Dear Professor,

Enclosed please find our substantially revised manuscript “*Lncident: A Tool for Rapid Identification of Long Non-Coding RNAs Utilizing Sequence Intrinsic Composition and Open Reading Frame Information*”. In this revised manuscript, we have carefully addressed all the concerns by the two reviewers. We greatly appreciate the Referee’s comments on the original draft of the paper and we hope that you and your reviewers find the revised version acceptable for publication in *International Journal of Genomics*. The following is our point-by-point response to each of the criticisms by the two reviewers. I would like to take this opportunity to thank you for handling the review of our paper.

Sincerely yours,

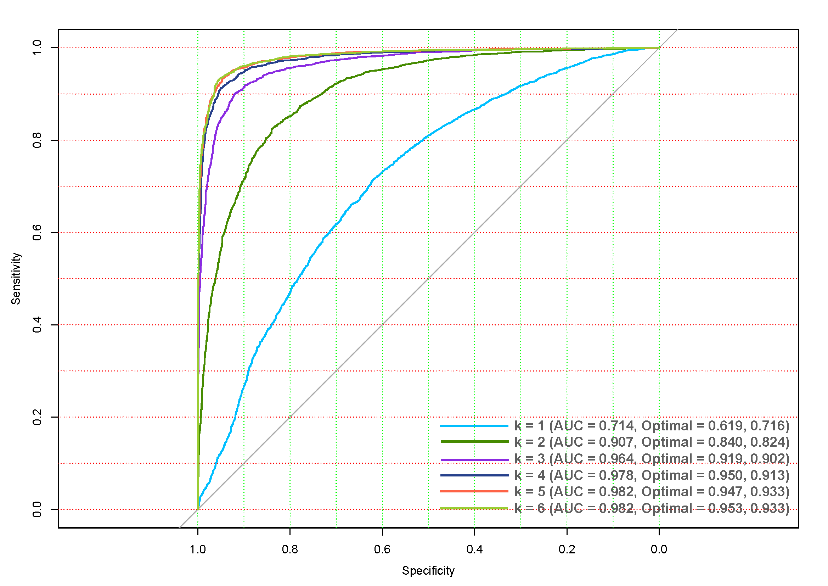
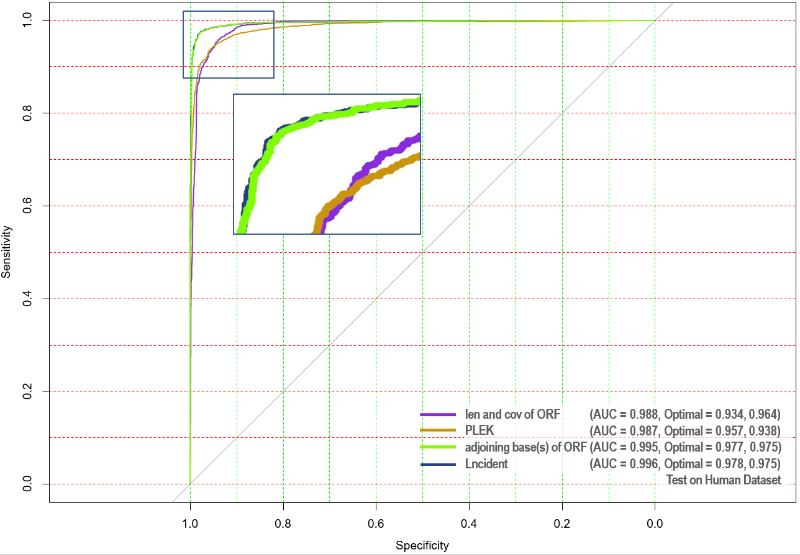
Ying Li, Ph.D.

