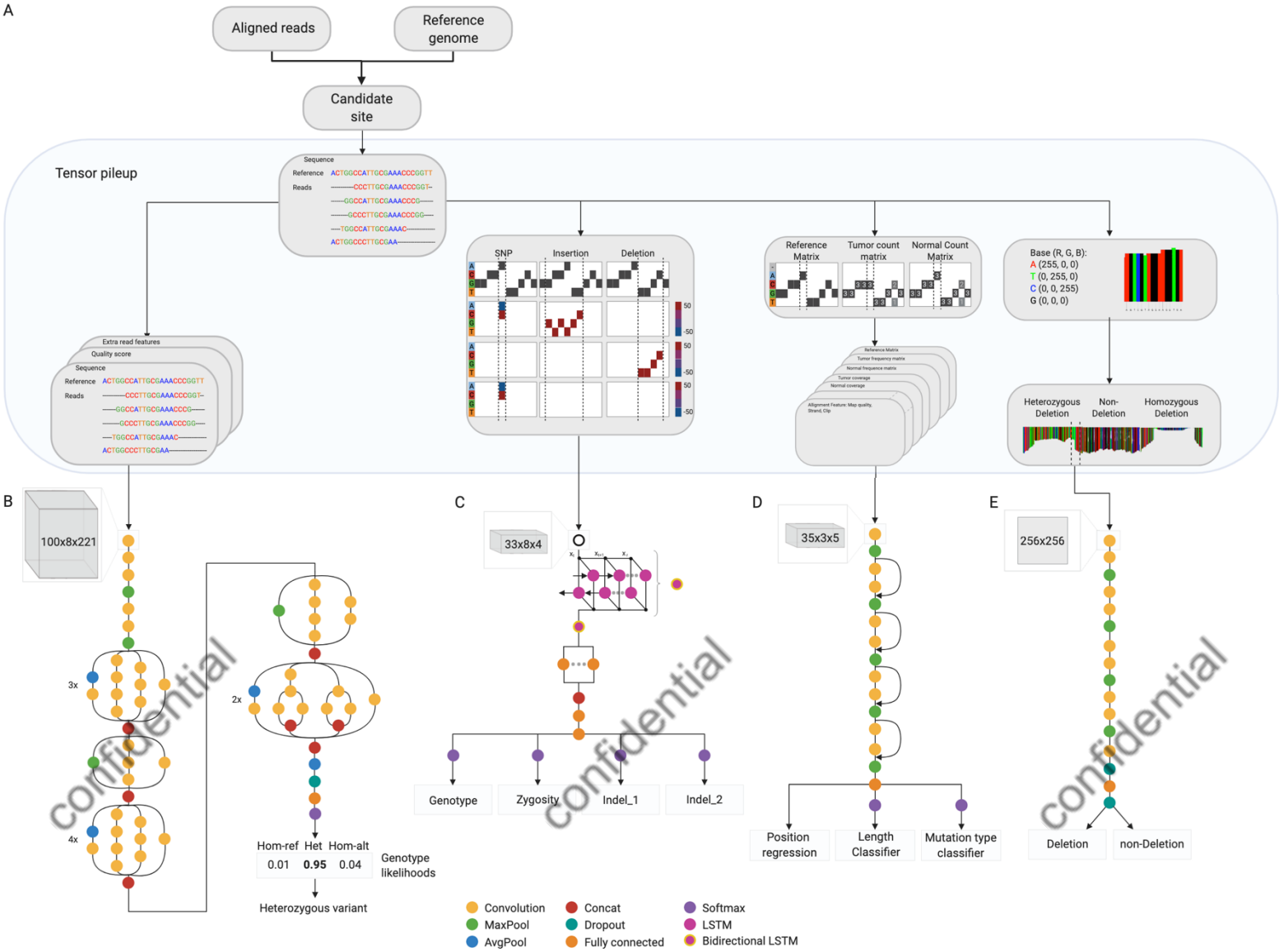

Figure 2: Variant analysis using different deep learning approaches. *A) Candidate sites are selected using various techniques such as K-means clustering. Next, various methods are deployed for data processing. B) DeepVariant uses a pileup of images into a multidimensional tensor[42]. For each candidate position reference sequence and all the other reads are captured, as well as encoding tensors for read quality and other features, resulting in a 100 wide, 221 height pixels with as default 8 tensors. The pretrained inception_v3 network is used and refined on high confidence calls from NA12878, NA24385 and eight datasets from Genome in a Bottle[91]. Inception cells accelerate in using less computational power due to the limited number of matrix multiplications required. This model is able to predict genotype likelihoods of variants and thereby predict and identify single nucleotide polymorphisms. C) Clair uses a different approach for data processing. Candidate position reads are encoded in a one-hot matrix, with the first tensor corresponding to the reference genome. The next three tensors use relative counts against the reference, with the second tensor encoding the inserted sequence, the third the deleted and forth the alternate allele. Clair unlike Clairvoyante uses bi-directional long short-term memory (LSTM) cells. These excel and finding spatial relationships, such as with longer indels. LSTM cells work well in this model due to the limited input sequence of only 16 flanking, in total 33 nucleotides. Four softmax function determine, genotype, zygosity, first indel and second indel length. D) NeuSomatic, uses a similar one-hot encoding into a reference matrix, tumor count matrix and normal count matrix. Additionally, multiple tensors can be used to encode various sequence information such as map quality, strand and clipping. NeuSomatic makes use of a ResNet like neural network in order to avoid vanishing gradients caused by deeper networks. NeuSomatic outputs the position as a sanity check and uses two softmax functions to identify type of mutation and length in case of indels. E) DeepSV encodes reads into RGB color maps, in which additional information can be encoded as well. Similar to DeepVariant, snapshots of these RGB maps of 50 bps are taken at candidate positions. These snapshots consist of 256 by 256-pixel images, which are run through a simple convolutional network.*

Predicting regulatory signatures

DNA-protein interactions have several important functions in genomics, such as regulating gene expression, 3D genome structure and epigenomic modifications. The DNA-protein interactions occur on specific sequence known as motifs. These DNA motifs are normally identified using ChIP-sequencing data with which you can sequence DNA regions bound to your protein of interest. ChIP-sequencing reads are transformed in a position probability matrix (PPM), which encodes the probability of a specific nucleotide occurring on each position of the captures reads. Afterwards, these PPMs are transformed into position weight matrixes (PWM), which log transform PPMs and account for background probability of a nucleotide occurring in the sequence. PWM likeness to region of interested are compared to identify novel transcription factor binding sites, using a 'sliding window' approach. Deep learning approaches can also be deployed for analyzing specific DNA motifs (Figure 3A). Generally, these networks use a one-hot encoding, which transforms a DNA sequence into a binary matrix. Next, CNN