MuStARD: Deep Learning for intra- and inter-species scanning of functional genomic patterns

Author list

Georgios K Georgakilas[1], Andrea Grioni[1,2,3], Konstantinos G Liakos[4], Eliska Malanikova[1,2], Fotis C Plessas[4] and Panagiotis Alexiou[1,*].

1. Central European Institute of Technology, Brno, Czech Republic
2. Faculty of Science, National Centre for Biomolecular Research, Masaryk University, Brno, Czech Republic
3. Centro Ricerca Tettamanti, Pediatric Department, University of Milano-Bicocca, Fondazione MBBM, Monza, Italy
4. Department of Electrical and Computer Engineering, School of Engineering, University of Thessaly, Volos, Greece

* Correspondence should be addressed to Panagiotis Alexiou.

Abstract

Regions of the genome that produce different classes of functional elements also exhibit different patterns in their sequence, secondary structure, and evolutionary conservation. Deep Learning is a family of Machine Learning algorithms recently applied to a variety of pattern recognition problems. Here we present MuStARD (gitlab.com/RBP_Bioinformatics/mustard) a Deep Learning framework that can learn and combine sequence, structure, and conservation patterns in sets of functional regions, and accurately identify additional members of the given set over wide genomic areas. MuStARD is designed with general use in mind, and has sophisticated iterative fully-automated background selection capability. We demonstrate that MuStARD can be trained without changes on different classes of human small RNA loci (pre-microRNAs and snoRNAs) and accurately build prediction models for both, outperforming state of the art methods specifically designed for each specific class. Furthermore, we demonstrate the ability of MuStARD for inter-species identification of functional elements by predicting mouse small RNAs using human trained models. MuStARD is easy to deploy and extend to a variety of genomic classification questions.

Introduction

The sequencing of the complete human genome first brought to light the realization that the majority of the genetic material in human cells does not code for protein coding genes[1]. Genome-wide in silico analyses of conserved mammalian regulatory sequences initially concentrated on the untranslated regions of mRNAs, promoter and enhancer elements[2]. In following years, both the variety of coded molecules, and the number of sequenced genomes have been increasing with fast pace—newly discovered molecule families such as microRNA (miRNA), Piwi-interacting RNA (piRNA), Short hairpin RNA (shRNA), Small interfering RNA (siRNA), Small nuclear RNA (snRNA), Small nucleolar RNA (snoRNA), Long non-coding RNAs (lncRNA) and others now populate the functional expression map of known genomes. The number of organisms with sequenced genomes has been increasing exponentially for the past decade, with NCBI currently listing just over 7000 eukaryotic sequenced genomes, of which almost 50 have fully assembled genomes, and approximately 1000 have assembled chromosomes. The majority of these newly sequenced genomes cannot be experimentally annotated to the depth of well used genomes such as human, mouse, or drosophila. Several computational methods attempt to accurately predict the location of non-coding RNA genomic positions. For example, tens of programs aiming at pre-microRNA identification have been developed, but none achieving accurate genome-wide prediction[3]. In silico methods utilizing sequence homology are often employed for the annotation of novel genomes, projecting functional regions of well annotated species to homologous genomic regions of less annotated genomes. Alternatively, whole genomes can be 'scanned' for regions of known characteristics, such as a specific motif, or sequence, and their putative function annotated.

Here, we present a machine learning method that improves the accuracy of non-coding RNA prediction in known species, and demonstrate that the models trained on a well annotated species can be used to scan large genomic regions and identify cross-species functional elements of the same class. We have chosen to apply our method on two different classes of small RNAs: precursor miRNAs (pre-miRNAs) and Small nucleolar RNAs (snoRNAs). Precursor miRNAs are intermediate RNA molecules of miRNA biogenesis that form stable hairpin structures of approximately 60-100 nucleotides. The first novel miRNAs were identified by sequencing total RNA of their approximate length[4–6]. Based on the characteristics of the first sequenced miRNAs, computational methods were introduced to accelerate the identification process. Current computational methods utilize some combination of manually produced features based on genomic sequence and conservation, as well as predictions of RNA folding. These features could include the free energy of folding, folding stem length, nucleotide content in the stem, occurrence of matching pairs and so on. In a recent thorough comparison of several highly cited programs, it was observed that no tool significantly outperforms all other tools on all tested data sets[3]. Additionally, none of the current tools can employ a 'scanning' mode for large genomic regions leading to accurate novel pre-miRNA loci identification. Currently, the latest miRBase release[7], the main repository of known miRNA sequences gives access to 38,589 pre-miRNAs from 271 organisms with 1,917 being of human origin.

The highly competitive field of pre-miRNA prediction can be juxtaposed with the relative scarcity of snoRNA prediction algorithms. Discovered shortly after the sequencing of the human genome[8] snoRNAs play an important role in the processing and modification of other classes of RNAs. Over ten years ago, the human genome was scanned for snoRNAs[9], identifying approximately 300 snoRNA loci. Hundreds more snoRNAs were identified by small RNA sequencing of diverse species and filtering through a computational algorithm[10]. The field of in silico snoRNA prediction appears too small to warrant the attention of large initiatives to implement complex machine learning architectures and manually curated features. Here, we will demonstrate that our method can accurately predict snoRNA locations, proving that it will be a useful tool for the generalized identification and annotation of less studied classes of functional elements.

Machine Learning (ML) describes the field of computer science that involves development of mathematical models and their implementations with the purpose of enabling computers to learn concepts and patterns embedded in data. Artificial neural networks are a collection of ML algorithms with a rich and at the same time interesting history. Neural Networks (NNs) were first proposed decades ago[11] as a simplified model describing the process of biological neurons in the brain. NNs approximate the process of learning in the brain by stacking interconnecting layers of artificial neurons or ‘nodes’. Nodes in early layers converge into recognizing simplified and more primitive patterns in the input data while propagating their computations deeper into the network. Neurons in subsequent layers receive and build on top of these patterns evolving into detectors of more abstract and complex concepts. Deep Learning (DL) is a term that refers to recent breakthroughs in the field of NNs including a collection of new methodologies that outperform well-established ML algorithms in multiple fields. Deep Neural Networks offer significant flexibility and remarkable accuracy provided enough data, especially for complex learning tasks. The majority of supervised ML algorithms require pre-processing of the input data set in the form of feature extraction especially in the case of biological problems that involve raw DNA or RNA sequences. This process involves an arbitrary number of features that have been conceptualized on ad hoc bases, usually derived on empirical data that are interpreted based on personal experiences and assumptions. This ad hoc process of feature extraction frequently introduces biases that might severely affect building robust models—while at the same time does not offer the possibility to utilize and also unveil all underlying patterns. DL models have a remarkable ability of not relying on arbitrary feature extraction procedures by incorporating a process known as convolution in the basis of the network architecture. Convolutional Neural Networks (CNNs) are able to operate directly on raw data such as images, time-series, DNA/RNA sequences and many more without the need of pre-processing and feature extraction. CNNs use convolutional layers to process the input prior to propagating the signal to the dense part of the network and in the process they act as extractors of hidden patterns themselves.

These properties have revolutionized speech recognition and image classification[12] in the past 6 years. The success of DL was almost immediately picked up from researchers in other fields such as physics[13] and chemistry[14]. Medical informatics and computational biology could not be an exception to the rule of DL slowly winning its place as a popular algorithmic framework in almost every scientific field. During the last 4 years there has been an explosion of DL applications providing novel or improving existing methodologies in

Medicine and Biology[15]. There are already dozens of published studies that applied a plethora of DL architectures in Biology. For example, epigenomic data were used to infer gene expression[16], and ovarian cancer subtypes were defined from gene and microRNA expression as well as DNA methylation[17]. DeepBind[18] was the first application of CNNs in transcription factor binding recognition tasks. DeepSEA[19] and DanQ[20] are also CNN-based frameworks that were trained on a large multi-cell-type compendium of chromatin-profiling data, including DNase I sensitivity, TF and histone-mark ChIP-seq data. Basset[21] and DeepEnhancer[22] both used CNN-based architectures on chromatin accessibility data to predict enhancers.

Results

Overview of our method

Here we introduce MuStARD (Machine-learning System for Automated RNA Discovery), a highly flexible Deep Learning framework that can be applied to any biological problem that involves deconvolution of patterns embedded in DNA/RNA sequences. The framework's flexibility stems from its modular design and minimum input requirements (Figure 1a). The majority of existing algorithms that perform classification tasks in various fields of biological research, pre-miRNA detection for example, rely on extraction of arbitrary features from raw input data. This process often requires significant expertise on the relevant field, it can cause increased computational overhead and most importantly it frequently introduces biases that can severely affect the training of robust models. MuStARD is a feature-agnostic DL framework that utilizes convolutional layers to scan the input data avoiding manual feature extraction. MuStARD input can currently be structured as any of the following three types or any combination of those: raw DNA sequence, RNAfold[23] derived secondary structure and PhyloP[24] basewise evolutionary conservation score. Another novel aspect of our framework is the automated iterative identification of background sequences that present similar characteristics with the positive sequences of the given set regions that would be otherwise impossible to detect by randomly selecting regions from all over the genome. This process is able to provide an enhanced version of background sequences specifically tuned for the classification task at hand. We showcase MuStARD's flexibility and robustness in providing accurate and high-resolution predictions on the tasks of scanning the genome and detecting pre-miRNAs and snoRNAs, two distinct classes of small RNAs that exhibit diverse patterns in terms of size, evolutionary conservation and secondary structure.

