Li *et al. BMC Genomics* 2020, **21**(Suppl 11):893
https://doi.org/10.1186/s12864-020-07296-1

BMC Genomics

RESEARCH Open Access

Deep learning for HGT insertion sites recognition

Chen Li, Jiaxing Chen and Shuai Cheng Li*

From The International Conference on Intelligent Biology and Medicine (ICIBM) 2020
Virtual. 9-10 August 2020

*Correspondence:
shuaicli@cityu.edu.hk
Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong SAR, HongKong, China

Abstract

Background: Horizontal Gene Transfer (HGT) refers to the sharing of genetic materials between distant species that are not in a parent-offspring relationship. The HGT insertion sites are important to understand the HGT mechanisms. Recent studies in main agents of HGT, such as transposon and plasmid, demonstrate that insertion sites usually hold specific sequence features. This motivates us to find a method to infer HGT insertion sites according to sequence features.

Results: In this paper, we propose a deep residual network, DeepHGT, to recognize HGT insertion sites. To train DeepHGT, we extracted about 1.55 million sequence segments as training instances from 262 metagenomic samples, where the ratio between positive instances and negative instances is about 1:1. These segments are randomly partitioned into three subsets: 80% of them as the training set, 10% as the validation set, and the remaining 10% as the test set. The training loss of DeepHGT is 0.4163 and the validation loss is 0.423. On the test set, DeepHGT has achieved the area under curve (AUC) value of 0.8782. Furthermore, in order to further evaluate the generalization of DeepHGT, we constructed an independent test set containing 689,312 sequence segments from another 147 gut metagenomic samples. DeepHGT has achieved the AUC value of 0.8428, which approaches the previous test AUC value. As a comparison, the gradient boosting classifier model implemented in PyFeat achieve an AUC value of 0.694 and 0.686 on the above two test sets, respectively. Furthermore, DeepHGT could learn discriminant sequence features; for example, DeepHGT has learned a sequence pattern of palindromic subsequences as a significantly (P-value=0.0182) local feature. Hence, DeepHGT is a reliable model to recognize the HGT insertion site.

Conclusion: DeepHGT is the first deep learning model that can accurately recognize HGT insertion sites on genomes according to the sequence pattern.

Keywords: Deep residual model, HGT insertion site, DNA sequence feature

© The Author(s). 2020 **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/. The Creative Commons Public Domain Dedication waiver (http://creativecommons.org/publicdomain/zero/1.0/) applies to the data made available in this article, unless otherwise stated in a credit line to the data.

Background

Horizontal Gene Transfer(HGT) [1] refers to the sharing of genetic materials between distant species that are not in a parent-offspring relationship [2]. HGT allows different species to share genomic fragments, thus creating a complex network among different species [3]. It is the fundamental mechanism for the spread of antibiotic resistance in bacteria [4, 5] and plays an important role in the evolution of bacteria [6–8]. Conjugation [9], transformation [10], and transduction [11] are the three most recognized mechanisms for HGT. Conjugation requires physical contact between a donor cell and a recipient cell. Then the genetic material, such as conjugative transposons [12], is transferred through plasmids. Transformation is the uptake of foreign genetic material from the surrounding environment and is relatively common in bacteria. Transduction is mainly mediated by phage and could occur more than 1,000 times in specific environments [13]. Through these mechanisms, functional unit of DNA, such as operon [14], and mobile genetic elements [15], such as transposons containing antibiotic resistance genes, could be incorporated into the genome of recipients [12]. Therefore, HGTs improve the bacteria's ability to adapt to changing environments. HGTs are often observed and well studied in prokaryotes. Recent research demonstrates that around 80% of genes in prokaryotes were involved in HGT at some point in their history [16]. HGTs could also occur between prokaryotes and eukaryotes [17]. From prokaryotes, eukaryotes acquire genes that are helpful to detoxify novel environments [18–20]. Moreover, through HGT, many eukaryotes benefit from the acquisition of genes encoding biosynthetic enzymes to live in extremely nutrient-poor environments [21, 22].

Mobile genetic elements (MGE), such as transposons, are the main agents of HGT [15]. Existing research on transposons demonstrates that the transposon ends usually have special sequence features, such as inverted repeats [23], AT-rich [12], etc. These sequence features make transposon easily transferred across cells by plasmids, phage, or integrative conjugative elements (ICEs). Other agents of HGT may also have specific sequence features [24]. These facts make it possible to recognize HGT insertion sites according to the sequence features at the sites. Deep learning is such a powerful method to extract features from DNA sequences. It is a class of machine learning algorithms based on artificial neural networks. It allows computational models composed of multiple processing layers to learn representations for data with multiple levels of abstraction [25]. Starting from 2012, deep learning has achieved great performance breakthroughs in computer vision [26, 27], speech recognition [28], and so on. More recently, deep learning was adopted to process DNA sequence data and Convolutional Neural Networks (CNN) is the most wildly used deep learning model in the field of bioinformatics. In 2015, [29] proposed DeepBind to predict DNA and RNA-binding proteins based on in vitro and in vivo assays. It performs better than the state-of-the-art methods from the DREAM5 in vitro TF-DNA motif recognition challenge [30]. Zhou and Troyanskaya [31] developed DeepSEA to predict chromatin effects of sequence alterations with single-nucleotide sensitivity by learning regulatory sequence codes from large-scale chromatin-profiling data. Researchers also have used CNN models to predict functional elements in the genome, such as promoter [32] and enhancer [33]. These applications imply that deep learning could effectively learn features from raw DNA sequences to perform the classification task. This motivates us to propose a deep learning model, named DeepHGT (https://github.com/lichen2018/

DeepHGT), which learns sequence features to recognize HGT insertion sites on reference sequences.

In order to train DeepHGT, we should get DNA sequences at HGT insertion sites. We utilize a traditional alignment tool LEMON [34] which is based on split reads realignment [35] and DBSCAN [36] to detect and label HGT insertion sites. Then we could collect DNA sequences at the detected HGT sites. In order to prove the specialty of DeepHGT, we also compare its performance with other machine learning models implemented in PyFeat [37]. PyFeat generates features from DNA sequences to train machine learning models. The generated features include zCurve, gcContent, ATGC ratio, Cumulative Skew, Chou's Pseudo composition, gap-based K-mer frequency, and so on. These features could capture the frequency distributions of various permutations of the base nucleotide in the sequences [37].

As described in the "Methods" section, by utilizing LEMON we collect a set of 1,556,694 sequence segments from 262 metagenomic samples [38]. 50% of the set are positive samples that are extracted at HGT insertion sites on reference genomes. The remaining sequences are negative samples. The set is randomly partitioned into three subsets: 80% of them as the training set, 10% as the validation set, and the remaining 10% as the test set. DeepHGT has achieved Area under the Curve of ROC (AUC) value of 0.8782 and Average-Precision (AP) value of 0.899 in the test set. Compared to the performance of four machine learning models implemented in PyFeat, features learned by DeepHGT are more discriminant than those generated by PyFeat and make DeepHGT achieve better performance. Besides, 125 correctly classified positive test sequences at HGT insertion sites contain palindromic subsequences. For each sequence, any continuous subsequence can be treated as a local feature. We define HGT-Index to measure the contribution of its local feature to the prediction value of the sequence. Statistic test results demonstrate that palindromic subsequences, which are typical sequence patterns in MGE, are significantly local features. In addition, to further evaluate the generalization of DeepHGT, we obtain an independent test set of 689,312 sequence segments from 147 metagenomic samples [39] using LEMON. The ratio between positive and negative samples is 1:1. DeepHGT has achieved the AUC value of 0.8428 and the AP value of 0.8743, which supports the good generalization of DeepHGT. So DeepHGT can accurately recognize HGT insertion sites on genomes according to sequence pattern.