filters learn to recognize motifs similar to scanning the DNA sequence with a PWM. Training the network can be achieved by using several DNA or RNA binding databases such as PBM, RNAcompete, ChIP-seq and HT-Selex. Thereby, it is able to analyze the DNA for all sorts of learned patterns. Simple CNNs, which can be as shallow as one-layer convolution, are already able to detect various motifs, such as enhancers, RNA structures and chromatin features[55–58]. Furthermore, some deeper networks are able to identify a larger number of features, such as DeepSEA which utilities two additional convolution and a fully connected layer to predict 919 chromatin features including DNase, Transcription Factor and histone features[17,59]. Notable, deeper CNNs or additional fully connected layers did not seem to improve accustomed motif analysis[56]. Interestingly, deep learning approaches have identified new DNA motifs and also showed to acquire higher accuracy in identifying transcription factor binding sites in comparison to conventional models[55].

Simple networks for motif analysis can be expanded to discover more complex or a large number of features from sequencing data. First, convolution parallelization of both the forward and reverse strand is able to further improve motif prediction[60]. Secondly, additional information about cell specific transcription factors can be fed to the neural network using gating in order to more accurately predict enhancer activity[61]. Thirdly, motifs, especially enhancers, often follow a specific spatial arraignment and occur frequent in combinations with other motifs[62,63]. Therefore, a hybrid neural networks, which includes LSTM or Bi-LSTM

layers after convolution provides improved enhancer prediction, as memory of previous motifs flows through the network (Figure 3B)[64,65]. Similarly, longer spatial interactions, such is the case with enhancer and promoters interactions can also be inferred using parallel convolutions and subsequently a connected recurrent layer uncovering long-range dependencies[66]. Furthermore, several networks are able to accurately predict higher complex features, such as ChIP-sequencing, DNase-sequencing and Cap Analysis Gene Expression (CAGE) sequencing data and thereby are able to mimic original sequencing data (Figure 3C)[23,67–69].

The ultimate goal of assessing regulatory signatures is the prediction of gene expression. Earlier advancements in gene expression prediction with neural networks came by analyzing histone modification around transcription start sites (TSS) with DeepChrome, which uses a comparable one-hot encoding of five histone modification in bins of 100 bp around a genes TSS and analyzed these signals with a convolutional network[70]. Recently, deep learning approach ExPecto, which is the successor of DeepSEA, showed to accurately predict cell-specific gene expression (Figure 3D)[17,25]. First, a CNN identifies and mimics 2,002 histone marks, chromatin features and TF features (DeepSEA predicted 919 features). Next, these features are summarized using spatial transformation to reduce dimensionality depended on the relative distance to the TSS. Following, using tissue expression profiles, cell-specific gene expression could be predicted using a regularized linear model. Similarly, Basenji is equally able to cell-specifically predict gene expression using epigenomic features derived from a more extensive convolutional network of its predecessor Basset[67]. To our knowledge, no benchmarking between these two methods have been conducted.