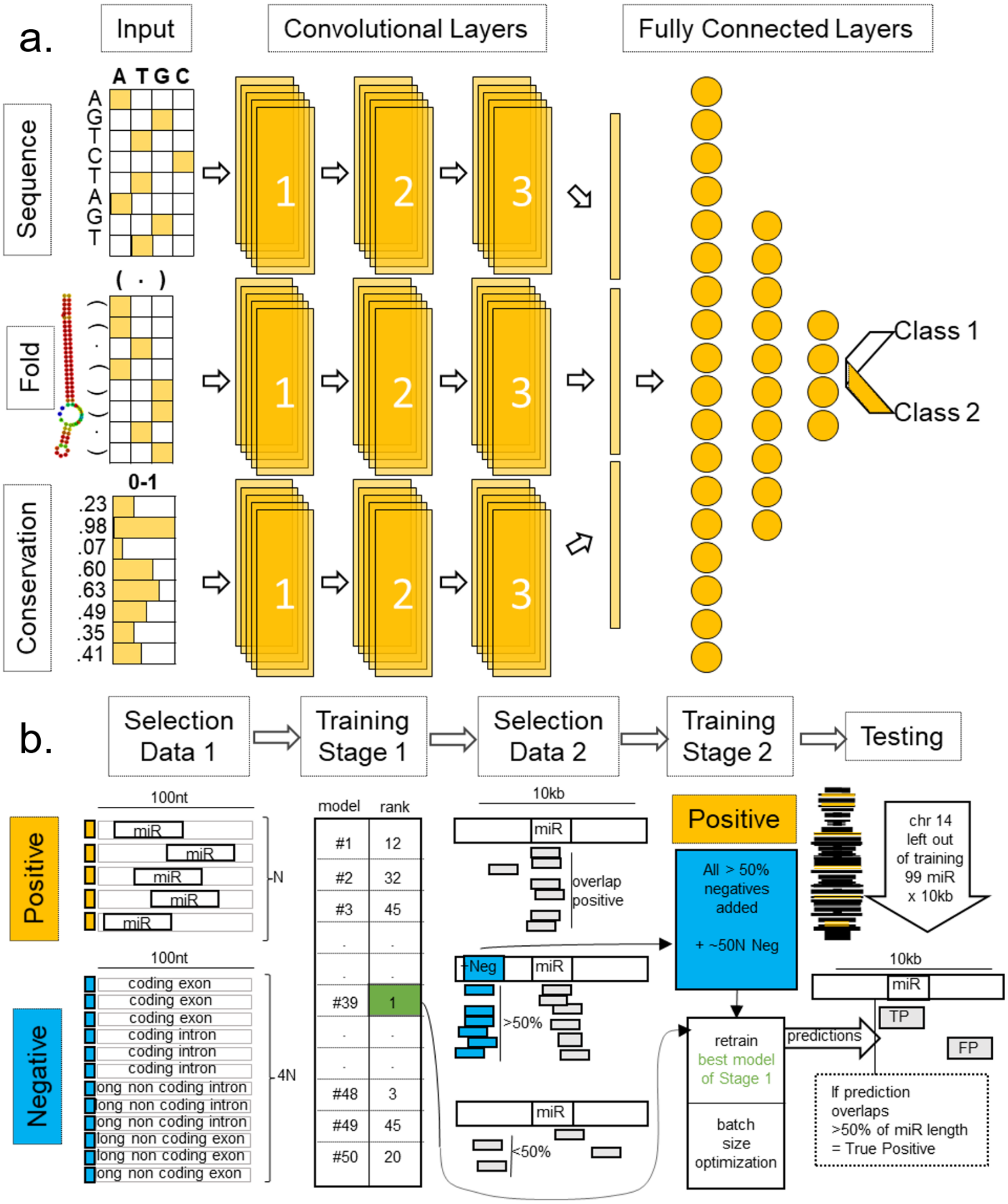

Figure 1. Overview of MuStARD modular architecture and iterative training pipeline. a) MuStARD is able to handle any combination of either raw DNA sequences, RNAfold derived secondary structure and basewise evolutionary conservation from PhyloP. DNA sequences and RNAfold output are one-hot encoded while PhyloP score is not pre-processed. Each feature category is forwarded to a separate 'branch' that consists of three convolutional layers. The computations from all branches are concatenated prior to being forwarded to the fully connected part of the network. b) The training pipeline of MuStARD consists of two steps. Initially, pre-miRNA sequences are randomly shuffled to exonic and intronic (protein-coding and lincRNA genes) regions of the genome to extract equal sized negative sequences with 1:4 positive to negative ratio. This process is repeated 50 times to facilitate the training

of equal number of models. The performance of each model is assessed based on the test set and all false positives that are supported by at least 25 models are extracted. This set of false positives is added to the negative pool of the best performing model to create an enhanced training set. The enhanced set is finally used to train the final MuStARD model.

Evaluation of Input Data Combinations on pre-miRNA prediction

We evaluated the performance of MuStARD on all combinations of input data for the pre-miRNA prediction dataset. Six combinations of input were tested, namely: all three inputs combined (MuStARD-mirSFC), sequence and conservation (MuStARD-mirSC), sequence and secondary structure (MuStARD-mirSF), secondary structure and conservation (MuStARD-mirFC), just sequence (MuStARD-mirS) and just secondary structure (MuStARD-mirF). An additional model was trained for the combination of all three inputs but with the Keras class weights option disabled (MuStARD-mirSFC-U). Each of these models underwent independent hyperparameter optimization for optimal batch size (Supplementary Table 1).

Scanning test sequences with these 7 different models reinforced our expectation that models including a higher number of meaningful input data branches would perform better in retrieval of pre-miRNAs (Figure 2). The model trained on secondary structure and conservation, was the best performing two input model. This result aligns with the identification of pre-miRNA hairpins by the Microprocessor complex during the biogenesis of miRNAs primarily by characteristics of their secondary structure rather than sequence[25]. Surprisingly, the three input model trained without class weights (MuStARD-mirSFC-U) slightly outperforms the weighted model (MuStARD-mirSFC) in this evaluation. Since MuStARD-mirSFC and -mirSFC-U perform better than two or one input models in all evaluations, we will only report results for these two models.