Results

Percentage distribution of positive samples at species/genus level

The 778,347 positive sequences extracted from 262 metagenomic samples belong to 3,070 species and 711 genera. Table 1 summarized the percentage distribution of the top 10 most abundant species/genera. As we can see, *Microbacterium esteraromaticum* is the species to which the greatest number of positive sequences belong. Its percentage is only 13.13%. The percentages of the other nine species are less than 10%. Furthermore, we calculate the percentage distribution of the 3,070 species. Its standard variance is 0.30%. *Microbacterium* is the genus to which the greatest number of sequences belong. Its percentage is 14.38%. The standard variance for the percentage distribution of the 711 genera is 0.92%. Therefore, sequences are evenly distributed across the 3,070 species and 711 genera. In another word, sequences are not enriched to a small number of species/genus. Therefore, the dataset is balanced at the appropriate species/genus level. Since this dataset

Table 1 Percentage distribution of Top 10 most abundant species/genera to which positive samples belong

Top 10 Species	Percentage (%)	Top 10 Genera	Percentage (%)
Microbacterium esteraromaticum	13.13	Microbacterium	14.38
Mycolicibacterium monacense	7.36	Bacteroides	12.35
Mycobacterium sp. 852002-51961_SCH5331710	3.08	Bifidobacterium	8.00
Faecalibacterium prausnitzii A2-165	2.39	Mycolicibacterium	7.73
Collinsella aerofaciens ATCC 25986	1.97	Mycobacterium	6.21
Collinsella sp. 4_8_47FAA	1.94	Collinsella	5.89
Gemmiger formicilis	1.69	Clostridium	3.23
Collinsella sp. TF06-26	1.64	Faecalibacterium	2.70
Bifidobacterium longum	1.55	Alistipes	2.49
Bacteroides caccae	1.50	Roseburia	2.37

is mainly used to train and validate DeepHGT, we call this dataset as the positive training dataset. The 344,656 positive sequences extracted from 147 metagenomic samples in the independent test set belong to 2,139 species. Appendix Table 7 compares the two percentage distributions of the top 10 most abundant species to which sequences in the positive training dataset and the independent positive test dataset belong. As we can see sequences in the two datasets have very different composition at species level.

Overview of DeepHGT

We propose DeepHGT as illustrated in Fig. 1. DeepHGT is a deep residual neural network [40], which contains four residual blocks. Each residual block contains two *Conv+BN+Relu* sub-blocks and one skip-connection, which directly connects the input and output of the residual block, here *Conv* denotes the Convolutional layer, *BN* denotes the Batch Normalization layer [41], and *Relu* denotes the ReLU activation layer. In general, as we increase the number of layers in the neural network, its performance on both training and test data will decrease, this is called the degradation problem [40]. By adding skip-connection to skip some layers, the residual neural network is equal to the integration of multiple neural networks with different depths. This solves the degradation problem and makes the residual neural network go deeper to extract more mid-level and high-level features than shallow models. These extra features also enable the residual neural network to achieve better performance than shallow models. In order to improve the generalization performance of DeepHGT, we add one Dropout layer [42] behind each residual block. Dropout is an efficient trick to reduce overfitting during training. By randomly dropping hidden nodes, the training process is equivalent to training a large number of neural networks with different architectures in parallel. This makes DeepHGT learn more robust features thus better generalize to new data.

DeepHGT is implemented by using Keras and contains 2,119,297 trainable parameters. We set the length of the input sequence as 100 [43] and convert each sequence to a 100×4 matrix using the one-hot encoding method, where each position corresponds to a four-element vector with one nucleotide's bit set to one [33]. All convolutional layers in DeepHGT have the same number of filters 128 and the same kernel size 4×1 with slide step 1. The first two Dropout layers have the same dropout rate 0.1, the dropout rate of the third Dropout layer is 0.25, and the dropout rate of the last Dropout layer is 0.5. By setting the dropout rate small in lower Dropout layers, we could maintain most low-level features. The large dropout rate in higher Dropout layers is helpful for learning useful

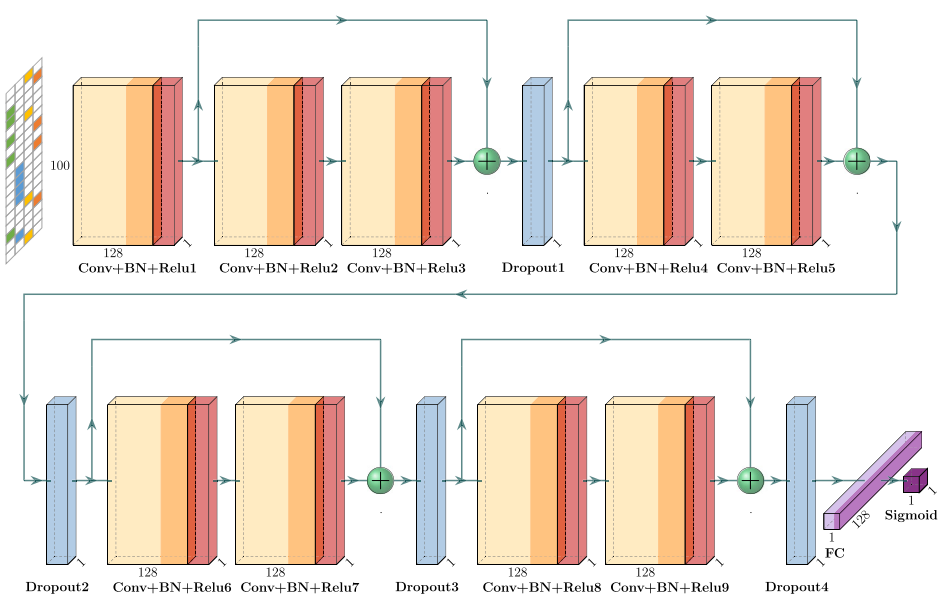

Fig. 1 Overview of DeepHGT

high-level features. Right behind the last Dropout layer is a fully connected layer, which contains 128 units. Since our task is a binary classification problem, the output layer is a *Sigmoid* function. We don't apply the pooling layer in DeepHGT since we found that pooling layers reduce the spatial dimensions of feature vector by a factor of 2, which leads to the loss of too much feature information and decreases the performance of DeepHGT in experiments.

DeepHGT predicts HGT insertion sites

We set the batch size as 120 and utilize Stochastic Gradient Descent (SGD) to minimize the objective loss function of DeepHGT. The learning rate is 0.001. Since our task is to predict whether the input sample is positive, we set the loss function as binary cross-entropy. The number of epochs for training is 2,900. By changing the number of residual blocks in DeepHGT, we get other four deep learning models as comparisons. Their performance is measured by AUC and precision-recall curve. The precision-recall curve shows the tradeoff between precision and recall for different thresholds and Average-Precision (AP) is the weighted mean of precisions achieved at each threshold.

The set of 1,556,694 sequence is randomly partitioned into three subsets: 80% of them as the training set, 10% as the validation set, and the remaining 10% as the test set. The DeepHGT is trained on one NVIDIA Tesla V100 GPU. Figure 2 illustrates the evolution of training and validation loss during the training process of DeepHGT. During the first 200 epochs, both of train loss and validation loss decrease fast, this demonstrates that DeepHGT efficiently learns useful sequence features from the training dataset to distinguish positive and negative samples. Since we have utilized data augmentation methods on the training set and not on the validation set, this makes the training set become more diverse and contain more hard samples to train. Therefore, the training loss is larger than the validation loss during 2,000 epochs. After 2,200 epochs, the validation loss fluctuates

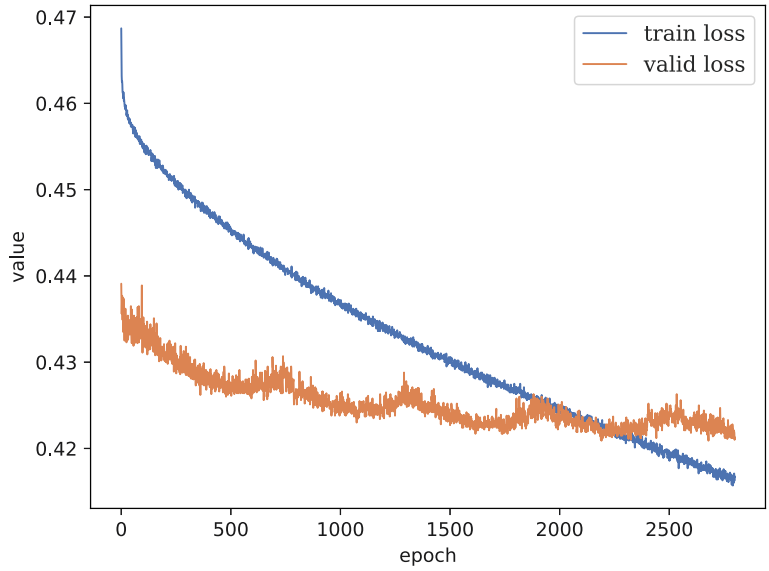

Fig. 2 Training and validation loss evolution during the training of DeepHGT

around 0.423, while the training loss continues to decrease rapidly and become lower than validation loss. This is because DeepHGT gets stuck in a local minimum, and continuing training makes DeepHGT overfit the training data without achieving better performance on the validation set. Therefore, we stop the training process after 2,900 epochs.