In future, additional information could be applied for the prediction of gene expression, such as DNA modifications and 3D genome structure. Several deep learning approaches have already been applied for the identification of DNA methylation profiles[24,71–73]. For instance, the use of hybrid bidirectional recurrent networks to infer DNA methylation from long-read nanopore sequencing[71] and single cell bisulfite sequencing[73]. Alternatively, convolution networks as seen before are also able to learn DNA methylation motifs from whole-genome bisulfite sequencing[24]. On the topic of 3D genome prediction, only few deep learning approaches have been devised to analyze or predict Hi-C data[74,75]. Nevertheless, CNNs have been deployed for increasing the Hi-C mapping resolution[75] (Figure 1B) and predicting 3D chromatin architecture[74].

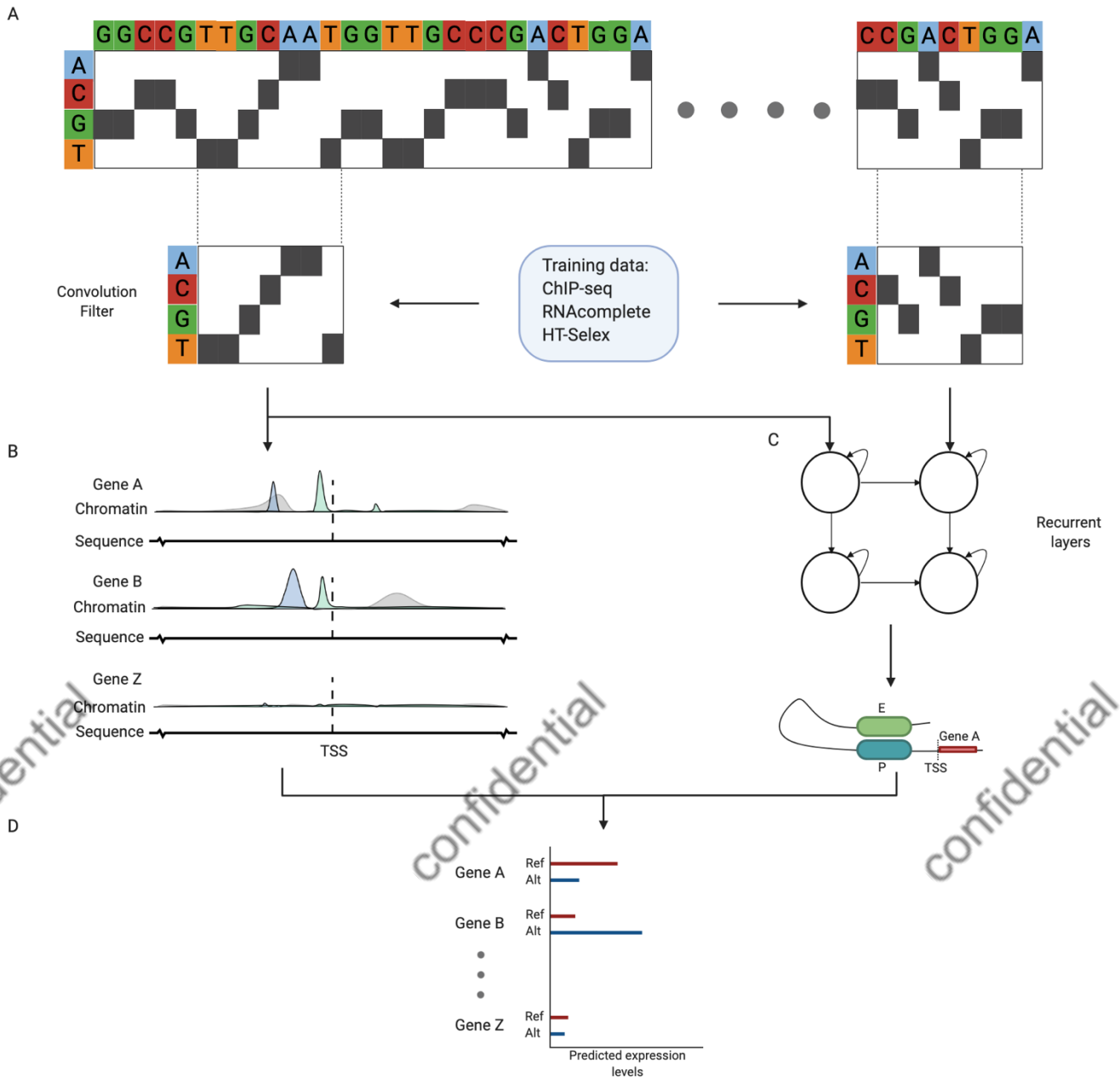

Figure 3: Predicting the regulatory landscape using deep learning. *A) Convolutional neural networks can be trained to recognize DNA motifs from several different training set. Each learned motif is encoded in a convolution filter which scans the one-hot encoded sequence for matching patterns. B) These convolutional neural networks can be extended to identify multiple different chromatin marks such as well-known gene activation marks H3K4me3, H3K27ac and H3K79me2. C) parallel convolutional networks connected with recurrent layers can accurately predict enhancer – promoter interaction. D) Features extracted with the convolutional neural networks can be used to predict gene expression levels[25]. Additionally, using the same model, differential gene expression can be inferred from altered chromatin features for non-coding variants. Figures are adjusted from Jian Zhou et al., 2018[25].*