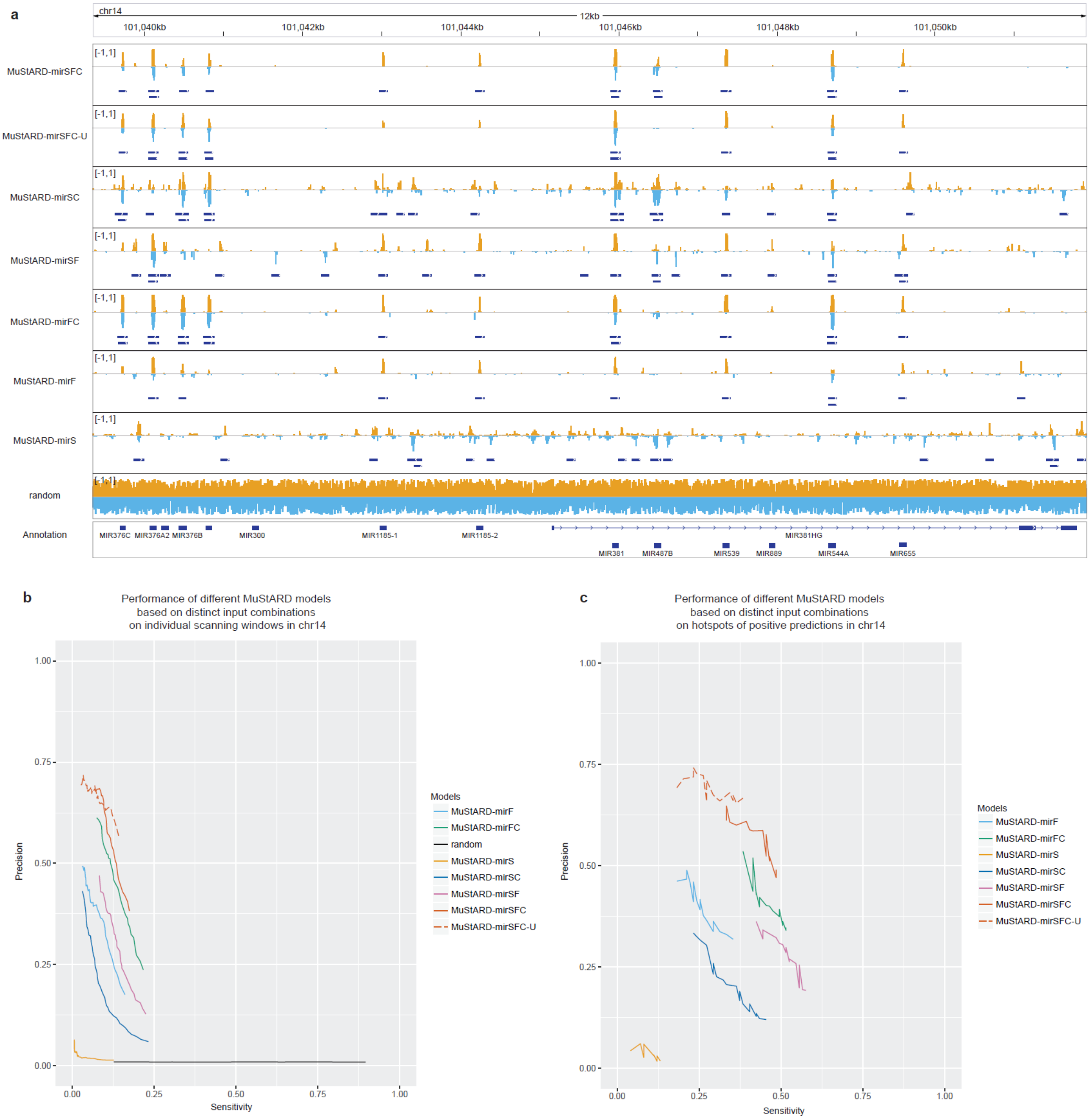

Figure 2. Performance of MuStARD models trained on multiple combination of input data. a) Visualization of each model's performance on the scanning windows surrounding pre-miRNAs of MIR381HG locus in chromosome 14. The orange colored tracks represent prediction scores on the forward strand while light blue corresponds to the reverse strand. The dark blue boxes underneath each score track represent hotspots of overlapping positively predicted windows for each model. A track with randomly assigned scores for every window has also been added serving as the baseline. MuStARD output score range is [0,1]. However, for visualization purposes, the score of windows in the reverse strand were multiplied by -1. As described in the methods, the hotspots of positive predictions were assembled after filtering out windows with score less than 0.5. b,c) Performance, based on precision and sensitivity, of MuStARD models trained on different input combinations such as sequence with secondary structure and conservation (MuStARD-mirSFC), sequence with conservation (MuStARD-mirSC), sequence with secondary structure (MuStARD-mirSF), secondary structure and conservation (MuStARD-mirFC), secondary structure (MuStARD-mirF) and

sequence (MuStARD-mirS) only. b) Performance is assessed by taking into account all scanning windows as individual predictions. The 'random' model has been created by randomly assigning a score in the range of [0,1] to each window and serves as the baseline. c) Performance is measured by creating hotspots of positive predictions by merging overlapping windows that exhibit a score greater than 0.5.

Evaluation of pre-miRNA prediction algorithms on chromosome 14 scanning

While training MuStARD models, we kept the entirety of chromosome 14 aside as a final benchmarking set that could be fairly used to evaluate MuStARD's performance against the current state of the art in pre-miRNA prediction algorithms. There are currently over 30 published pre-miRNA prediction algorithms indexed in OMICtools[26] repository. The majority of these studies could not be coerced to run on our benchmarking dataset (See Methods for details). We managed to run and evaluate five state-of-the-art programs: HuntMi[27], microPred[28], miPred[29], miRBoost[30] and triplet-SVM[31]. Of these five, only triplet-SVM, miPred and miRBoost provide output scores in the form of probabilities allowing assessment of their performance on multiple score thresholds. HuntMi and microPred provide fixed output labels (yes/no) limiting their performance comparison on a fixed threshold (Supplementary Table 2). When compared to state-of-the-art algorithms (Figure 3a), the two MuStARD models show increase in precision and sensitivity along all thresholds (Figures 3b and 3c) while at the same time providing sharp predictions of shorter length than other algorithms (Figure 3d) centered closest to the real pre-miRNAs (Figure 3e).

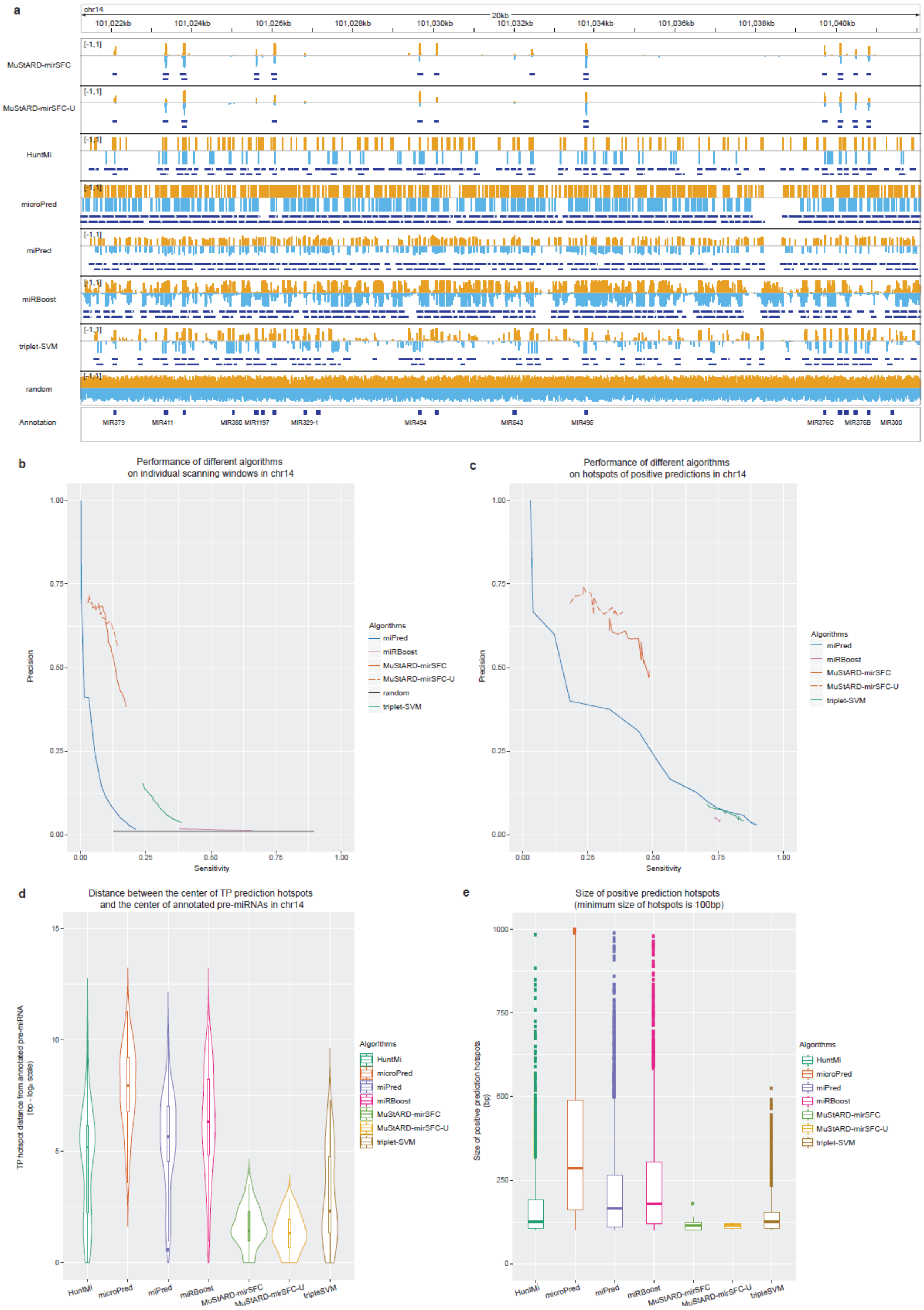

Figure 3. Comparison between MuStARD and existing pre-miRNA detection algorithms on scanning chromosome 14. a) Genome browser visualization of each algorithm's performance on the scanning windows in a 20kb locus hosting several pre-miRNAs. b,c) Performance of

different algorithms based on assessing scanning windows individually (b) and on hotspots of positive predictions (c). HuntMi and microPred algorithms were excluded from these comparisons since they provide class labels as output rather than continuous values that can be subjected to different thresholds. d) Distributions of distance between true positive hotspots of positive predictions and the overlapping annotated pre-miRNA. e) Size distributions of hotspots of positive predictions.