Figure 3a and b illustrate the Receiver OperatingCharacteristic (ROC) and Precision-Recall curves of DeepHGT and the other five deep learning models with different architectures. 1#Res_Block denotes that the deep residual model contains 1 residual block, 2#Res_Block denotes that the deep residual model contains 2 residual blocks, and so on. CNN_model has the same number of convolutional blocks as DeepHGT without skip-connections. DeepHGT contains 4 residual blocks as shown in Fig. 1. DeepHGT has achieved the highest AUC value of 0.8782 and the AP value of 0.8994. As we can see, by increasing the number of residual blocks, the deep learning model achieves better performance, which means that deeper models can learn more high-level features and get better generalization than shallow models. DeepHGT has better performance than CNN_model, which validates skip-connections are useful to prevent overfitting. Therefore, designing deep learning models with proper architectures is important to achieve good performance. Figure 3c and d compare the performance of DeepHGT and four machine learning models implemented in PyFeat including Naive Bayes [44] (PyFeat_NB), Adaboost Classifier [45] (PyFeat_AB), Random Forest [46] (PyFeat_RF), and Gradient Boosting Classifier [47] (PyFeat_GB). PyFeat extract features from training and test datasets. The features are then used to train and test the four machine learning models. Their parameters are set in PyFeat. Compared to the four models in PyFeat, DeepHGT has achieved the best performance, although PyFeat_GB has achieved AUC value 0.694 and AP value 0.76 based on the features extracted by PyFeat. Table 2 compares the accuracy of DeepHGT and other methods. DeepHGT achieved the highest accuracy score of 0.794. Therefore, sequence features learned by DeepHGT are more general and efficient than those extracted by PyFeat. And this also makes DeepHGT achieve better classifier performance than the other four models in PyFeat.

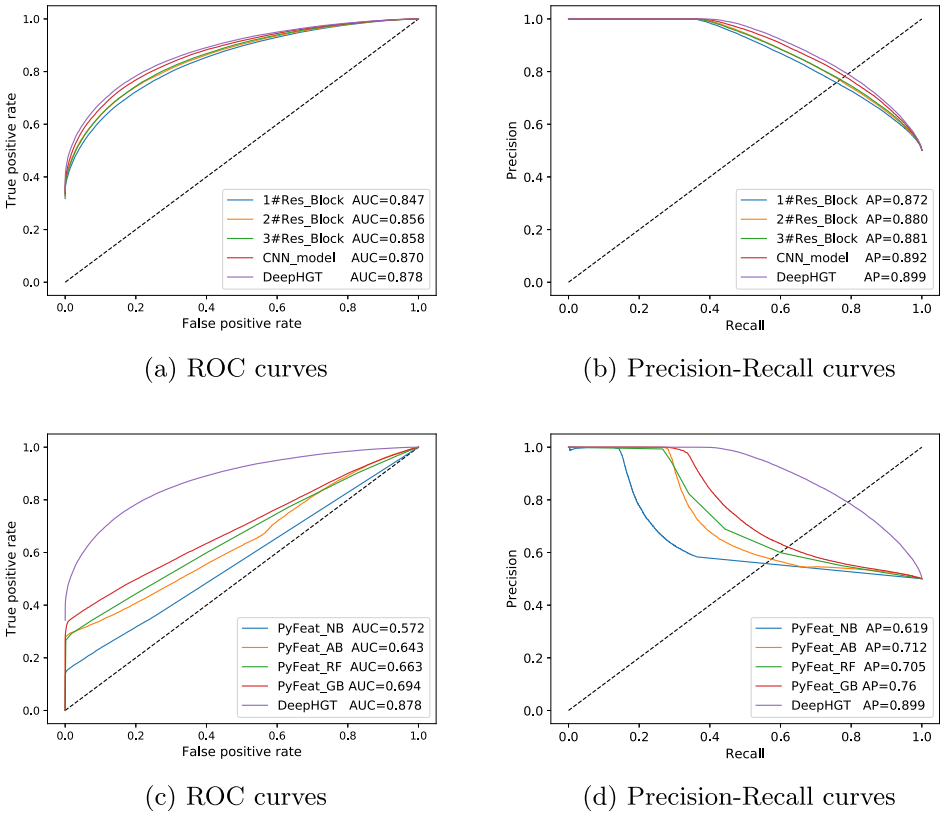

Fig. 3 **a** and **b** compare ROC and Precision-Recall curves of DeepHGT and the other five deep learning models. DeepHGT has achieved the highest AUC value 0.878 and AP value 0.899. **c** and **d** compare the performance of DeepHGT and four machine learning models implemented in PyFeat: Naive Bayes (PyFeat_NB), Adaboost Classifier (PyFeat_AB), Random Forest (PyFeat_RF), and Gradient Boosting Classifier (PyFeat_GB)

We then perform the Pairwise Delong test on AUCs of these models. The null hypothesis is that two ROC curves have the same AUC values. Small p-value denotes two AUC values are significantly different, which means the two models have significantly different performance. As illustrated in Appendix Table 8, all pairwise Delong test results have a p-value of less than 0.05, which means all models have significantly different performance.

DeepHGT predicts HGT insertion sites in different species

The test dataset consists of samples from several species, such as *Streptomyces griseofuscus*, *Oscillibacter sp. ER4*, *Roseburia inulinivorans*, *Acidovorax sp. SD340*, etc. We divide positive test samples into subsets based on species to which samples belong. Since each subset contains only positive samples from the same species, we randomly extract the same number of negative samples from reference sequences of the species. We then use these test subsets to evaluate the AUC values of DeepHGT on predicting HGT insertion sites in different species as illustrated in Fig. 4. AUC values for species *Acidovorax sp. SD340*, *Mycolicibacterium monacense* and *Streptomyces griseofuscus* approach to 1.

Table 2 Comparison of accuracy of DeepHGT and other methods

1#Res_Block	2#Res_Block	3#Res_Block	CNN_model	DeepHGT
0.761	0.772	0.773	0.781	0.794
PyFeat_rf	PyFeat_ab	PyFeat_gb	PyFeat_nb	DeepHGT
0.597	0.636	0.653	0.500	0.794

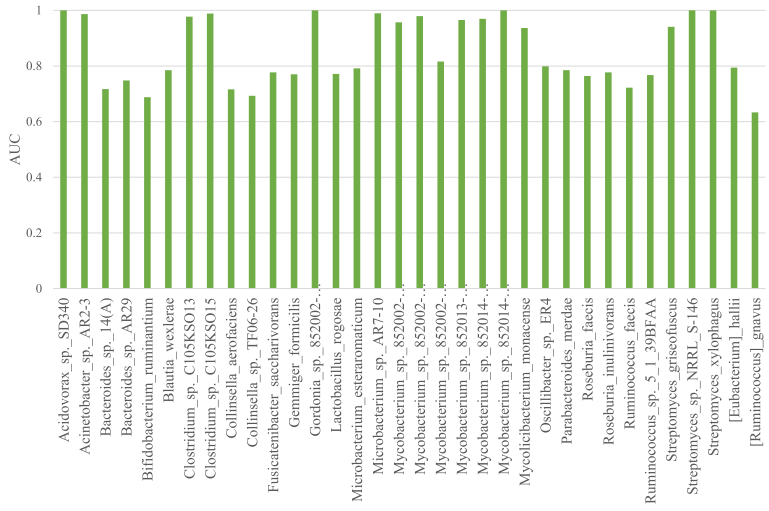

Fig. 4 AUC values of DeepHGT on predicting HGT insertion sites in different species

DeepHGT learns palindromic pattern at HGT insertion sites

The Repetitive Extragenic Palindromic (REP) sequence including an inverted repeat may have single or multiple adjacent copies [48]. A DNA sequence is considered as palindromic if it is equal to its reverse complement. For example, the DNA sequence CGTTGGCAACG is palindromic because its complement is GCAACCGTTGC, and reversing the complement gives CGTTGGCAACG, which is equal to the original sequence. The palindromic pattern makes REP sequences specific targets for some insertion sequence elements such as ISPsy8 in *Pseudomonas syringae* DC3000, ISPa11 in *P. aeruginosa* PA01, and so on [49]. This also demonstrates that REP elements are hot spots for specific transposition [50], which plays an important role in HGT, and contribute to genome instability, bacterial evolution, and speciation [51].