Predicting the pathogenicity of non-coding variants

Variant effect predictions are important for both functional analysis and disease prediction. For coding variants, pathogenic effect prediction is often simpler, as changes at the protein level can be easily implied, therefore several models are able to achieve high accuracy for the prediction of coding variants[76–78]. On the other hand, non-coding variants which can have

effect on regulatory signatures, such as enhancers are much more difficult to infer due to the shear distance to an effector gene. Additionally, the disruption of other protein-binding sites, could have various effects, like the addition of CTCF sites, altering genomic spatial interactions. In order to determine non-coding effects several machine learning approaches, including deep learning, have been deployed[16,79–82]. These methods use functional genomic annotations from DNase sequencing and ChIP-seq using logistic regression[79], support vector machines (SVM)[80,81] or a neural network[16] to make predictions on underlying variants in the data. Additionally, all methods aside from deltaSVM make use of conservation scores to determine the effect of a variant, as these regions are more likely to be functionally important as they have been conserved throughout evolution. Similar to SV calling, for the identification of non-coding variant effect, ensemble methods, which uses a variety of algorithms, can be employed[83]. However, as comparative studies have shown, predicting effect of non-coding variant is far from perfect, with accuracy fluctuating between datasets[84,85]. Moreover, predictive performance of 18 deleteriousness-scoring methods showed high variability between methods[85]

In the previous segment we summarized several methods to identify motifs and regulatory signatures using deep learning models. Many of these models provide additional non-coding variant influence scores. For example, disruption of DNA motifs and thus altered DNA-protein interaction can be inferred with these models, for each nucleotide substitution[55,65]. Similarly, changes in higher complex features, such as chromatin, ChIP-seq and DNase maps can be determined for variants in comparison to reference[17,23]. Furthermore, end-to-end gene expression prediction models, such as ExPecto and Basenji, predict altered gene expression profiles for non-coding variant (Figure 3D)[25,67]. Notably, the ExPecto network has revealed the role in disease of non-coding variants in 1790 patients with autism spectrum disorder (ASD)[86]. In general, it would be interesting to see these methods of non-coding variant prediction implemented in future benchmark papers and ensemble methods.

In order to fully determine the mechanistic functions of non-coding variants in disease, experimental validation should be conducted on putative causal variants. The opportunity for deep learning approaches lies in the improved selection of these in silico important variants and therefore reducing the number of variants for experimental follow-up. In the future, the expansion of labeled high quality datasets as well as case studies should

improve the predicting power of deep learning variant scoring. Finally, different functional genomic annotation, such as 3D interactions and DNA modification, should be considered for variant prediction, which are currently also not applied in conventional variant scoring algorithms.

Discussion

Deep learning methods have already been broadly applied in various areas of genomics. One of the major advantages of deep learning is the at least partial removal of preprocessing steps, which are often error-prone and time-consuming. Neural networks as end-to-end models integrate these preprocessing steps in a single model often resulting in improved predictive power. Additionally, with the inclusion of feature extraction in these models, such as DNA motifs, Hi-C interactions and chromatin environment we could potentially learn novel biological relationships in genomic data. Another advantage of deep learning is its capability of finding relations in high dimensional and spatial dependent data. Nevertheless, deep learning models suffer from several limitations. For one, deep neural networks generally require larger amounts of data for training and while data is abundant in genomics, high quality gold standard labeled datasets are still few, providing difficulties for supervised learning approaches. Therefore, conventional models still have advantages when data is scarce. Another limitation of deep learning is the complexity in training, both for choosing network design as well as required computational power.

Areas of genomics suffers differently from these limitations. For instance, SNP and Indel calling with traditional machine learning algorithms already achieves high accuracy. Therefore, deep learning approaches in these areas, although promising, are far less impactful than deep learning has achieved in computer vision. Arguable, any improvement in accuracy, while not groundbreaking, is still worthwhile especially when used in diagnostic setting considering the number of patients it can be applied to. In contrast, improvements in SV calling is more achievable as traditional methods leave much to be desired. Similarly, non-coding variant pathogenic scoring approaches may result in comparable changes in its field. However, both methods are generally more difficult due limitations in available truth sets. Additionally, SV calling suffers from the available mapping quality which is often error prone. Another area in which deep learning applications could have major applications are data augmentation and artificial data creation. These areas are nearly exclusively achievable

using deep learning approaches and will both provide new training data as well as highlight underlying relations in genomics. As a summary, we provide an insight into areas we consider to be applicable for deep learning approaches (Figure 4). Subclasses of genomics are considered on innovativeness against traditional statistical and machine learning models, computational tractability, generalization on other platforms, data abundance and availability of truth sets.