**Review #1**

**General comments**: Han et. al. provided a novel tool, Lncident, to discriminate lncRNAs from coding transcripts in various vertebrate and invertebrate species without the help of sequence alignment. Lncident encodes lncRNA sequences using *k*-mer frequencies and the length and the coverage of open reading frames, then trains several SVM binary classifiers that are specific to model organisms to achieve better prediction accuracy. The authors adopted 10-fold cross validation and several measures including F1-score and AUC to compare their results with other existing tools. They claim that Lncident finds a better balance between sensitivity and specificity, and that it significantly outperforms other methods in *S. cerevisiae*. In addition, Lncident is able to complete the prediction process in a few minutes, suggesting unprecedented potential in the large-scale genome-wide analysis. Lncident was implemented in both a web server interface and as an R package. Though several interesting ideas behind the method, a few problems should be resolved before further consideration.

**Response:** Thank you for your comments. In this revised version, we mainly added several experiments to further display the performance of Lncident. Compared Lncident with other tools on a new human dataset, Lncident shows a satisfying result. For invertebrate species, we retrained the CPAT [1] and PLEK [2] using the same training dataset as Lncident and made new comparisons. In addition, we conducted RFE (Recursive Feature Elimination) on different feature groups to test whether ORF features are redundant. The important score for each feature is computed based on RFE algorithm and all features were sorted based on the importance scores. We listed the top 10 features according to the 10 highest scores, where the ORF features have the highest importance scores. Finally, therunning time of Lncident and other tools was computed and compared to evaluate the efficiency of Lncident.

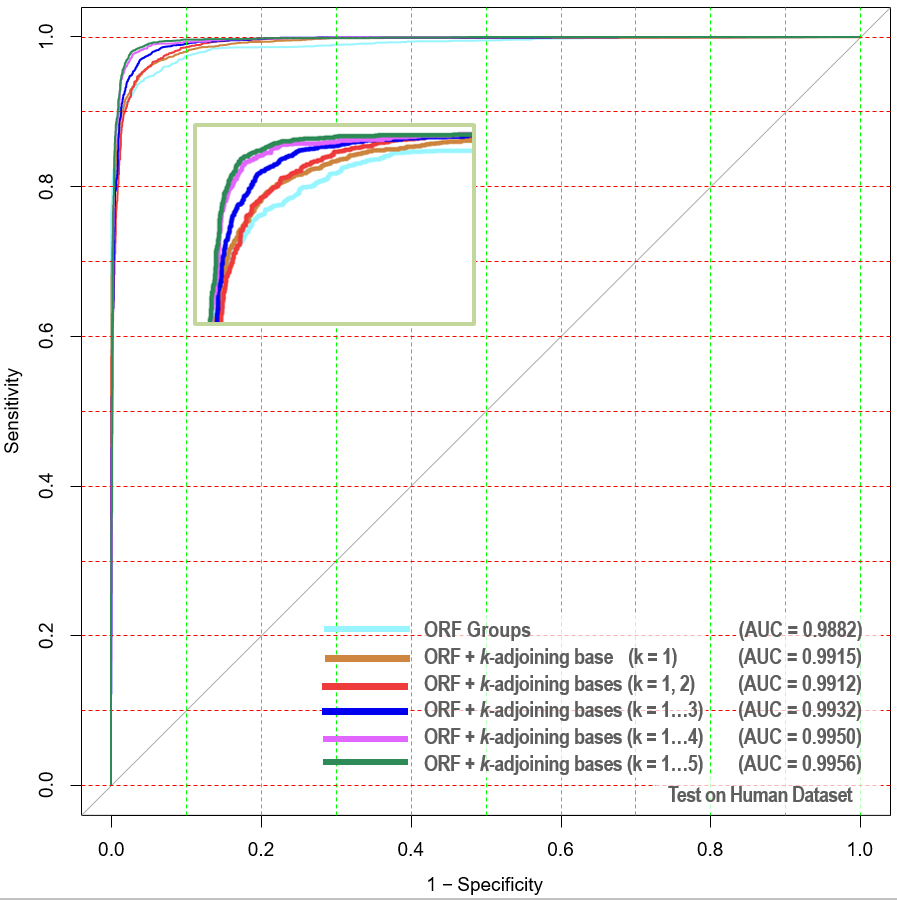
**Concern 1:** First, please rank each feature using a well-known feature selection method such as MRMR (maximum relevancy and minimum redundancy) or RFE (recursive feature elimination) and report the top ones to facilitate better biological understanding. Please report if the performance can be further improved if only a subset of features is included based on 10-fold cross validation scheme. Based on Figure 3, it appears that the length and the coverage of open reading frames are somehow redundant.