Evaluation of pre-miRNA prediction algorithms on labelled data

The process of genome-wide scanning for pre-miRNAs requires windows of fixed size, a property that perfectly fits the input requirements of DL algorithms. In fact, the enhanced dataset consists of positive and negative sequences of variable length. These sequences are extended to 100bp prior to MuStARD processing (see Methods for details). However, the majority of existing algorithms instead perform feature extraction and normalization to account for differences on sequence sizes.

Using a benchmark dataset of fixed sized sequences should not introduce any biases to comparing the performance of MuStARD and existing algorithms. Nevertheless, we performed an additional comparison based on benchmark sequences (chromosome 14) of the enhanced set without reinforcement. Only for MuStARD, but not for existing algorithms, we applied the extension procedure of these sequences to 100bp (Supplementary Table 3).

Both MuStARD models significantly outperform every algorithm in terms of precision. MuStARD-mirSFC-U in particular exhibits unprecedented levels of precision even for score thresholds as low as 0.1 (Figure 4). MiPred rises above MuStARD for a score threshold of 0.84, however, at that threshold it only manages to provide 5 True Positives (TP) for 0 False Positives (FP) while MuStARD-mirSFC-U provides 15 TPs and 1 FP and MuStARD-mirSFC 22 TPs and 2 FPs.

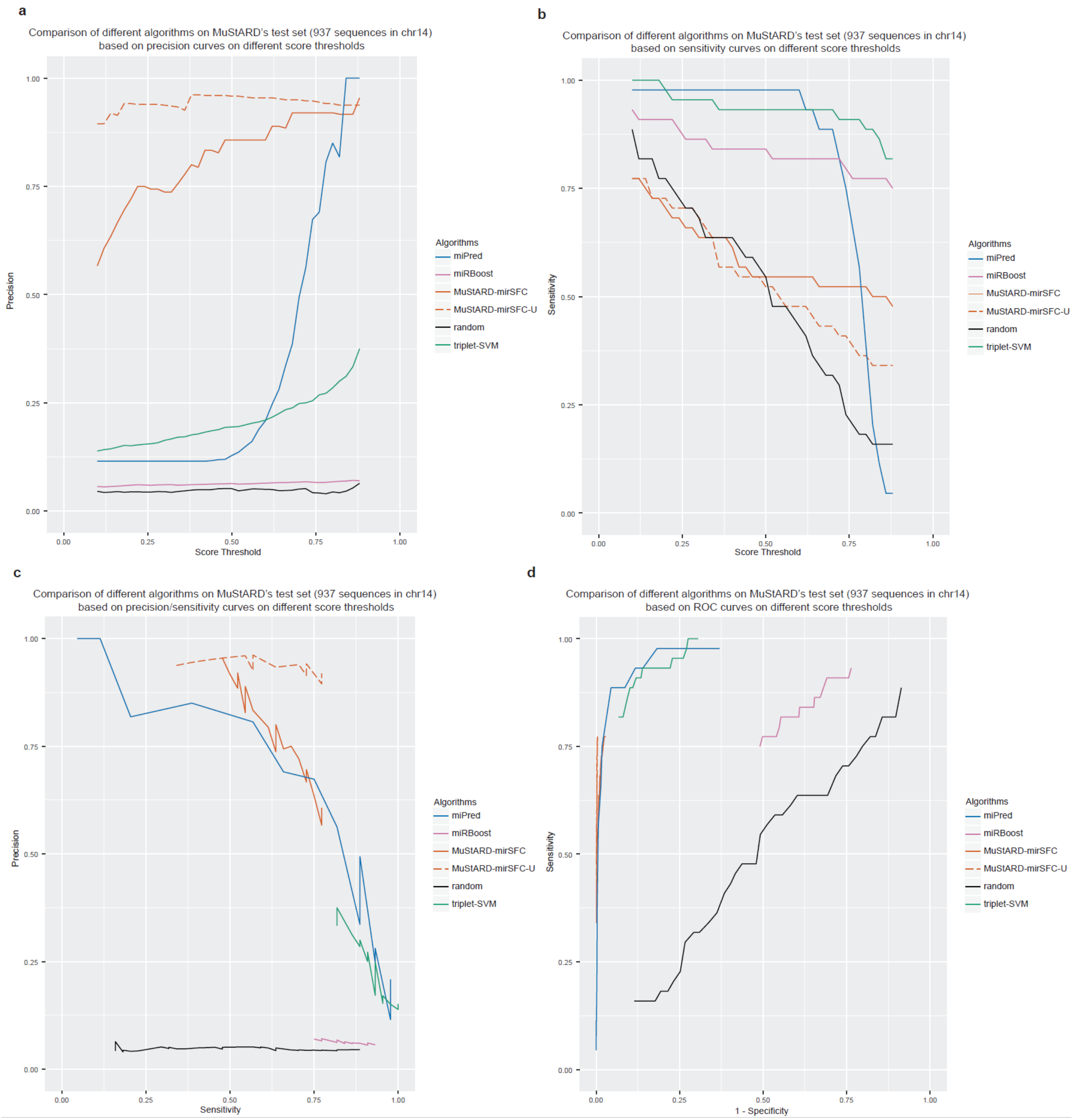

Figure 4. Comparison between MuStARD and existing pre-miRNA detection algorithms on the enhanced test set of chromosome 14. a) Precision assessment on multiple score thresholds of different algorithms. b) Performance of algorithms in terms of sensitivity on different score threshold. c) Combination of precision and sensitivity metrics on multiple score thresholds in a single plot. d) ROC curves of different algorithms on different score thresholds.

Cross-species prediction

We attempted the cross-species application of MuStARD-mirSFC and -mirSFC-U models trained on human pre-miRNAs to areas of the mouse genome. We scanned a 10kb wide window centered around a mouse miRNA precursor (N=1,227) resulting in approximately 3.9M sliding windows that were individually assessed and scored. Figure 5a depicts the visualization of the scanning results over 20kb of the *Mirg* locus in chromosome 12. Both models maintain the same properties observed in human results (Figures 2a and 3a) by providing precise and high-resolution predictions in mouse even though there are differences in the evolutionary conservation profile annotation provided for the two species. The performance of both MuStARD models was also assessed in terms of precision and sensitivity both by using scanning windows individually and using hotspots of positive predictions in the comparison between human and mouse (Figures 5b and 5c). As expected, both models exhibit lower performance in mouse but they maintain exceptional levels of generalisation capacity (Supplementary Table 4). This is also depicted in Figure 5d. From the total number of 1,227 mouse pre-miRNAs, MuStARD-mirSFC-U manages to correctly predict 306 (hotspot score threshold of 0.5) including 94 of the 129 known human orthologues.