In the future, we expect to find new applications of deep learning in many fields of genomics. Additionally, it will be important how current deep learning models are applied by the scientific community and if functional studies can take advantage of these models. For now, deep learning models implementation still requires a fair amount of knowledge on the subject and possible retraining for deployment. Therefore, researchers should strive towards the creation of deep learning models which are both highly generalizable and easy to use. Another limitation which should be addressed is the creation of large datasets for training of new models. Bundling information such as done with the Encode Project should help the progression in this area[87]. Moreover, generative models, such as the GAN networks, which could stimulate the amount of genomic data, without the need of experimental assays. Although for now the impact of deep learning has still to be seen, due to the complexity and magnitude of genomic information, we are certain that deep learning will be an important tool for uncovering all genomic elements.

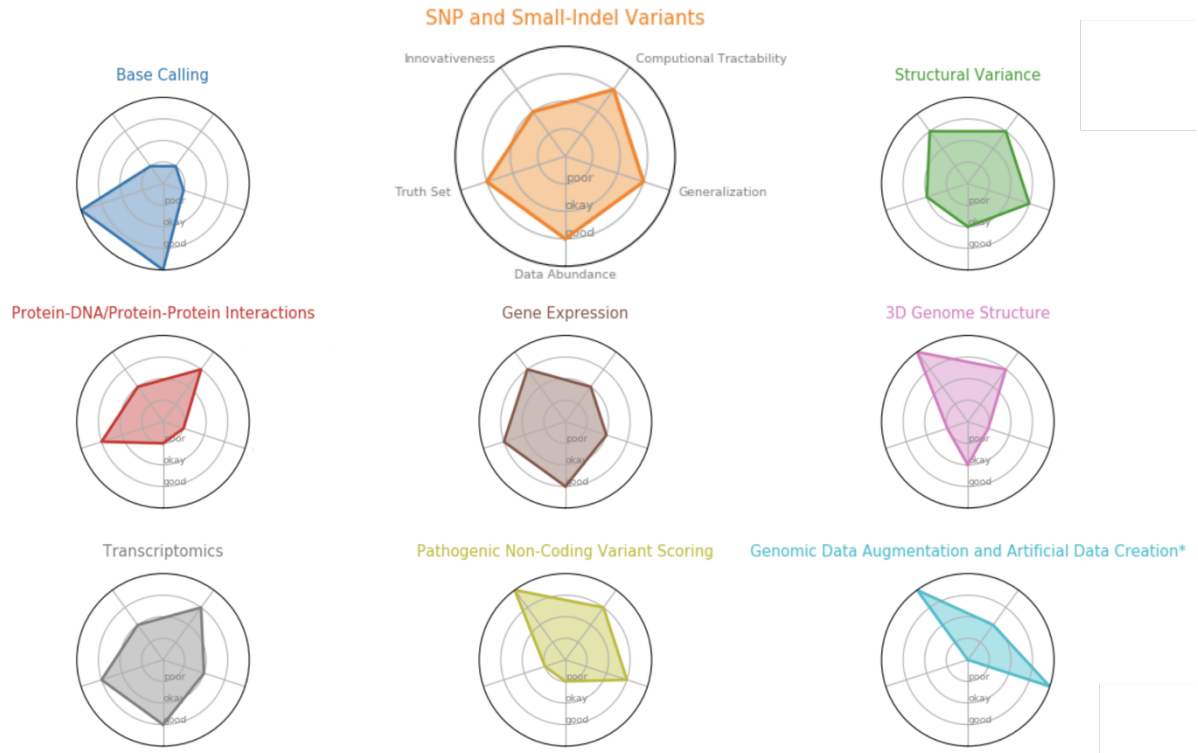

Figure 4: Considerations for deep learning approach applicability in different areas of genomics. *Categories are judged as non-existent, poor, okay, good, great. *Genomic data augmentation and artificial data creation is done using non-supervised learning approaches which do not require truth sets.*