To test whether DeepHGT has learned sequence patterns such as palindromic subsequences as a significant local feature to recognize HGT insertion sites. First, we should measure the importance of palindromic patterns in the prediction process. We choose an optimal threshold to select correctly classified positive samples from the test dataset since the prediction value made by DeepHGT for each test sample is between 0 and 1. The optimal threshold in our experiment is set as 0.4935, which is calculated according to Youden's J statistic [52]. From the correctly classified positive samples, each of which has true label as 1 and prediction value larger than the optimal threshold, we find 125 samples containing palindromic subsequence S_{palin}={s_i, $i = 1, ..., 125$}, such as the palindromic sequence TAAAAAGATAAGTTGAATATTCAACTTATCTTTTTA in the positive sample AACAAGGAATTGGATTATAAAATTGTAAAAAGATAAGTTGAATATTCAACTTATCTTTTTATTTCACGTCTATTTATCTTAAAACCTATTTTTTCTTCTATTTCTTTTAGCTGATTTTCA. Then, for each sample s_i in S_{palin}, we measure the importance of its palindromic subsequences in affecting the prediction result. By comparing palindromic subsequences to randomly selected subsequences, we could test whether palindromic subsequences are significant important local features learned by DeepHGT.

In general, deep learning models transform and combine local features, which are extracted from input data through convolution layers, to make a prediction. As for one

DNA sequence, any continuous subsequence can be treated as a local feature. However, not all local features are of equal importance in determining the prediction result. Therefore, we define the HGT-Index (HI) to measure the contribution of local features to the prediction value of the sequence. For sequence S, we record its prediction value made by DeepHGT as l. Then for any local feature or local subsequence f_i, we set the output of this local subsequence in the first convolutional layer as 0 and record the corresponding prediction value of the sequence as l_i. So the HGT-Index HI_i for f_i is defined as follows,

$$HI_i = |l - l_i| \tag{1}$$

By setting the output of the local subsequence in the first convolutional layer as 0, DeepHGT could not learn any feature information from this region. The local subsequence could not contribute to the final prediction. It also does not have any coincide with other important sequence features learned by DeepHGT. So $|l_0 - l_i|$ measures the contribution of the local subsequence f_i to the prediction.

From each sample s_i in S_{palin}, we randomly select one palindromic subsequence f_i and compute its HGT-Index HI_i^{palin}, the length of f_i is at least 10 bp. As a comparison, we randomly select a subsequence f_i^0 with equal length no matter it is palindromic. The HGT-Index of f_i^0 is HI_i^0. This generates two sets of HGT-Index $HI^{palin}=\left\{HI_1^{palin}, ..., HI_{125}^{palin}\right\}$ and $HI^0=\left\{HI_1^0, ..., HI_{125}^0\right\}$. The null hypothesis is that HI^{palin} and HI^0 have identical average values, which denotes that palindromic subsequences and random subsequences are consistent with each other. We calculate the T-test for the means of HI^{palin} and HI^0. The T-test result is t-statistic=2.65864, P-value=0.00835, which rejects the null hypothesis. Therefore, palindromic subsequences are significantly important local features learned by DeepHGT to make the prediction.

We collect 6 palindromic sequences from REP sequences found in [49]. In order to test whether the 6 palindromic sequences contribute significantly to the prediction of DeepHGT, for a palindromic sequence s_{palin}, we randomly select a sequence S from our test data set and record its prediction value l, then we randomly select a subsequence s of S and replace it with s_{palin}. Now we have a modified sequence S_{palin} containing s_{palin}. We feed S_{palin} into DeepHGT and record prediction value l_{palin}. The HGT-Index of s_{palin} is $\left|l - l_{palin}\right|$. As a comparison, we generate a randomized DNA sequence s_{null}. s, s_{palin}, and s_{null} have equal length. We replace s with s_{null} to get another modified sequence S_{null}. The prediction value of S_{null} is l_{null}. So the HGT-Index of s_{null} is $|l - l_{null}|$. We repeat these operations 5000 times and get two sets of HGT-Index $HI_{palin}=\{\left|l^1 - l_{palin}^1\right|, ..., \left|l^{5000} - l_{palin}^{5000}\right|\}$ and $HI_{null}=\{\left|l^1 - l_{null}^1\right|, ..., \left|l^{5000} - l_{null}^{5000}\right|\}$. The null hypothesis is that HI_{palin} and HI_{null} have identical average values, which means that there is no difference between the palindromic sequence and a random sequence in affecting the prediction of DeepHGT. Table 3 illustrates the statistical tests of 6 palindromic sequences found in Insertion Sequence elements. As we can see the three palindromic sequences found in ISPa11, ISRm22, ISPpu9, and ISRm19 significantly contribute to the prediction of DeepHGT.

Evaluation of DeepHGT in an independent set of metagenomic samples

To further evaluate the generalization of DeepHGT, the set of 689,312 sequences obtained from 147 metagenomic samples [39] is used as an independent test. Figure 5a and b compare ROC and Precision-Recall curves of DeepHGT and the other five models. DeepHGT

Table 3 Statistical test of 6 palindromic sequences in Insertion Sequence elements

Insertion Sequence elements	palindromic sequence	*p*-value
ISPsy8	TGCCGACGCAGAGCGTCGCA	0.4304
ISPsy8	GGACGCGGAGCGTCC	0.3625
ISPa11	GGCGATCGCGCGATCGCC	$1.0934e^{-10}$
ISPpu9	GCGGGCTAACCCGC	$5.9209e^{-14}$
ISRm22	CCTTCCCCCGCGCGGGGGAAGG	$8.0776e^{-7}$
ISRm19	ACCTTTCCCCGAGCGGGCGAAG	0.0068

has achieved the highest AUC value 0.8448 and AP value 0.8743, which are a little lower than previous test results. Figure 5c and d compare the performance of DeepHGT with four machine learning models implemented in PyFeat. PyFeat_GB has achieved AUC value 0.686 and AP value 0.738 which are worse than DeepHGT. Table 4 compares the accuracy of DeepHGT and other methods. DeepHGT achieved the highest accuracy score of 0.762. As illustrated in Appendix Table 9, all pairwise Delong test results have a *p*-value of less than 0.05. So all models have significantly different performance. These experimental results demonstrate that DeepHGT has learned general sequence patterns that are shared by various HGT insertion sites on reference sequences. So DeepHGT could still achieve better performance than other models in this independent dataset. DeepHGT is a powerful model to accurately recognize HGT insertion sites.

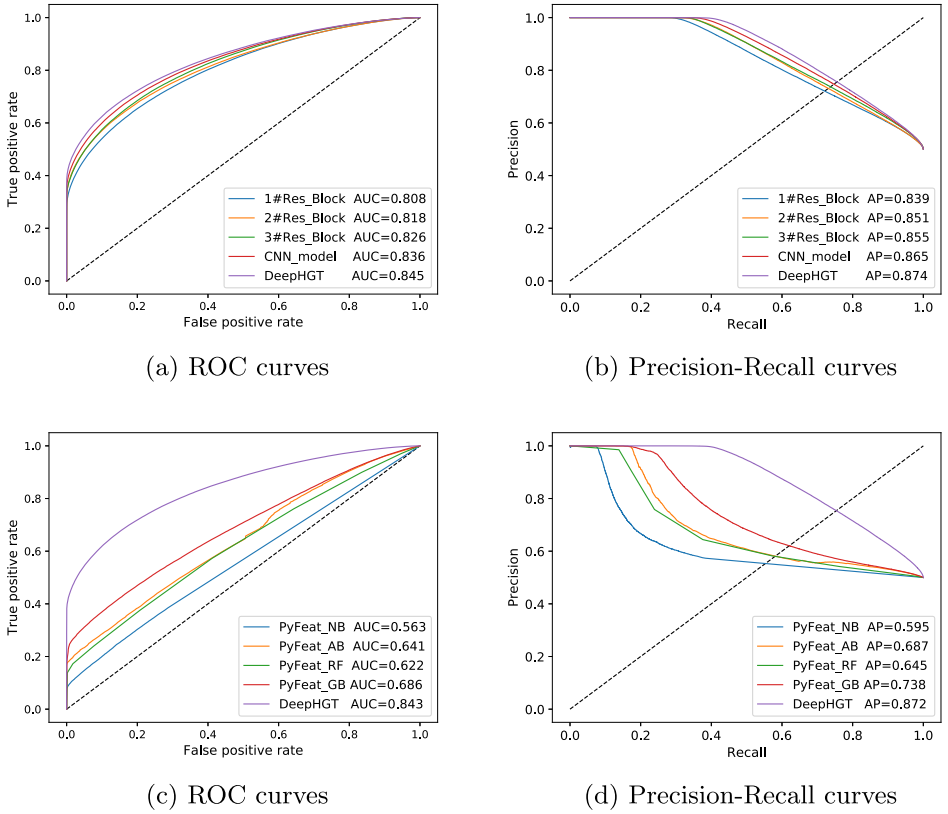

Fig. 5 **a** and **b** compare ROC and Precision-Recall curves of DeepHGT and the other five deep learning models. DeepHGT has achieved the highest AUC value 0.8448 and AP value 0.8743. **c** and **d** compare the performance of DeepHGT and four machine learning models implemented in PyFeat