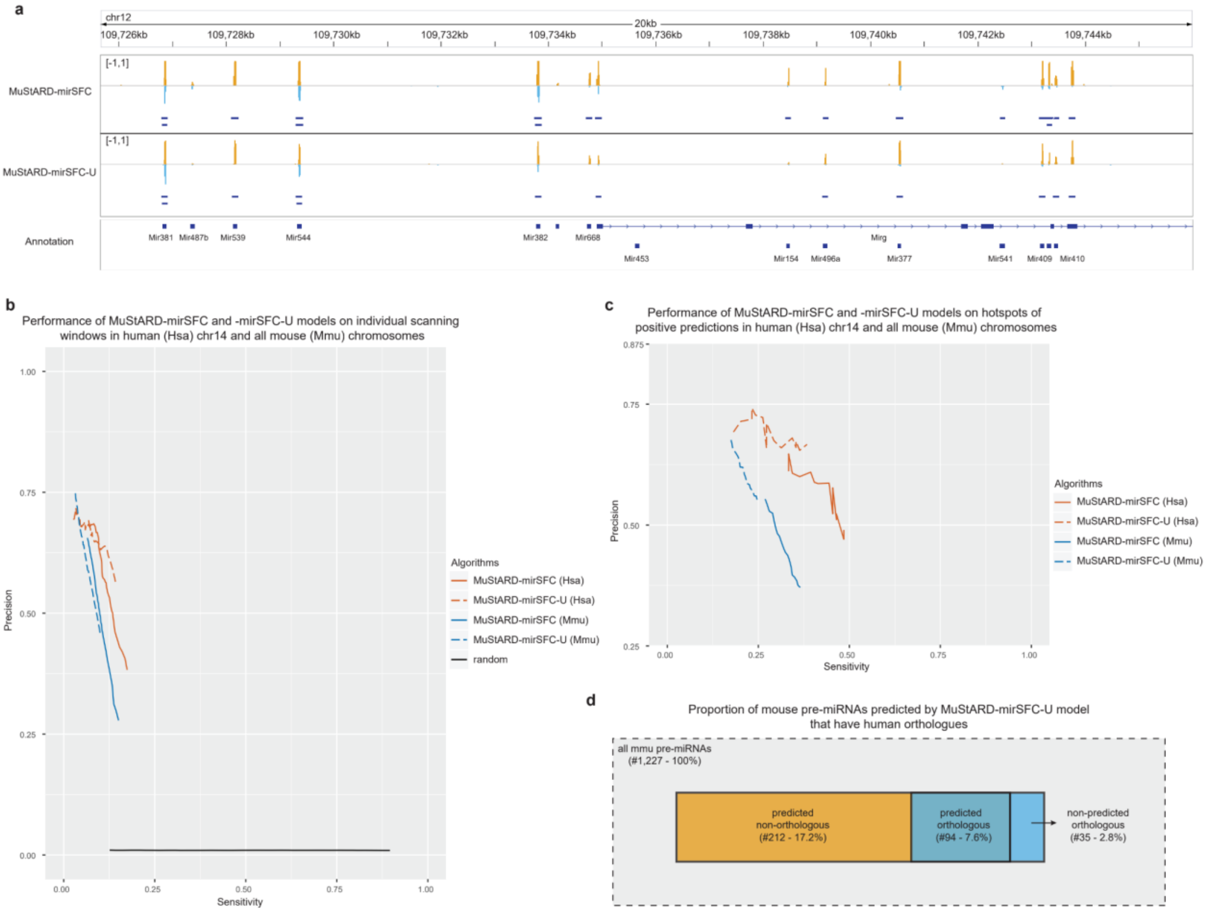

Figure 5. Performance of human MuStARD-mirSFC and -mirSFC-U models on the mouse genome. a) Genome browser visualization of both models on the scanning windows in Mirg pre-miRNA cluster. b) Performance in terms of precision and sensitivity based on assessing scanning windows individually c) Performance in terms of precision and sensitivity on hotspots of positive predictions. The performance on human chromosome 14 (also shown in Figure 3) is depicted with orange color (random with black color) while the performance on all mouse chromosomes is shown in light blue. d) Proportion of mouse pre-miRNAs predicted by MuStARD-mirSFC-U that also have human orthologues.

Training MuStARD to detect snoRNAs

In the previous sections, we have demonstrated several properties of MuStARD that enable breakthrough performances in the field of pre-miRNA detection intra- as well as inter-species. Despite its performance, MuStARD was not specifically developed for pre-miRNA hairpin detection. Our intention is to provide a highly flexible computational framework that can be applied on the identification of a variety of biological patterns. To highlight the flexibility of MuStARD, we applied the same training pipeline used on pre-miRNA identification to snoRNA sequences in human (Figure 6a). We trained two distinct models on the same dataset, MuStARD-snoSFC (Keras class weights enabled) and MuStARD-snoSFC-U (Keras class weights disabled), using raw DNA sequence, secondary structure and conservation as input (Supplementary Table 5). We also applied these models to the mouse genome to verify their generalisation capacity (Figure 6b). For every mouse snoRNA (N=1,507), we used both models to scan a 10kb window centered on the snoRNA. This resulted in 4.8M sliding windows that were individually assessed and scored. Performance metrics were calculated based on taking into account each sliding window as an individual prediction as well as based on hotspots of positive predictions (Figures 6c and 6d). We did not compare our findings to state of the art algorithms for snoRNA detection as we are not aware of the existence of such algorithms. MuStARD achieved comparable levels of precision and resolution of predicted snoRNAs and miRNAs, and at the same time captured the characteristics of the problem and transferred that knowledge on another organism (Supplementary Tables 6 and 7).

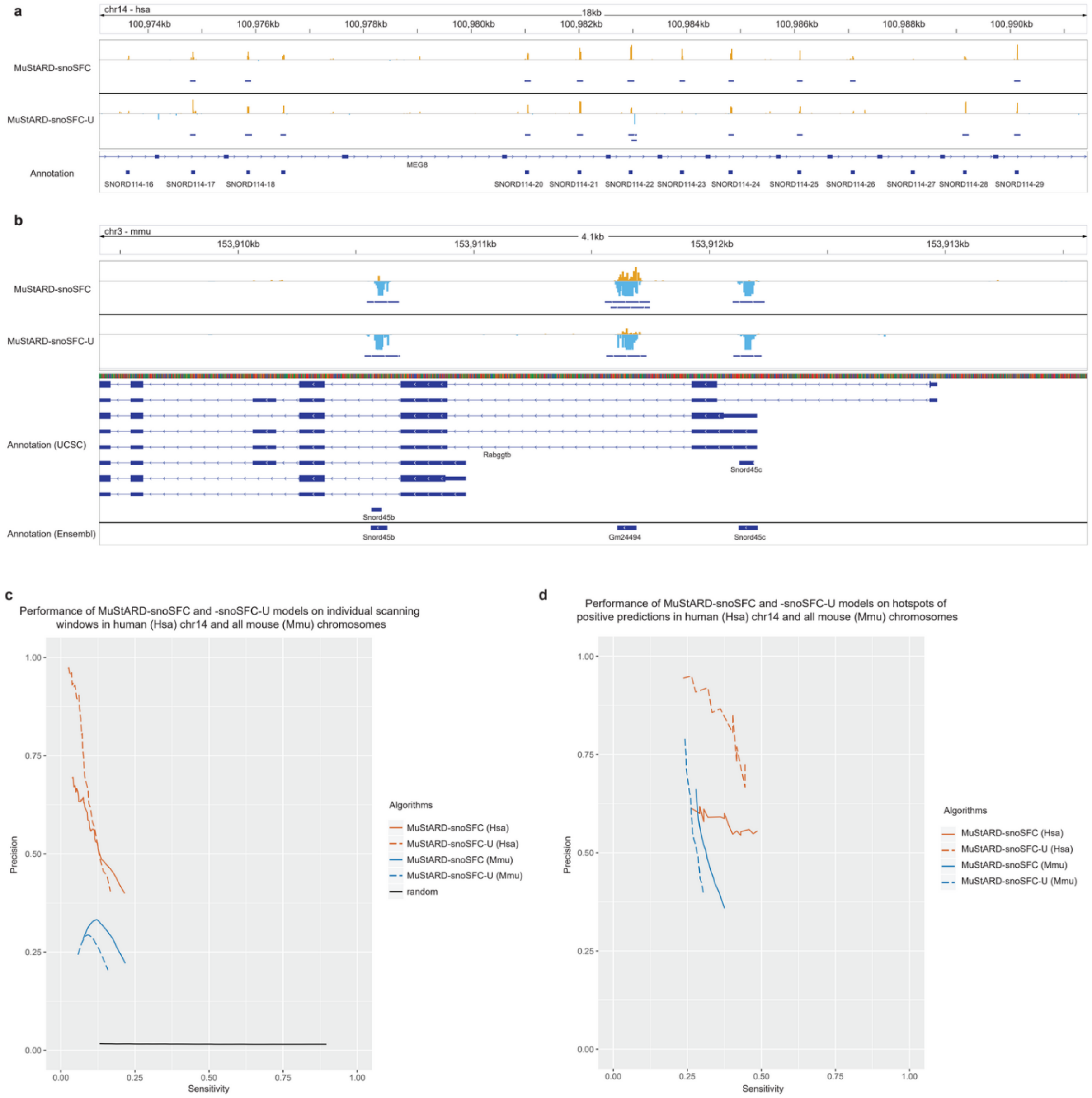

Figure 6. Performance of human MuStARD-snoSFC and -snoSFC-U models on human (chromosome 14) and mouse genomes. a) Visualization of both models on the scanning windows on MEG8 locus in human chromosome 14. b) Genome browser visualization of MuStARD snoRNA models on Rabggtb locus. c) Performance in terms of precision and sensitivity based on assessing scanning windows individually d) Precision and sensitivity performance on hotspots of positive predictions.

Discussion

We have presented a novel method able to learn from example and identify similar functional loci over large regions. We demonstrated an improvement in accuracy of prediction over several methods specifically developed for a single task with expert knowledge, and have furthermore, for the first time, successfully attempted a genomic scan in the scale of several million nucleotides. Finally, we point out the potential of MuStARD to annotate classes cross-species along moderate evolutionary distances.

An innovative aspect of our method involves the iterative selection of negative examples based on high scoring false positives. Machine Learning methods are only as good as their features and training set. While Deep Learning eliminates the need for expert curated features—some of the pre-miRNA prediction methods utilized up to 700 features[3]—the need for negative training sets that effectively capture most of the background variation is still crucial. We initially prototyped our method with a small set of negatives, four for each real training example. We quickly realized that while our method could separate between these categories easily, it still produced a large amount of false positives in the more realistic scanning test. Training fifty models on fifty sets of negatives improved the performance, but we noticed that specific regions were identified as false positives by a large number of models, i.e. the false positives were not randomly distributed in the background. By enriching our background set with these false positives and retraining the best models in this iterative fashion, we achieved a great leap in performance. An important point for the iterative background enrichment step is that it is fully automated within our method. This allows the method to generalize more easily, since the best background mixture for each class of functional elements cannot be known in advance.