Reference

1. Hieter, P. & Boguski, M. Functional genomics: It's all how you read it. *Science* **278**, 601–602 (1997).
2. Novelli, G., Ciccacci, C., Borgiani, P., Amati, M. P. & Abadie, E. Genetic tests and genomic biomarkers: Regulation, qualification and validation. *Clinical Cases in Mineral and Bone Metabolism* **5**, 149–154 (2008).
3. Chanock, S. J. *et al.* Replicating genotype-phenotype associations. *Nature* **447**, 655–660 (2007).
4. Kellis, M. *et al.* Defining functional DNA elements in the human genome. *Proceedings of the National Academy of Sciences of the United States of America* **111**, 6131–6138 (2014).
5. Qu, H. & Fang, X. A Brief Review on the Human Encyclopedia of DNA Elements (ENCODE) Project. *Genomics, Proteomics and Bioinformatics* **11**, 135–141 (2013).
6. Park, P. J. ChIP-seq: Advantages and challenges of a maturing technology. *Nature Reviews Genetics* **10**, 669–680 (2009).
7. Buenrostro, J. D., Wu, B., Chang, H. Y. & Greenleaf, W. J. ATAC-seq: A Method for Assaying Chromatin Accessibility Genome-Wide. *Curr. Protoc. Mol. Biol.* **109**, 21.29.1–9 (2015).
8. Belton, J.-M. *et al.* Hi–C: A comprehensive technique to capture the conformation of genomes. *Methods* **58**, 268–276 (2012).
9. Jian, X., Boerwinkle, E. & Liu, X. In silico tools for splicing defect prediction: A survey from the viewpoint of end users. *Genetics in Medicine* **16**, 497–503 (2014).
10. Heintzman, N. D. *et al.* Distinct and predictive chromatin signatures of transcriptional promoters and enhancers in the human genome. *Nat. Genet.* **39**, 311–318 (2007).
11. Down, T. A. & Hubbard, T. J. P. Computational detection and location of transcription start sites in mammalian genomic DNA. *Genome Res.* **12**, 458–461 (2002).
12. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning*. (MIT Press, 2016).
13. Deng, L. & Yu, D. *Deep learning : methods and applications*.
14. Havaei, M. *et al.* Brain Tumor Segmentation with Deep Neural Networks. (2015). doi:10.1016/j.media.2016.05.004
15. Lévy, D. & Jain, A. Breast Mass Classification from Mammograms using Deep Convolutional Neural Networks. (2016).

16. Quang, D., Chen, Y. & Xie, X. DANN: a deep learning approach for annotating the pathogenicity of genetic variants. *Bioinformatics* **31**, 761–763 (2015).
17. Zhou, J. & Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat. Methods* **12**, 931–934 (2015).
18. Park, Y. & Kellis, M. Deep learning for regulatory genomics. *Nature Biotechnology* **33**, 825–826 (2015).
19. Jones, W., Alasoo, K., Fishman, D. & Parts, L. Computational biology: deep learning. *Emerg. Top. Life Sci.* **1**, 257–274 (2017).
20. Rang, F. J., Kloosterman, W. P. & de Ridder, J. From squiggle to basepair: Computational approaches for improving nanopore sequencing read accuracy. *Genome Biology* **19**, (2018).
21. Poplin, R. *et al.* A universal snp and small-indel variant caller using deep neural networks. *Nature Biotechnology* **36**, 983 (2018).
22. Hill, S. T. *et al.* A deep recurrent neural network discovers complex biological rules to decipher RNA protein-coding potential. *Nucleic Acids Res.* **46**, 8105–8113 (2018).
23. Kelley, D. R., Snoek, J. & Rinn, J. L. Basset: Learning the regulatory code of the accessible genome with deep convolutional neural networks. *Genome Res.* **26**, 990–999 (2016).
24. Zeng, H. & Gifford, D. K. Predicting the impact of non-coding variants on DNA methylation. *Nucleic Acids Res.* **45**, e99–e99 (2017).
25. Zhou, J. *et al.* Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. *Nat. Genet.* **50**, 1171–1179 (2018).
26. Zou, J. *et al.* A primer on deep learning in genomics. *Nat. Genet.* **51**, 12–18 (2019).
27. Lecun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
28. Wainberg, M., Merico, D., Delong, A. & Frey, B. J. Deep learning in biomedicine. *Nat. Biotechnol.* **36**, 829–838 (2018).
29. Angermueller, C., Pärnamaa, T., Parts, L. & Stegle, O. Deep learning for computational biology. *Mol. Syst. Biol.* **12**, 878 (2016).
30. Bishop, C. M. *Pattern Recogniton and Machine Learning*. *Information Science and Statistics* (Springer-Verlag New York, 2006).
31. Liu, H. & Cocea, M. Semi-random partitioning of data into training and test sets in granular computing context. *Granul. Comput.* **2**, 357–386 (2017).