Table 4 Comparison of accuracy of DeepHGT and other methods for an independent set of Metagenomic samples

1#Res_Block	2#Res_Block	3#Res_Block	CNN_model	DeepHGT
0.725	0.738	0.742	0.752	0.762
PyFeat_rf	PyFeat_ab	PyFeat_gb	PyFeat_nb	DeepHGT
0.580	0.588	0.604	0.500	0.762

Some applications of DeepHGT

Likelihood of bacteria genomes harboring HGT insertions sites

Compared to LEMON, DeepHGT does not need next-generation sequenced (NGS) data as input. DeepHGT could recognize HGT insertion sites on raw DNA sequences according to sequence features. For reference bacteria genomes in NCBI, we could utilize DeepHGT to calculate their likelihood of harboring HGT insertions sites. For each reference, we use a 100 bp slide window to extract subsequences. The stride length is 50 bp. The subsequences are then feed into DeepHGT to get prediction values $P = \{P_i, i = 1, ..., n\}$. The likelihood of the reference is the mean value of P. Table 5 shows some references with high likelihood. These bacteria maybe more easily to receive adaptive advantages through HGT and are worthy of further research.

Find bacterial genes enriched with potential HGT insertion sites

To find genes enriched with potential HGT insertion sites, we collect bacterial genes available from NCBI. For each bacterial gene, we use a 100 bp slide window on the gene region to extract subsequences. The stride length is 10 bp. The subsequences are then feed into DeepHGT to get prediction values $G = \{G_i, i = 1, ..., n\}$. The likelihood of the gene is $\bar{G} = \frac{1}{n}\sum_{i=1}^{n} G_i$. If $\bar{G} > 0.5$, we regard the gene is enriched with potential HGT insertion sites. Finally, we collect 1,404 genes. We then perform Gene Ontology (GO) analysis for these genes. Table 6 shows some biologic processes associated with the most number of genes enriched with potential HGT insertion sites. As we can see 48 genes are associated with translation, whose efficiency is closely related to HGT [57]. Besides, 5 genes are associated with DNA integration, which is the integration mechanism of transferred genes.

Find potential hotspot of HGT insertion sites

By utilizing DeepHGT to scan reference bacteria genomes, we could find regions enriched with potential HGT insertion sites. For a reference sequence, we apply DeepHGT to calculate the distribution of potential HGT insertion sites over it. For each nucleotide of the reference, we extract a 100 bp subsequence, which has the nucleotide in the middle, as the input of DeepHGT. If the prediction probability is larger than 0.5, the nucleotide position is treated as one potential HGT insertion site. Then we slide a window over the reference

Table 5 Likelihood of harboring HGT insertions sites for reference bacteria genomes in NCBI

Genus	Accession	Likelihood	References
Streptomyces	NZ_JOJH01000630.1	0.535	[53, 54]
	NZ_LMFT01000033.1	0.997	
	NZ_LYOT01000881.1	0.999	
Mycobacterium	NZ_MVHE01000390.1	0.871	[55, 56]
	NZ_MVIC01000125.1	0.984	
	NZ_LZSE01000001.1	0.999	

Table 6 Biologic processes associated with the most number of genes enriched with HGT insertion sites

GO ID	Biologic Process	Gene count
GO:0006412	translation	48
GO:0005975	carbohydrate metabolic process	15
GO:0006355	regulation of transcription, DNA-templated	12
GO:0032259	methylation	10
GO:0009116	nucleoside metabolic process	7
GO:0045454	cell redox homeostasis	6
GO:0015074	DNA integration	5

sequence and utilize DeepHGT to get the number of potential HGT insertion sites in each window. The window size is $l = 100$ and the number of potential HGT insertion sites in each window is n, then we use $r = \frac{n}{l}$ to measure the rate of HGT insertion sites in one sliding window. In our experiment, we set $r > 0.2$ to filter out windows with low likelihood.

Figure 6 illustrates the distribution of potential HGT insertion sites predicted by DeepHGT over a reference sequence NZ_LZST01000177.1, which belongs to species *Mycolicibacterium monacense*. As we can see, regions with a high rate of potential HGT insertion sites are randomly distributed across the reference sequence. Some regions are close together. These regions are potential hot spots of HGT insertion sites and deserve more research to explore their relationship with HGT.

Conclusion

In this paper, we propose a deep residual model named DeepHGT to predict HGT insertion sites on reference sequences. By utilizing LEMON, which is based on the traditional alignment technology to detect HGT breakpoints, we obtained two independent sets of sequence segments to train and test DeepHGT. On these two sets, DeepHGT outperforms PyFeat. Since DeepHGT recognizes HGT insertion sites on reference sequence according to sequence patterns, DeepHGT is not affected by sequencing coverage. So DeepHGT is a reliable model to recognize the HGT insertion site. It could help us detect potential HGT sites for further analysis. As we collect more HGT insertion sites and use them to train DeepHGT, it could learn more and accurate general sequence features about HGT insertion sites.

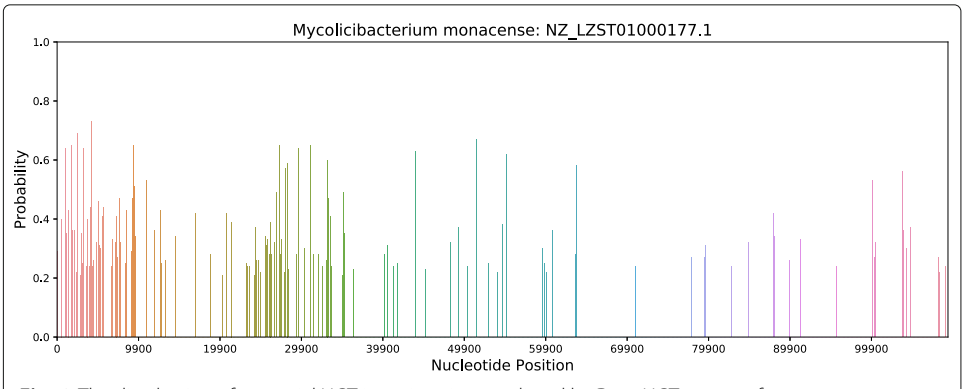

Fig. 6 The distribution of potential HGT insertion sites predicted by DeepHGT over a reference sequence NZ_LZST01000177.1, which belongs to species *Mycolicibacterium monacense*

Discussion

DeepHGT is the first deep learning model which could recognize HGT sites directly on bacterial genomes. DeepHGT is a very complicated model since it contains 2,119,297 trainable parameters. So, to make DeepHGT achieve powerful performance, we construct a very large data set to train DeepHGT. In our experiments, the main reason for DeepHGT achieving better AUC and AP values than other machine learning methods implemented in Pyfeat is that DeepHGT could learn more discriminant sequence features than the ones defined in Pyfeat. These features learned by DeepHGT should be treated as data-driven features. Furthermore, compared to LEMON, the main advantage of DeepHGT is that it need not paired-end DNA sequencing reads as input. So, by running DeepHGT on bacteria genomes and their coding genes available from NCBI, we could calculate the likelihood of bacteria genomes harboring HGT insertions sites and find bacterial genes enriched with potential HGT insertion sites. These preliminary results help us further research the mechanism, function, and benefit of HGT. This is also our future work.

Methods

Dataset

We collect bacterial reference sequences from the National Center for Biotechnology Information (NCBI) to construct a reference sequence set. It contains 109,419 bacterial reference sequences, which belong to 16,093 bacterial species. We index all references together to generate the Burrows-Wheeler Transform (BWT) indexes. LEMON is based on the traditional alignment method which takes shotgun metagenomic reads and the reference sequence set as inputs to detect and label HGT breakpoints. Based on the detected HGT breakpoints we collect 100 bp DNA sequences at HGT insertion sites on bacterial reference sequences. Each sequence has one HGT insertion site in the middle.