The visualization tracks in Figure 3a highlight an important issue about the performance of algorithms on the task of scanning the genome for the identification or pre-miRNAs. The hotspots of positive predictions for all existing algorithms and especially for miPred, miRBoost and microPred exhibit a size that is typically several hundred base pairs. As a consequence, hotspots frequently overlap more than one annotated pre-miRNA creating positive bias for these algorithms in the precision and sensitivity mediated assessment of their performance. Figures 3d and 3e depict the distribution of the distance between the center of TP prediction hotspots and the center of overlapping annotated pre-miRNAs as well as the distribution of hotspot sizes. These results highlight that MuStARD provides high resolution predictions with unprecedented precision. Allowing a long merged prediction length does not give an advantage on the evaluation of our method. To the contrary, methods that predict a large number of positives tend to merge their predictions into long stretches, thus improving their evaluation metrics. For that reason, when evaluating algorithms for practical use we need to also take into account the tightness of the prediction fit to the positive region. MuStARD's predictions were the tightest fit on real loci of all evaluated algorithms while our method also achieved higher precision and sensitivity balance than other algorithms. A side product of our merging positive regions output is the possibility to use MuStARD to 'stitch together' longer functional loci. There is an open question of whether MuStARD or a similar method based on iterative enrichment of backgrounds could be used to identify exons, untranslated regions, promoters, enhancers, long non coding RNAs etc.

The evaluation of different input modes by itself gave us interesting insight in line with the scientific knowledge of pre-miRNAs. We managed a qualitative ranking of the contribution for each input branch to the final predictive model. Deeper interpretation of the model is beyond the scope of this paper, but is an exciting further field of research. One interesting observation coming from our training, is that class weight balanced training seems to be inferior in accurately predicting pre-miRNAs compared to unbalanced training. Class weight balancing is used in training Deep Learning models so that the model does not attempt to learn the characteristics of a disproportionately populous class while ignoring sparser classes. However, in our more realistic scanning test, one positive example corresponds to at least one hundred negative examples. Our training data with a maximum ratio of approximately one positive example to fifty negatives—although heavily unbalanced—is less unbalanced than the realistic testing data. Exploring the class balancing issue will be necessary for the further improvement of the field towards the ultimate goal of genome wide scan prediction.

Using a number of pre-miRNA prediction algorithms for region scanning was time consuming and arduous labor. To calculate hundreds of features on regions spanning less than one percent of the human genome, all other algorithms (with miRBoost being the sole exception) required to group the scanning region into smaller batches of 2000 sequences in order to parallelize the analysis into a computer cluster (MetaCentrum-CERIT). Even so, the average computing time for a single batch was 4 days. In contrast, our algorithms was able to scan the mouse benchmark dataset that includes several million base pairs in a few hours on one CPU. With GPU access enabled this process can be even faster. MuStARD has made the possibility of a full mammalian genome scan feasible on a high-end personal computer. However, even with our improved prediction accuracy, the number of false positives identified on a full genome scan would still be disproportionate to the true positives. We will continue exploring improvements and iterative training modification with the goal to achieve genome wide scan capabilities.

Given the increasing number of sequenced genomes becoming available, annotation is lagging. We have demonstrated that MuStARD can be efficiently trained on one species and then used to predict members of the same functional class in another. As a proof of concept we trained models on human pre-miRNAs and snoRNAs and then identified their counterparts in mouse. These species are both well annotated, but have a considerable evolutionary distance. The pre-miRNAs we correctly identified on the mouse genome were enriched in evolutionary conserved pre-miRNAs in human (approximately 30% of our true positive predictions vs 10% of all mouse miRNAs). That said, the majority (70%) of our predicted pre-miRNAs are not homologous to human pre-miRNAs and would not be easily identified by a simple homology search.

We chose pre-miRNAs as a first example because they have well established annotation, consistent secondary structure, conservation and other sequence characteristics. Our second use example was snoRNAs where most of these assumptions fail. The snoRNA class consists of several families that do not share common secondary structure, motif sequences, or conservation profiles. Conservation at large is much less pronounced in snoRNAs compared to pre-miRNAs. Additionally, the size distribution of snoRNAs (118.8bp mean, 59.1bp standard deviation) is much wider than miRNAs (81.9bp mean, 16.9bp standard deviation) making it harder for our method to accurately identify them. Despite these drawbacks, we manage to identify snoRNAs accurately within the human genome and in cross-species scan.

When it comes to development of MuStARD we focused on making our method versatile and extensible, but also easy to deploy and run with minimal user input. To make it modular and versatile we used a Keras architecture that can be easily extended from more experienced users. Keras is widely accepted by the Deep Learning community and offers ease of use and several layers of abstraction in terms of code sharing when compared to tensorflow or other lower end frameworks. Extending the architecture with more diverse branches is straightforward. We have already developed templates for reading sequence and secondary structure (one hot encoding) as well as conservation (scores). Adding different types of signal is as easy as downloading a relevant track from UCSC Genome Tracks and asking MuStARD to include it in the training. For basic users, the input has been kept minimal, requiring just a bed file of regions in the functional class of interest, a sequence file of the genome, and a conservation track of the same size. Given these inputs, MuStARD preprocesses the regions of interest, extracts sequences, simulates folding, picks random genomic sequences for background, optimizes hyperparameters, and so on until the final model is trained. With a trained model, region scans can be quickly with minimal effort.

Methods

Dataset collection and preprocessing

Human (GRCh38) and mouse (GRCm38) genomes and corresponding gene annotations were downloaded from Ensembl v93 repository[32]. Human gene annotation was filtered to only include genes exhibiting a protein-coding or lincRNA biotype. The exons of protein-coding and lincRNA genes were combined to produce two disjoint data sets; parts of the genome that correspond to exons and loci that are represented by introns. A separate collection was created by selecting regions marked with the snoRNA biotype. Human and mouse pre-miRNAs were downloaded from miRBase v22[7]. Human pre-miRNAs were subsequently filtered based on the experimentally validated information provided in miRBase to keep only high-quality sequences for training. Basewise conservation scores, based on phyloP algorithm, of 99 and 59 vertebrate genomes with human and mouse respectively were downloaded from UCSC genome repository[33].

MuStARD training module architecture

The aim of MuStARD is to provide a highly flexible, feature-agnostic computational framework that can be applied in a plethora of Biological problems providing state-of-the-art performance while at the same time having minimal input requirements. To this end, MuStARD has been specifically designed to follow a modular architecture where each module carries out different functionalities that can be run and assessed independently and/or in parallel (Figure 1). The framework is implemented using python for the deep learning aspect, R for the majority of meta-analyses and plotting, and perl for general purpose file filtering, formatting and module connectivity. Users only need to provide bed formatted files as input and the appropriate genome assembly files as well as the wiggle formatted PhyloP evolutionary conservation score files derived from UCSC repository.

The training module of MuStARD is composed of a convolutional architecture based on tensorflow and the Keras functional API. More advanced users can directly add or remove parts of the architecture according to the problem at hand. For the purposes of this study, the chosen architecture consists of 3 convolutional branches that can be dynamically added, removed and combined in multiple ways according to the properties of the corresponding use-case. These branches depict distinct ‘agents’ that are able to independently model different input modes such as raw DNA sequence, RNA secondary structure and evolutionary conservation. Subsequently, the outputs of the convolutional branches are flattened, concatenated and forwarded to the dense part of the architecture that produces the final prediction scores. In every layer output, dropout and batch normalization regularization techniques are applied to improve the generalisation capacity of the network. Regardless of the chosen network architecture, hyperparameters are known to be notoriously hard to optimize and depending on the complexity of the input, small changes in the hyperparameter selection can greatly affect the results. The training module has been designed to incorporate a grid-search type of approach for finding the optimal combination of hyperparameters. We have chosen to apply grid-search over 4 hyperparameters, the ones that based on our experience are able to greatly affect the results; batch size, learning rate, dropout rate and number of filters in the convolutional layers. Users can freely remove or add hyperparameters into the grid-search process and most importantly adjust the network architecture according to their needs. Each model trained over a different combination of hyperparameters is saved in a separate directory alongside train/validation accuracy/loss plots and a detailed log of the performance in each epoch. This allows users to find the exact combination of hyperparameters that produces the optimal training. Unless stated otherwise, in all use-cases presented in this study, each convolutional branch consists of 3 convolutional layers. The first convolutional layer in the raw DNA sequence processing branch uses a filter size of 16 nucleotides with stride 1 and no padding, the second layer uses a filter size of 12 and the third layer a filter size of 8. The first convolutional layer in the RNAfold processing branch uses a filter size of 30 nucleotides with stride 1 and no padding, the second layer uses a filter size of 20 and the third layer a filter size of 10. The first convolutional layer in the evolutionary conservation processing branch uses a filter size of 20 nucleotides with stride 1 and no padding, the second layer uses a filter size of 15 and the third layer a filter size of 10. The outputs of the convolutional branches are flattened and concatenated before being forwarded to the dense part of the network that includes 3 layers of 100, 75 and 50 nodes respectively. All layers use leaky ReLu activation except the final prediction layer that uses the softmax function. The chosen optimizer is SGD with Nesterov momentum set at 0.9. All models were trained over 600 epochs after enabling early stopping with patience set at 40 and delta at 0 with a learning rate of 10^{-4}. The code is accompanied with a configuration file and examples of how to edit parts of the architecture and train or test new or existing models.