32. Bengio, Y. Practical Recommendations for Gradient-Based Training of Deep Architectures. in 437–478 (2012). doi:10.1007/978-3-642-35289-8_26
33. Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A. & Talwalkar, A. Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. (2016).
34. Karlik, B. Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks.
35. Smith, L. N. Cyclical learning rates for training neural networks. in *Proceedings - 2017 IEEE Winter Conference on Applications of Computer Vision, WACV 2017* 464–472 (Institute of Electrical and Electronics Engineers Inc., 2017). doi:10.1109/WACV.2017.58
36. McKenna, A. *et al.* The genome analysis toolkit: A MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* **20**, 1297–1303 (2010).
37. Kim, S. *et al.* Strelka2: fast and accurate calling of germline and somatic variants. *Nat. Methods* **15**, 591–594 (2018).
38. Depristo, M. A. *et al.* A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.* **43**, 491–501 (2011).
39. Zook, J. M. *et al.* Integrating human sequence data sets provides a resource of benchmark SNP and indel genotype calls. *Nat. Biotechnol.* **32**, 246–251 (2014).
40. Xu, H., DiCarlo, J., Satya, R. V., Peng, Q. & Wang, Y. Comparison of somatic mutation calling methods in amplicon and whole exome sequence data. *BMC Genomics* **15**, (2014).
41. Ding, J. *et al.* Feature-based classifiers for somatic mutation detection in tumour–normal paired sequencing data. *Bioinformatics* **28**, 167–175 (2012).
42. Poplin, R. *et al.* A universal snp and small-indel variant caller using deep neural networks. *Nature Biotechnology* **36**, 983 (2018).
43. Luo, R., Sedlazeck, F. J., Lam, T. W. & Schatz, M. C. A multi-task convolutional deep neural network for variant calling in single molecule sequencing. *Nat. Commun.* **10**, (2019).
44. Luo, R. *et al.* Clair: Exploring the limit of using a deep neural network on pileup data for germline variant calling. *bioRxiv* 865782 (2019). doi:10.1101/865782
45. Zhou, P. *et al. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification*.

46. Sahraeian, S. M. E. *et al.* Deep convolutional neural networks for accurate somatic mutation detection. *Nat. Commun.* **10**, (2019).
47. He, K., Zhang, X., Ren, S. & Sun, J. Identity Mappings in Deep Residual Networks. (2016).
48. Sahraeian, S. M. E., Fang, L. T., Mohiyuddin, M., Hong, H. & Xiao, W. Robust Cancer Mutation Detection with Deep Learning Models Derived from Tumor-Normal Sequencing Data. *bioRxiv* 667261 (2019). doi:10.1101/667261
49. Ainscough, B. J. *et al.* A deep learning approach to automate refinement of somatic variant calling from cancer sequencing data. *Nat. Genet.* **50**, 1735–1743 (2018).
50. Ho, S. S., Urban, A. E. & Mills, R. E. Structural variation in the sequencing era. *Nat. Rev. Genet.* (2019). doi:10.1038/s41576-019-0180-9
51. Kosugi, S. *et al.* Comprehensive evaluation of structural variation detection algorithms for whole genome sequencing. *Genome Biol.* **20**, 117 (2019).
52. Zook, J. M. *et al.* A robust benchmark for germline structural variant detection. *bioRxiv* 664623 (2019). doi:10.1101/664623
53. Kuzniar, A. *et al.* sv-callers: a highly portable parallel workflow for structural variant detection in whole-genome sequence data. doi:10.7717/peerj.8214
54. Cai, L., Wu, Y. & Gao, J. DeepSV: accurate calling of genomic deletions from high-throughput sequencing data using deep convolutional neural network. *BMC Bioinformatics* **20**, 665 (2019).
55. Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.* **33**, 831–838 (2015).
56. Zeng, H., Edwards, M. D., Liu, G. & Gifford, D. K. Convolutional neural network architectures for predicting DNA–protein binding. *Bioinformatics* **32**, i121–i127 (2016).
57. Lanchantin, J., Singh, R., Lin, Z. & Qi, Y. Deep Motif: Visualizing Genomic Sequence Classifications. (2016).
58. Budach, S. & Marsico, A. pysster: classification of biological sequences by learning sequence and structure motifs with convolutional neural networks. *Bioinformatics* **34**, 3035–3037 (2018).
59. Min, X. *et al.* Predicting enhancers with deep convolutional neural networks. *BMC*

Bioinformatics **18**, (2017).