As illustrated in Fig. 7a, S is a sequenced strain, which consists of a harbor sequence R_1 and one horizontal transferred segment T from reference sequence R_2. B is a HGT breakpoint on strain S, which is supported by one paired-end read and two split reads. Two split reads a_1 and b_2 are clipped on the B, which means that one portion a_1' of a_1 is aligned to left side of B_1, and the other portion a_1'' of a_1 is aligned to right side of B_2 as illustrated in Fig. 1b. Reads c_1 and c_2, which belong to one paired-end read, are on

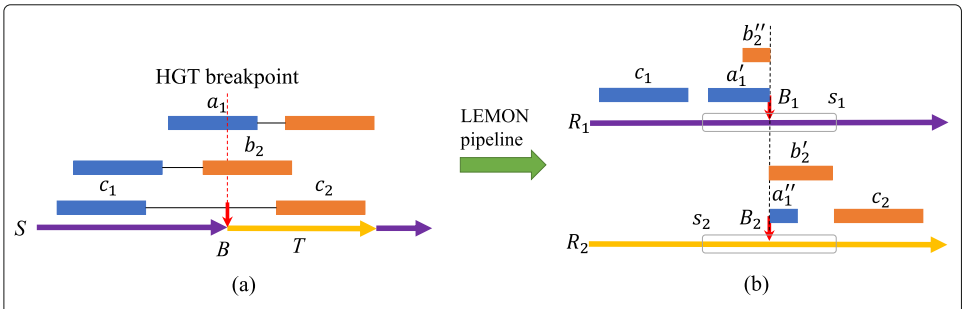

Fig. 7 S is a sequenced strain, which contains one transferred segment T from reference sequence R_2. B is a HGT breakpoint, which is supported by one paired-end read and two split reads a_1 and b_2. Through LEMON pipeline, we detect two HGT insertion sites B_1 and B_2 on references R_1 and R_2 respectively. For each split read, one portion of it is aligned to the left side of B_1 and the other portion is aligned to the right side of B_2. We then extract two 100 bp sequences s_1 and s_2 as input data of DeepHGT. B_1 and B_2 are in the middle of s_1 and s_2 respectively

the two sides of B. Since HGT denotes the insertion of foreign gene, the sequences on two sides of B are belong to two different reference sequences, so B is corresponding to two breakpoint positions B_1 and B_2 on references R_1 and R_2 respectively as shown in Fig. 7b. We define B_1 and B_2 as HGT insertion sites. We then extract two 100 bp sequences s_1 and s_2, which have B_1 and B_2 in the middle respectively, as input data of DeepHGT. LEMON pipeline in Fig. 7 denotes the process of detecting HGT insertion sites from raw paired-end reads [34]. Firstly, we apply BWA software for mapping raw reads against the reference genomes. Then we utilize samtools to extract split reads and unique mapping reads. The mapping quality of unique reads is 20. The unique mapping reads and split reads are inputs of LEMON. LEMON utilizes paired-end reads to get candidate regions for HGT insertion sites and split reads to infer the precise insertion site positions on reference sequences. Each HGT insertion site is supported by at least one pair-end read and one split read. For example, the insertion sites B_1 and B_2 in Fig. 7b are supported by two split reads a_1, b_2 and one paired-end read $c_1 - c_2$. Sequences s_1 and s_2 are treated as positive samples. To get negative samples, we randomly extract 100 bp DNA sequences from regions that are at least 10,000 bp away from the nearest HGT insertion sites on reference sequences. So there is no overlap between positive and negative samples.

Data augmentation

Data augmentation is an efficient technique to improve modern deep learning performance on image classification. Through a series of operations on images, the technique will expand the training set, which can aid deep learning models in learning robust features and achieve better performance. Therefore, to make DeepHGT fight overfitting and get better generalization in DNA sequence learning, we have tried two data augmentation methods as illustrated in Fig. 8. The first method in Fig. 8.a is shifting sampling positions near HGT insertion sites within a small region ($\pm 5bp$) to get a vast amount of augmented training samples. Since the region is very small, the augmented samples contain most sequence information of HGT insertion sites. The second method in Fig. 8.b is to randomly change a small number of nucleotides for each training sample. The maximum number of nucleotides that are randomly changed is 10. Since most nucleotides of one sequence are retained, this method will not change the sequence pattern but increase the diversity of training samples, which helps DeepHGT to learn robust sequence features. The two augmentation methods are applied to all positive and negative training samples to generate augmented positive and negative training samples, which are used as the input of DeepHGT. In our experiments, these two techniques could improve around 0.01∼0.02 AUC value.

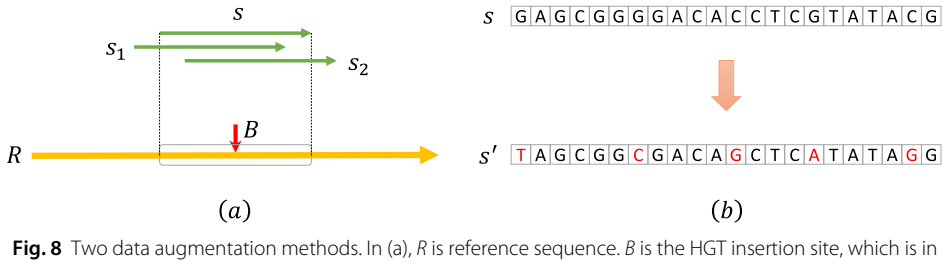

Fig. 8 Two data augmentation methods. In (a), R is reference sequence. B is the HGT insertion site, which is in the middle of training sequence s. s_1 and s_2 are augmented training samples, which are sampled from near the HGT insertion site B within a small region. In (b), through randomly changing nucleotides in s, we get the augmented training sample s', which retains most nucleotides of s

Appendix

Table 7 The two percentage distributions of the top 10 most abundant species to which sequences in the positive training dataset and the independent positive test dataset belong

The positive training dataset	Percentage (%)	The independent positive test dataset	Percentage (%)
Microbacterium esteraromaticum	13.13	Faecalibacterium prausnitzii A2-165	7.69
Mycolicibacterium monacense	7.36	Microbacterium esteraromaticum	4.84
Mycobacterium sp. 852002-51961_SCH5331710	3.08	Prevotella copri DSM 18205	4.38
Faecalibacterium prausnitzii A2-165	2.39	Mycobacterium sp. 852002-51961_SCH5331710	3.5
Collinsella aerofaciens ATCC 25986	1.97	Mycolicibacterium monacense	3.04
Collinsella sp. 4_8_47FAA	1.94	Bacteroides stercoris ATCC 43183	2.53
Gemmiger formicilis	1.69	Roseburia faecis	2.33
Collinsella sp. TF06-26	1.64	Roseburia intestinalis L1-82	2.01
Bifidobacterium longum	1.55	Gemmiger formicilis	1.56
Bacteroides caccae	1.50	Acinetobacter sp. AR2-3	1.48

Table 8 Pairewise Delong test on AUCs of DeepHGT and other methods for test data set

	1#Res_Block	2#Res_Block	3#Res_Block	CNN_model	DeepHGT
1#Res_Block	–	$< 2.2e^{-16}$	$< 2.2e^{-16}$	$< 2.2e^{-16}$	$< 2.2e^{-16}$
2#Res_Block	–	–	0.000239	$< 2.2e^{-16}$	$< 2.2e^{-16}$
3#Res_Block	–	–	–	$< 2.2e^{-16}$	$< 2.2e^{-16}$
CNN_model	–	–	–	–	$< 2.2e^{-16}$
DeepHGT	–	–	–	–	–
	PyFeat_AB	PyFeat_GB	PyFeat_NB	PyFeat_RF	DeepHGT
PyFeat_AB	–	$< 2.2e^{-16}$	$< 2.2e^{-16}$	$< 2.2e^{-16}$	$< 2.2e^{-16}$
PyFeat_GB	–	–	$< 2.2e^{-16}$	$< 2.2e^{-16}$	$< 2.2e^{-16}$
PyFeat_NB	–	–	–	$< 2.2e^{-16}$	$< 2.2e^{-16}$
PyFeat_RF	–	–	–	–	$< 2.2e^{-16}$
DeepHGT	–	–	–	–	–

Table 9 Pairewise Delong test on AUCs of DeepHGT and other methods for an independent set of Metagenomic samples

	1#Res_Block	2#Res_Block	3#Res_Block	CNN_model	DeepHGT
1#Res_Block	–	$< 2.2e^{-16}$	$< 2.2e^{-16}$	$< 2.2e^{-16}$	$< 2.2e^{-16}$
2#Res_Block	–	–	$< 2.2e^{-16}$	$< 2.2e^{-16}$	$< 2.2e^{-16}$
3#Res_Block	–	–	–	$< 2.2e^{-16}$	$< 2.2e^{-16}$
CNN_model	–	–	–	–	$< 2.2e^{-16}$
DeepHGT	–	–	–	–	–
	PyFeat_AB	PyFeat_GB	PyFeat_NB	PyFeat_RF	DeepHGT
PyFeat_AB	–	$< 2.2e^{-16}$	$< 2.2e^{-16}$	$< 2.2e^{-16}$	$< 2.2e^{-16}$
PyFeat_GB	–	–	$< 2.2e^{-16}$	$< 2.2e^{-16}$	$< 2.2e^{-16}$
PyFeat_NB	–	–	–	$< 2.2e^{-16}$	$< 2.2e^{-16}$
PyFeat_RF	–	–	–	–	$< 2.2e^{-16}$
DeepHGT	–	–	–	–	–

Abbreviations
AB: Adaboost Classifier; AUC: Area under the Curve of ROC; AP: Average-Precision; BWA: Burrows-Wheeler Alignment; BWT: Burrows-Wheeler Transform; BN: Batch Normalization; CNN: Convolutional Neural Network; DBSCAN: Density-Based Spatial Clustering of Applications with Noise; GB: Gradient Boosting; GO: Gene Ontology; HI: HGT-Index; HGT: Horizontal Gene Transfer; ICEs: integrative conjugative elements; MGE: Mobile genetic elements; NB: Naive Bayes; NCBI: National Center for Biotechnology Information; NGS: Next-Generation Sequencing; REP: Repetitive Extragenic Palindromic; RF: Random Forest; ROC: Receiver Operating Characteristic; SGD: Stochastic Gradient Descent

Acknowledgements
Not applicable.