MuStARD prediction module

The prediction module of MuStARD framework has been explicitly designed to facilitate both long region scanning and static assessment of specific loci. In the case of long region scanning, users are able to select the appropriate parameters such as the window size (it should match with the training window size), sliding step and the model that will be used for scoring each window. The framework includes standalone code for generating bedGraph tracks that can facilitate the visualization of results in any genome browser as well as code for creating 'hotspots' of positive predictions and for evaluating the results based on custom tracks and/or annotations. In the case of static assessment of specific loci, the prediction module provides a bed formatted file that included the score of each region in the 5th column.

Learning process of pre-miRNA detection MuStARD models

As described in previous sections, the training (Figure 1) of the pre-miRNA recognition model was based on experimentally verified human precursor sequences from miRBase. Only pre-miRNAs with size less than 100bp were used to form the positive set resulting in 579 sequences. The negative set was formulated with bedtools 2.27.0v[34] 'shuffle' mode using the positive set on the exon/intron genomic segments described in previous section. For each positive instance 4 equally sized negatives were randomly selected from protein- and non-coding exonic as well as intronic regions, 1 for every category. This process was repeated 50 times in total creating 50 different training sets that were used to train an equal amount of distinct preliminary models. Hyperparameters were fixed at 256 batch size, 0.2 dropout rate, 0.0001 learning rate and 80/40/20 number of filters in the 3 convolutional layers of each branch and the class weight option in Keras was enabled. Based on this repetitive negative shuffling configuration we ensured that a reasonable balance between training time as well approximating sequence and evolutionary conservation variation in background or non-precursor genomic loci was maintained. Instances of the training set that were located in chromosomes 2 and 3 were used for validation, instances in chromosome 14 were left out of the training process and all remaining instances were used for training. One of our objectives was to optimize the genome scanning process. The majority of existing algorithms utilize positive sequences that are fixated around the center of pre-miRNAs. However, in genome scanning scenarios there will always be instances in which part or the whole hairpin sequence will not be located in the center of the scanning window. This phenomenon might heavily affect the secondary structure of the RNA sequence corresponding to each window and therefore the generalisation capacity of the model. To overcome this problem, the MuStARD training module has been equipped with an optional 'reinforcement' feature that generates copies of the input instances with randomly placed positive or negative sequences within the 100nt sequence. For the purposes of this study, the number of reinforced instances for every input sequence was 5.

Ideally, if the combination of using intronic/exonic regions as a background sequence pool and the 1:4 positive to negative ratio was enough to fully capture the non-precursor sequence variation in the 3 input feature space (raw DNA sequence, secondary structure and basewise evolutionary conservation) then a near perfect performance in terms of both precision and sensitivity would be achieved in a scenario where all 50 preliminary models are used to scan the genome for predicting pre-miRNA sequences. To test this hypothesis all human pre-miRNAs were extended by +/- 5,000bp and the resulting regions were merged in the case of strand specific overlaps. Both strands of the merged loci were scanned with all 50 preliminary models using a window of 100bp and a stride of 10bp. This resulted in a benchmark dataset of 33.2 million bp divided into 3.2 million overlapping 100bp windows. For each model, out of the 3.2 million windows only those exhibiting a score above 0.5 were retained to form 'hotspots' of positively predicted regions after merging cases of strand specific overlaps. These regions were subsequently cross checked with the annotated pre-miRNAs to extract performance metrics in the 0.5-0.9 score range for every preliminary model (Supplementary Table 8).

False positive predictions based on a 0.5 score threshold were kept only if they were supported by 25 out of 50 preliminary models and did not overlap with any negative instance used to train these models. The resulting 23,750 false positive loci were added to the negative dataset of the best performing preliminary model. These false positives represent regions of the genome that were not captured by the process of 'shuffling' positive instances to exonic/intronic loci and exhibit feature characteristics that are more similar to positive than negative instances. This process assisted in establishing an enhanced set of sequences that was used to train the final pre-miRNA detection model that was selected through performing a hyperparameter space grid-search over the batch size and the Keras option of training with/without class weights (Supplementary Table 1). The class weights option in Keras enables the equal contribution of all classes during the training of unbalanced datasets. The remaining hyperparameters were not changed.

This process was repeated 6 times to train (Supplementary Table 8), with the Keras class weights option enabled, an equal number of distinct MuStARD pre-miRNA detection models composed of different input combinations; raw sequence with secondary structure and conservation (MuStARD-mirSFC model), raw sequence and conservation (MuStARD-mirSC), raw sequence and secondary structure (MuStARD-mirSF), secondary structure and conservation (MuStARD-mirFC), secondary structure only (MuStARD-mirF) and sequence only (MuStARD-mirS). For the combination of raw sequence, secondary structure and conservation, we have trained an additional model after disabling the class weights option in Keras (MuStARD-mirSFC-U model). For MuStARD-mirSFC model, the optimal (balance between precision and sensitivity) batch size was 1024, 256 for MuStARD-mirSFC-U, 1024 for MuStARD-mirSC, 256 for MuStARD-mirSF and MuStARD-mirFC, 512 for MuStARD-mirF and MuStARD-mirS. The procedure for evaluating the performance of each model is described in the following section.

For the purposes of the second use-case presented in this study, the same pipeline was used to create two MuStARD snoRNA detection models (Supplementary Table 8) using raw sequence, secondary structure and conservation with the class weights Keras option enabled (MuStARD-snoSFC) and disabled (MuStARD-snoSFC-U). However, for this use-case only snoRNAs with size less than 100bp were used to form the positive set resulting in 386 positive sequences.

Testing on genomic region scanning data

The process of testing algorithms on a static labelled dataset can often provide misleading results about performance especially in cases of models that have been designed for genome-wide scanning. Such 'stress' tests are often also able to unveil interesting aspects about the computational complexity and the time required by algorithms to complete a task. To this end, all human pre-miRNAs located on chromosome 14 were extended by +/- 5,000bp and the resulting regions were merged in the case of strand specific overlaps. Both strands of the merged loci were scanned with all MuStARD's final pre-miRNA detection models (Figure 2, Supplementary Table 1) as well as with existing algorithms (Figure 4, Supplementary Table 3) using a window of 100bp and a stride of 5bp. This resulted in a scanning benchmark dataset of 1 million bp divided into 208,708 overlapping 100bp windows.

Two distinct strategies were employed to assess the performance of each algorithm. In the first approach, each window was assessed independently (method A) while in the second only windows exhibiting a score above 0.5 were retained to form ‘hotspots’ of positively predicted regions after merging cases of strand specific overlaps (method B). These regions were subsequently cross checked with the annotated chromosome 14 pre-miRNAs (99 in total) to extract performance metrics in the 0.5-0.9 score range, when possible. In both scenarios, positive predictions were considered true positives (TPs) if they covered at least 50% of the overlapping annotated pre-miRNA’s length. HuntMi[27] and microPred[28] algorithms only provide hard labelled results instead of a probabilistic score, therefore they were not included in the graph. However, to facilitate a fair comparison between all algorithms, performance metrics based on both methods A and B were extracted at a fixed score threshold of 0.5 (Supplementary Tables 2 and 3).

Method B was also applied to annotated mouse precursors as well as human/mouse snoRNAs but only using MuStARD’s SFC and SFC-U use-case relevant models (Figure 6, Supplementary Tables 6 and 7).

Testing on labelled data

For the purposes of testing the final pre-miRNA detection model on labelled data and comparing with existing algorithms, all positive and negative instances located on chromosome 14 were used from the enhanced data set described in the previous section after removing sequences with size less than 100bp. The total number of positive instances in the test set was 44 and the total number of negatives 893 (Figure 4, Supplementary Table 3).

Application of existing algorithms

There are over 30 pre-miRNA prediction algorithms listed in OMICtools repository. The majority of these studies provide access to the trained models only through web-server interfaces which allow a small number of sequences to be processed at once. Only a handful of studies provide stand-alone implementations that can be downloaded and applied on benchmark datasets locally. However, a small fraction of these implementations are able to properly function and provide results.