60. Wang, M., Tai, C., E, W. & Wei, L. DeFine: deep convolutional neural networks accurately quantify intensities of transcription factor-DNA binding and facilitate evaluation of functional non-coding variants. *Nucleic Acids Res.* **46**, e69–e69 (2018).
61. Qin, Q. & Feng, J. Imputation for transcription factor binding predictions based on deep learning. *PLOS Comput. Biol.* **13**, e1005403 (2017).
62. Quang, D. & Xie, X. EXTREME: an online EM algorithm for motif discovery. *Bioinformatics* **30**, 1667–1673 (2014).
63. Quang, D. X., Erdos, M. R., Parker, S. C. J. & Collins, F. S. Motif signatures in stretch enhancers are enriched for disease-associated genetic variants. *Epigenetics and Chromatin* **8**, (2015).
64. Quang, D. & Xie, X. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. *Nucleic Acids Res.* **44**, e107–e107 (2016).
65. Hassanzadeh, H. R. & Wang, M. D. *DeeperBind: Enhancing Prediction of Sequence Specificities of DNA Binding Proteins*.
66. Singh, S., Yang, Y., Póczos, B. & Ma, J. Predicting enhancer-promoter interaction from genomic sequence with deep neural networks. *Quant. Biol.* **7**, 122–137 (2019).
67. Kelley, D. R. *et al.* Sequential regulatory activity prediction across chromosomes with convolutional neural networks. *Genome Res.* **28**, 739–750 (2018).
68. Quang, D. & Xie, X. FactorNet: A deep learning framework for predicting cell type specific transcription factor binding from nucleotide-resolution sequential data. *Methods* **166**, 40–47 (2019).
69. Avsec, Ž. *et al.* Deep learning at base-resolution reveals motif syntax of the cis-regulatory code. doi:10.1101/737981
70. Singh, R., Lanchantin, J., Robins, G. & Qi, Y. DeepChrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics* **32**, i639–i648 (2016).
71. Liu, Q. *et al.* Detection of DNA base modifications by deep recurrent neural network on Oxford Nanopore sequencing data. *Nat. Commun.* **10**, (2019).
72. Wang, Y. *et al.* Predicting DNA Methylation State of CpG Dinucleotide Using Genome Topological Features and Deep Networks. *Sci. Rep.* **6**, (2016).
73. Angermueller, C., Lee, H. J., Reik, W. & Stegle, O. DeepCpG: accurate prediction of

single-cell DNA methylation states using deep learning. *Genome Biol.* **18**, 67 (2017).

74. Schreiber, J., Libbrecht, M., Bilmes, J. & Noble, W. S. Nucleotide sequence and DNaseI sensitivity are predictive of 3D chromatin architecture. *bioRxiv* 14 (2017). doi:10.1101/103614
75. Zhang, Y. *et al.* Enhancing Hi-C data resolution with deep convolutional neural network HiCPlus. *Nat. Commun.* **9**, (2018).
76. Fu, W. *et al.* Analysis of 6,515 exomes reveals the recent origin of most human protein-coding variants. *Nature* **493**, 216–220 (2013).
77. Kumar, P., Henikoff, S. & Ng, P. C. Predicting the effects of coding non-synonymous variants on protein function using the SIFT algorithm. *Nat. Protoc.* **4**, 1073–1082 (2009).
78. De Baets, G. *et al.* SNPeffect 4.0: on-line prediction of molecular and structural effects of protein-coding variants. *Nucleic Acids Res.* **40**, D935–D939 (2012).
79. Rentzsch, P., Witten, D., Cooper, G. M., Shendure, J. & Kircher, M. CADD: predicting the deleteriousness of variants throughout the human genome. *Nucleic Acids Res.* **47**, D886–D894 (2019).
80. Lee, D. *et al.* A method to predict the impact of regulatory variants from DNA sequence. *Nat. Genet.* **47**, 955–961 (2015).
81. Rogers, M. F. *et al.* FATHMM-XF: accurate prediction of pathogenic point mutations via extended features. *Bioinformatics* **34**, 511–513 (2018).
82. Ionita-Laza, I., Mccallum, K., Xu, B. & Buxbaum, J. D. A spectral approach integrating functional genomic annotations for coding and noncoding variants. *Nat. Genet.* **48**, 214–220 (2016).
83. McLaren, W. *et al.* The Ensembl Variant Effect Predictor. *Genome Biol.* **17**, 122 (2016).
84. Nishizaki, S. S. & Boyle, A. P. Mining the Unknown: Assigning Function to Noncoding Single Nucleotide Polymorphisms. *Trends in Genetics* **33**, 34–45 (2017).
85. Dong, C. *et al.* Comparison and integration of deleteriousness prediction methods for nonsynonymous SNVs in whole exome sequencing studies. *Hum. Mol. Genet.* **24**, 2125–2137 (2015).
86. Zhou, J. *et al.* Whole-genome deep-learning analysis identifies contribution of noncoding mutations to autism risk. *Nat. Genet.* **51**, 973–980 (2019).
87. ENCODE Project Consortium. An integrated encyclopedia of DNA elements in the

human genome. *Nature* **489**, 57–74 (2012).

88. Teng, H. *et al.* Chiron: translating nanopore raw signal directly into nucleotide sequence using deep learning. *Gigascience* **7**, (2018).
89. Miculinić, N., Ratković, M. & Šikić, M. MinCall - MinION end2end convolutional deep learning basecaller. (2019).
90. Goodfellow, I. J. *et al. Generative Adversarial Nets*.
91. Zook, J. M. *et al.* Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Sci. Data* **3**, (2016).