About this supplement
This article has been published as part of BMC Genomics Volume 21 Supplement 11 2020: Bioinformatics methods for biomedical data science. The full contents of the supplement are available at https://bmcgenomics.biomedcentral.com/articles/supplements/volume-21-supplement-11.

Authors' contributions
SL conceived the project. CL designed the algorithm and implemented the algorithm. CL performed the analyses and evaluated the results, CL and JXC wrote the manuscript. All authors reviewed the manuscript. All authors read and approved the final manuscript.

Funding
The work is supported by City University of Hong Kong(Project 7005215). Publication costs are funded by City University of Hong Kong(Project 7005215). The funding body did not play any role in the design of the study and collection, analysis, interpretation of data, and manuscript writing.

Availability of data and materials
262 Metagenomic samples were deposited to Sequence Read Archive (BioProject: PRJNA475246). 147 Metagenomic samples were deposited to Sequence Read Archive (BioProject:PRJNA389280). Codes, training and test samples are freely available at https://github.com/lichen2018/DeepHGT.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 11 November 2020 Accepted: 28 November 2020 Published: 29 December 2020

References
1. Ochman H, Lawrence JG, Groisman EA. Lateral gene transfer and the nature of bacterial innovation. Nature. 2000;405(6784):299–304. https://doi.org/10.1038/35012500.
2. Soucy SM, Huang J, Gogarten JP. Horizontal gene transfer: building the web of life. Nat Rev Genet. 2015;16(8):472–82. https://doi.org/10.1038/nrg3962.
3. Li C, Chen J, Li SC. Understanding horizontal gene transfer network in human gut microbiota. Gut Pathogens. 2020;12(1):. https://doi.org/10.1186/s13099-020-00370-9.
4. Gyles C, Boerlin P. Horizontally transferred genetic elements and their role in pathogenesis of bacterial disease. Vet Pathol. 2013;51(2):328–40. https://doi.org/10.1177/0300985813511131.
5. Warnes SL, Highmore CJ, Keevil CW. Horizontal transfer of antibiotic resistance genes on abiotic touch surfaces: Implications for public health. mBio. 2012;3(6):. https://doi.org/10.1128/mbio.00489-12.
6. Gogarten JP, Doolittle WF, Lawrence JG. Prokaryotic evolution in light of gene transfer. Mol Biol Evol. 2002;19(12):2226–38. https://doi.org/10.1093/oxfordjournals.molbev.a004046.
7. Andam CP, Gogarten JP. Biased gene transfer in microbial evolution. Nat Rev Microbiol. 2011;9(7):543–55. https://doi.org/10.1038/nrmicro2593.
8. Polz MF, Alm EJ, Hanage WP. Horizontal gene transfer and the evolution of bacterial and archaeal population structure. Trends Genet. 2013;29(3):170–5. https://doi.org/10.1016/j.tig.2012.12.006.
9. Heinemann JA, Sprague GF. Bacterial conjugative plasmids mobilize DNA transfer between bacteria and yeast. Nature. 1989;340(6230):205–9. https://doi.org/10.1038/340205a0.
10. Johnston C, Martin B, Fichant G, Polard P, Claverys J-P. Bacterial transformation: distribution, shared mechanisms and divergent control. Nat Rev Microbiol. 2014;12(3):181–96. https://doi.org/10.1038/nrmicro3199.
11. Watson BNJ, Staals RHJ, Fineran PC. CRISPR-cas-mediated phage resistance enhances horizontal gene transfer by transduction. mBio. 2018;9(1):e02406–17. https://doi.org/10.1128/mbio.02406-17.
12. Rubio-Cosials A, Schulz EC, Lambertsen L, Smyshlyaev G, Rojas-Cordova C, Forslund K, Karaca E, Bebel A, Bork P, Barabas O. Transposase-DNA complex structures reveal mechanisms for conjugative transposition of antibiotic resistance. Cell. 2018;173(1):208–2020. https://doi.org/10.1016/j.cell.2018.02.032.