We only managed to assess the prediction efficiency of HuntMi, microPred, MiPred, triplet-SVM, and MirBoost on our benchmark datasets. HuntMi and microPred tools do not support parallelization, and the average processing time for a sequence of 100nt is 3 minutes. The scanning benchmark sequences were grouped into 100 bins to faster the analysis for HuntMi and microPred. Also, microPred random sequence generation parameter setting was 500. Each bin was analyzed independently by HuntMi and microPred on virtual machines provided by the MetaCentrum-CESNET supercomputer cluster. MiRBoost’s SVM model was re-trained to support probabilistic output using the dataset included in the code repository and parameters ‘svm-train -h 0 -c 8.0 -g 0.125 -w1 1 -w-1 1 -b 1’. Then miRBoost was applied on our benchmark dataset with parameters ‘miRBoost -d 0.25’. For triplet-SVM, we initially applied RNAfold on our benchmark dataset with parameters ‘RNAfold --noPS --noconv --jobs=10’ and the output was forwarded to the triplet-SVM perl script with parameters ‘triplet_svm_classifier.pl 22’ that pre-processes the data and reformats it for the final prediction modules that requires libsvm. The final triplet-SVM results were obtained using svm-predict with parameters ‘svm-predict -b 1’. MiPred was applied on the benchmark dataset with default parameters.

Assessing MuStARD’s ability to detect non-human-homologous pre-miRNAs in mouse

Mouse hairpins regions of miRNA transcripts (N=1227) were derived from the miRBase database; orthologous miRNA (N=129) between mouse and human were retrieved from the Ensembl BioMart hub[35]. Initially, accurate MuStARD predictions (true positives) were recognized as overlapping with mouse hairpins regions through bedtools intersect v2.27.1. Subsequently, non-human-homologous pre-miRNAs were distinguished as the negative intersection between accurate MuStARD predictions and the human orthologous miRNA dataset. Bedtools options 'same strandedness' and 'overlaps=0.5' were used in both cases (-s and -f, respectively).

Software and hardware requirements

MuStARD is developed in Python 2.7 for the Deep Learning aspect (tensorflow 1.10 and Keras 2.2.2), R for visualizing the performance and Perl for file processing, reformatting and module connectivity. Full list of dependencies can be found on MuStARD’s gitlab page.

MuStARD is able to execute either on CPU or GPU depending on the underlying hardware configuration by taking into advantage tensorflow’s flexibility. The framework has been designed to maintain a minimal memory footprint thus allowing the execution even on personal computers. Running time heavily depends on input dimensionality, number of instances in the training set, learning rate and GPU availability.

Acknowledgments

This research was funded by Postdoc@MUNI with project registration number CZ.02.2.69/0.0/0.0/16_027/0008360 grant to G.K.G. We would like to also acknowledge the fellowship program Brno Ph.D. talent 2017 to A.G., ‘MetaCentrum NGI’ for providing computational infrastructure necessary for running parts of the analyses presented in this study.

Author Contributions

P.A. and G.K.G. conceived the study. P.A. oversaw the whole study. G.K.G. developed the code, implemented all analyses and comparisons. A.G. applied HuntMi and microPred on the benchmark datasets and performed the mouse/human pre-miRNA homology analysis. G.K.G. and E.M. applied miPred on the benchmark datasets. K.G.L. and F.C.P. applied miRBoost and triplet-SVM on the benchmark datasets. P.A. and G.K.G. wrote the manuscript and prepared the figures.

Competing Interests

The authors declare no competing interests.

References

1. Lander, E. S. *et al.* Initial sequencing and analysis of the human genome. *Nature* **409,** 860–921 (2001).
2. Suzuki, Y. *et al.* Sequence comparison of human and mouse genes reveals a homologous block structure in the promoter regions. *Genome Res.* **14,** 1711–1718 (2004).
3. Saçar Demirci, M. D., Baumbach, J. & Allmer, J. On the performance of pre-microRNA detection algorithms. *Nat. Commun.* **8,** 330 (2017).
4. Lagos-Quintana, M., Rauhut, R., Meyer, J., Borkhardt, A. & Tuschl, T. New microRNAs from mouse and human. *RNA* **9,** 175–179 (2003).
5. Lee, R. C. & Ambros, V. An extensive class of small RNAs in Caenorhabditis elegans. *Science* **294,** 862–864 (2001).
6. Lau, N. C., Lim, L. P., Weinstein, E. G. & Bartel, D. P. An abundant class of tiny RNAs with probable regulatory roles in Caenorhabditis elegans. *Science* **294,** 858–862 (2001).
7. Kozomara, A., Birgaoanu, M. & Griffiths-Jones, S. miRBase: from microRNA sequences to function. *Nucleic Acids Res.* (2018). doi:10.1093/nar/gky1141
8. Kiss, T. Small nucleolar RNAs: an abundant group of noncoding RNAs with diverse cellular functions. *Cell* **109,** 145–148 (2002).
9. Yang, J.-H. *et al.* snoSeeker: an advanced computational package for screening of guide and orphan snoRNA genes in the human genome. *Nucleic Acids Res.* **34,** 5112–5123 (2006).
10. Yang, J.-H., Shao, P., Zhou, H., Chen, Y.-Q. & Qu, L.-H. deepBase: a database for deeply annotating and mining deep sequencing data. *Nucleic Acids Res.* **38,** D123 (2010).
11. Fitch, F. B. McCulloch Warren S. and Pitts Walter. A logical calculus of the ideas immanent in nervous activity. Bulletin of mathematical biophysics, vol. 5 (1943), pp. 115–133. *J. Symbolic Logic* **9,** 49–50 (1944).
12. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521,** 436–444 (2015).
13. Baldi, P., Sadowski, P. & Whiteson, D. Searching for exotic particles in high-energy physics with deep learning. *Nat. Commun.* **5,** 4308 (2014).

14. Yu, F., Thayer, M., Qasemi, E., Zhu, K. & Assadi, A. Deep Learning Features in Atmospheric Chemistry: Prediction of Cancer Morbidity Due to Air Pollution. in *2017 International Conference on Computational Science and Computational Intelligence (CSCI)* (2017). doi:10.1109/csci.2017.307

15. Ching, T. *et al.* Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface* **15,** (2018).

16. Singh, R., Lanchantin, J., Robins, G. & Qi, Y. DeepChrome: deep-learning for predicting gene expression from histone modifications. *Bioinformatics* **32,** i639–i648 (2016).

17. Liang, M., Li, Z., Chen, T. & Zeng, J. Integrative Data Analysis of Multi-Platform Cancer Data with a Multimodal Deep Learning Approach. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **12,** 928–937 (2015).

18. 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).

19. Zhou, J. & Troyanskaya, O. G. Predicting effects of noncoding variants with deep learning-based sequence model. *Nat. Methods* **12,** 931–934 (2015).

20. 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 (2016).

21. Altman, R. B. *et al. Biocomputing 2017: Proceedings of the Pacific Symposium*. (World Scientific Publishing Company, 2016).

22. Xu Min, Min, X., Chen, N., Chen, T. & Jiang, R. DeepEnhancer: Predicting enhancers by convolutional neural networks. in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (2016). doi:10.1109/bibm.2016.7822593

23. Lorenz, R. *et al.* ViennaRNA Package 2.0. *Algorithms Mol. Biol.* **6,** 26 (2011).

24. Pollard, K. S., Hubisz, M. J., Rosenbloom, K. R. & Siepel, A. Detection of nonneutral substitution rates on mammalian phylogenies. *Genome Res.* **20,** 110–121 (2010).

25. Roden, C. *et al.* Novel determinants of mammalian primary microRNA processing revealed by systematic evaluation of hairpin-containing transcripts and human genetic variation. *Genome Res.* **27,** 374–384 (2017).

26. Henry, V. J., Bandrowski, A. E., Pepin, A.-S., Gonzalez, B. J. & Desfeux, A. OMICtools: an informative directory for multi-omic data analysis. *Database* **2014,** (2014).

27. Gudyś, A., Szcześniak, M. W., Sikora, M. & Makałowska, I. HuntMi: an efficient and taxon-specific approach in pre-miRNA identification. *BMC Bioinformatics* **14,** 83 (2013).

28. Batuwita, R. & Palade, V. microPred: effective classification of pre-miRNAs for human miRNA gene prediction. *Bioinformatics* **25,** 989–995 (2009).

29. Jiang, P. *et al.* MiPred: classification of real and pseudo microRNA precursors using random forest prediction model with combined features. *Nucleic Acids Res.* **35,** W339–44 (2007).

30. Tran, V. D. T., Tempel, S., Zerath, B., Zehraoui, F. & Tahi, F. miRBoost: boosting support vector machines for microRNA precursor classification. *RNA* **21,** 775–785 (2015).

31. Xue, C. *et al.* Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. *BMC Bioinformatics* **6,** 310 (2005).

32. Zerbino, D. R. *et al.* Ensembl 2018. *Nucleic Acids Res.* **46,** D754–D761 (2018).

33. Karolchik, D. *et al.* The UCSC Table Browser data retrieval tool. *Nucleic Acids Res.* **32,** D493–6 (2004).

34. Quinlan, A. R. & Hall, I. M. BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26,** 841–842 (2010).

35. Kinsella, R. J. *et al.* Ensembl BioMarts: a hub for data retrieval across taxonomic space. *Database* **2011,** bar030 (2011).