13. Chen J, Quiles-Puchalt N, Chiang YN, Bacigalupe R, Fillol-Salom A, Chee MSJ, Fitzgerald JR, Penadés JR. Genome hypermobility by lateral transduction. Science. 2018;362(6411):207–12. https://doi.org/10.1126/science.aat5867.
14. Kominek J, Doering DT, Opulente DA, Shen X-X, Zhou X, DeVirgilio J, Hulfachor AB, Groenewald M, Mcgee MA, Karlen SD, Kurtzman CP, Rokas A, Hittinger CT. Eukaryotic acquisition of a bacterial operon. Cell. 2019;176(6): 1356–6610. https://doi.org/10.1016/j.cell.2019.01.034.
15. Frost LS, Leplae R, Summers AO, Toussaint A. Mobile genetic elements: the agents of open source evolution. Nat Rev Microbiol. 2005;3(9):722–32. https://doi.org/10.1038/nrmicro1235.
16. Dagan T, Artzy-Randrup Y, Martin W. Modular networks and cumulative impact of lateral transfer in prokaryote genome evolution. Proc Natl Acad Sci. 2008;105(29):10039–44. https://doi.org/10.1073/pnas.0800679105.
17. Husnik F, McCutcheon JP. Functional horizontal gene transfer from bacteria to eukaryotes. Nat Rev Microbiol. 2017;16(2):67–79. https://doi.org/10.1038/nrmicro.2017.137.
18. Wybouw N, Pauchet Y, Heckel DG, Leeuwen TV. Horizontal gene transfer contributes to the evolution of arthropod herbivory. Genome Biol Evol. 2016;8(6):1785–801. https://doi.org/10.1093/gbe/evw119.
19. Wybouw N, Dermauw W, Tirry L, Stevens C, Grbić M, Feyereisen R, Leeuwen TV. A gene horizontally transferred from bacteria protects arthropods from host plant cyanide poisoning. eLife. 2014;3:. https://doi.org/10.7554/elife.02365.
20. Acuna R, Padilla BE, Florez-Ramos CP, Rubio JD, Herrera JC, Benavides P, Lee S-J, Yeats TH, Egan AN, Doyle JJ, Rose JKC. Adaptive horizontal transfer of a bacterial gene to an invasive insect pest of coffee. Proc Natl Acad Sci. 2012;109(11):4197–202. https://doi.org/10.1073/pnas.1121190109.
21. Husnik F, Nikoh N, Koga R, Ross L, Duncan RP, Fujie M, Tanaka M, Satoh N, Bachtrog D, Wilson ACC, von Dohlen CD, Fukatsu T, McCutcheon JP. Horizontal gene transfer from diverse bacteria to an insect genome enables a tripartite nested mealybug symbiosis. Cell. 2013;153(7):1567–78. https://doi.org/10.1016/j.cell.2013.05.040.
22. Luan J-B, Chen W, Hasegawa DK, Simmons AM, Wintermantel WM, Ling K-S, Fei Z, Liu S-S, Douglas AE. Metabolic coevolution in the bacterial symbiosis of whiteflies and related plant sap-feeding insects. Genome Biol Evol. 2015;7(9):2635–47. https://doi.org/10.1093/gbe/evv170.
23. Berg DE, Johnsrud L, McDivitt L, Ramabhadran R, Hirschel BJ. Inverted repeats of tn5 are transposable elements. Proc Natl Acad Sci. 1982;79(8):2632–5. https://doi.org/10.1073/pnas.79.8.2632.
24. Wilde C, Bachellier S, Hofnung M, Clement J-M. Transposition of IS1397 in the family enterobacteriaceae and first characterization of ISKpn1, a new insertion sequence associated with klebsiella pneumoniae palindromic units. J Bacteriol. 2001;183(15):4395–404. https://doi.org/10.1128/jb.183.15.4395-4404.2001.
25. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436–44. https://doi.org/10.1038/nature14539.
26. Ciresan D, Meier U, Schmidhuber J. Multi-column deep neural networks for image classification. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. Providence, Rhode Island: IEEE; 2012. https://doi.org/10.1109/cvpr.2012.6248110.
27. Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1, NIPS'12. USA: Curran Associates Inc.; 2012. p. 1097–105. http://dl.acm.org/citation.cfm?id=2999134.2999257.
28. Hannun A, Case C, Casper J, Catanzaro B, Diamos G, Elsen E, Prenger R, Satheesh S, Sengupta S, Coates A, Y. Ng A. Deepspeech: Scaling up end-to-end speech recognition. Preprint. 2014. https://arXiv.org/1412.5567.
29. Alipanahi B, Delong A, Weirauch MT, Frey BJ. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. Nat Biotechnol. 2015;33(8):831–8. https://doi.org/10.1038/nbt.3300.
30. Weirauch MT, Cote A, Norel R, Annala M, Zhao Y, Riley TR, Saez-Rodriguez J, Cokelaer T, Vedenko A, Talukder S, Bussemaker HJ, Morris QD, Bulyk ML, Stolovitzky G, Hughes TR. Evaluation of methods for modeling transcription factor sequence specificity. Nat Biotechnol. 2013;31(2):126–34. https://doi.org/10.1038/nbt.2486.
31. Zhou J, Troyanskaya OG. Predicting effects of noncoding variants with deep learning–based sequence model. Nat Methods. 2015;12(10):931–4. https://doi.org/10.1038/nmeth.3547.
32. Umarov RK, Solovyev VV. Recognition of prokaryotic and eukaryotic promoters using convolutional deep learning neural networks. PLOS ONE. 2017;12(2):0171410. https://doi.org/10.1371/journal.pone.0171410.
33. Kelley DR, Snoek J, Rinn JL. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Res. 2016;26(7):990–9. https://doi.org/10.1101/gr.200535.115.
34. Li C, Jiang Y, Li S. LEMON: a method to construct the local strains at horizontal gene transfer sites in gut metagenomics. BMC Bioinformatics. 2019;20:(S23). https://doi.org/10.1186/s12859-019-3301-8.
35. Karakoc E, Alkan C, Roak B, Dennis MY, Vives L, Mark K, Rieder MJ, Nickerson DA, Eichler EE. Detection of structural variants and indels within exome data. Nat Methods. 2011;9(2):176–8. https://doi.org/10.1038/nmeth.1810.
36. Ester M, Kriegel H-P, Sander J, Xu X. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96. Portland: AAAI Press; 1996. p. 226–231. http://dl.acm.org/citation.cfm?id=3001460.3001507.
37. Muhammod R, Ahmed S, Farid DM, Shatabda S, Sharma A, Dehzangi A. PyFeat: a python-based effective feature generation tool for DNA, RNA and protein sequences. Bioinforma. 2019;35(19):3831–3. https://doi.org/10.1093/bioinformatics/btz165.
38. Yassour M, Jason E, Hogstrom LJ, Arthur TD, Tripathi S, Siljander H, Selvenius J, Oikarinen S, Hyöty H, Virtanen SM, Ilonen J, Ferretti P, Pasolli E, Tett A, Asnicar F, Segata N, Vlamakis H, Lander ES, Huttenhower C, Knip M, Xavier RJ. Strain-level analysis of mother-to-child bacterial transmission during the first few months of life. Cell Host & Microbe. 2018;24(1):146–544. https://doi.org/10.1016/j.chom.2018.06.007.
39. Schirmer M, Franzosa EA, Lloyd-Price J, McIver LJ, Schwager R, Poon TW, Ananthakrishnan AN, Andrews E, Barron G, Lake K, Prasad M, Sauk J, Stevens B, Wilson RG, Braun J, Denson LA, Kugathasan S, McGovern DPB, Vlamakis H, Xavier RJ, Huttenhower C. Dynamics of metatranscription in the inflammatory bowel disease gut microbiome. Nature Microbiol. 2018;3(3):337–46. https://doi.org/10.1038/s41564-017-0089-z.
40. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas: IEEE; 2016. p. 770–8. https://doi.org/10.1109/cvpr.2016.90.

41. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariateshift. In: Proceedings of the 32nd International Conference on Machine Learning, vol. 37. Lille: PMLR Press; 2015. p. 448–56. http://proceedings.mlr.press/v37/ioffe15.htmlshift.
42. Srivastava N, Hinton GE, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014;15:1929–58.
43. Georgakilas GK, Grioni A, Liakos KG, et al. Multi-branch Convolutional Neural Network for Identification of Small Non-coding RNA genomic loci. Sci Rep. 2020;10:9486. https://doi.org/10.1038/s41598-020-66454-3.
44. Maron ME. Automatic indexing: An experimental inquiry. J ACM. 1961;8(3):404–17. https://doi.org/10.1145/321075.321084.
45. Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci. 1997;55(1):119–39. https://doi.org/10.1006/jcss.1997.1504.
46. Breiman L. Random Forests. Mach Learn. 2001;45(1):5–32. https://doi.org/10.1023/a:1010933404324.
47. Mason L, Baxter J, Bartlett P, Frean M. Boosting algorithms as gradient descent. In: Proceedings of the 12th International Conference on Neural Information Processing Systems, NIPS'99. Cambridge, MA, USA: MIT Press; 1999. p. 512–8. http://dl.acm.org/citation.cfm?id=3009657.3009730.
48. Stern MJ, Ames GF-L, Smith NH, Robinson EC, Higgins CF. Repetitive extragenic palindromic sequences: A major component of the bacterial genome. Cell. 1984;37(3):1015–26. https://doi.org/10.1016/0092-8674(84)90436-7.
49. Tobes R, Pareja E. Bacterial repetitive extragenic palindromic sequences are dna targets for insertion sequence elements. BMC Genomics. 2006;7(1):62. https://doi.org/10.1186/1471-2164-7-62.
50. Wilde C. Transposases are responsible for the target specificity of IS1397 and ISKpn1 for two different types of palindromic units (PUs). Nucleic Acids Res. 2003;31(15):4345–53. https://doi.org/10.1093/nar/gkg494.
51. Darmon E, Leach DRF. Bacterial genome instability. Microbiol Mol Biol Rev. 2014;78(1):1–39. https://doi.org/10.1128/mmbr.00035-13.
52. Youden WJ. Index for rating diagnostic tests. Cancer. 1950;3(1):32–5. https://doi.org/10.1002/1097-0142(1950)3:1<32::aid-cncr2820030106>3.0.co;2-3.
53. Doroghazi JR, Buckley DH. Widespread homologous recombination within and between streptomyces species. ISME J. 2010;4(9):1136–43. https://doi.org/10.1038/ismej.2010.45.
54. Tidjani A-R, Lorenzi J-N, Toussaint M, van Dijk E, Naquin D, Lespinet O, Bontemps C, Leblond P. Massive gene flux drives genome diversity between sympatric streptomyces conspecifics. mBio. 2019;10(5):. https://doi.org/10.1128/mbio.01533-19.
55. Panda A, Drancourt M, Tuller T, Pontarotti P. Genome-wide analysis of horizontally acquired genes in the genus mycobacterium. Sci Rep. 2018;8(1):. https://doi.org/10.1038/s41598-018-33261-w.
56. Boritsch EC, Khanna V, Pawlik A, Honoré N, Navas VH, Ma L, Bouchier C, Seemann T, Supply P, Stinear TP, Brosch R. Key experimental evidence of chromosomal DNA transfer among selected tuberculosis-causing mycobacteria. Proc Natl Acad Sci. 2016;113(35):9876–81. https://doi.org/10.1073/pnas.1604921113.
57. Tuller T, Girshovich Y, Sella Y, Kreimer A, Freilich S, Kupiec M, Gophna U, Ruppin E. Association between translation efficiency and horizontal gene transfer within microbial communities. Nucleic Acids Res. 2011;39(11):4743–55. https://doi.org/10.1093/nar/gkr054.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ready to submit your research? Choose BMC and benefit from:

- fast, convenient online submission
- thorough peer review by experienced researchers in your field
- rapid publication on acceptance
- support for research data, including large and complex data types
- gold Open Access which fosters wider collaboration and increased citations
- maximum visibility for your research: over 100M website views per year

At BMC, research is always in progress.

Learn more biomedcentral.com/submissions