**Response:** Thank you for your comments.

**Figure S2**

**Figure 3**

1. Here are the figures to evaluate the performances of different feature groups: The left one is the Figure S2 used to determine the optimum *k*. The worst performance is *k* = 1. The best result is *k* = 6. The right one is the Figure 3 used to display that the best performance occurs when all these feature groups combined together. The ORF feature group only contain two features: the length and coverage of the maximum ORF. However, these two features achieve better result (AUC = 0.988, sensitivity = 0.964, specificity = 0.934, in Figure 3) than the feature group of *k*-mer (*k* = 6) which contains 4,096 features (AUC = 0.982, sensitivity = 0.953, specificity = 0.933, in Figure S2).

(2). The Figure S2 shows different performances under different *k*. We conducted RFE to display the performances when different *k* are combined together. For different *k*, considering that only the whole *k*-mer feature group can present the complete nucleotide composition, we only utilized RFE on feature groups. We can evaluate the importance of ORF featutes: AUC is 0.714 when only using *k*-adjoining base (*k* = 1) features (Figure S2); AUC is increased to 0.992 when combining ORF with *k*-adjoining base (*k* = 1) features (see left figure). Quite the contrary, the performance improves marginally with *k* increasing. Since ORF + *k*-adjoining bases (*k* = 1, 2,…,5) presents the best result, we built our model with all the features.

The length and coverage of the maximum ORF have been used as features in many lncRNA identification tools including CPC [3], CPAT [1], lncRNA-ID [4], lncRNApred [5] and lncRNA-MFDL [6]. lncRNA-ID, lncRNApred and lncRNA-MFDL do not provide source code. Therefore we did not compare Lncident with lncRNA-ID, lncRNApred and lncRNA-MFDL in this manuscript.

(3). The top 10 features with 10 highest importance scores are listed in Table 1 shows t. The scores were calculated based on SVM-RFE algorithm.

|  |  |  |  |
| --- | --- | --- | --- |
| Table 1. The top 10 features and their importance scores | | | |
| Feature | **Importance Score** | **Feature** | **Importance Score** |
| Length of max ORF | 100.00 | CGG | 34.02 |
| Coverage of max ORF | 74.21 | GCGGC | 32.56 |
| CG | 49.72 | CGC | 31.53 |
| CGGCG | 41.22 | CCG | 31.43 |
| GCG | 36.37 | CGGC | 30.48 |

In Table 1, the scores were normalized according to the highest score, and the score of the most important feature was scaled to 100. The length and coverage of the maximum ORF are the most important features. We noticed that, except ORF feature group, the other important features are related to nucleotides “C” and “G”. In addition, the ranks of *k*-adjoining base (*k* = 1) features are listed in Table 2. For 1,366 features of Lncident, all the *k*-adjoining base (*k* = 1) features are in the top 100.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table 2. Rank of A, C, G, T | | | | |
| Features | A | C | G | T | |
| Rank | 86 | 73 | 38 | 50 | |

**Concern 2:** Second, since the improvement is minor, a more extensive comparison is necessary. Adding more measures such as Matthews’ correlation coefficient (MCC) or considering more species and methods may improve the strength of the manuscript. If other methods can be trained on invertebrate or microorganism sequences, please compare their prediction results with Lncident for *C. elegans* and *S. cerevisiae* as well. In addition, please test Lncident’s accuracy on synthetic data by generating hundreds of random sequences and running each method to see if Lncident produces fewer false positives.

**Response:** Thank you for this comment. In this revised manuscript, we made several modifications on experimental evaluation.

1. Improvement of Lncident:

According to the experimental results on GENCODE [7] human dataset for lncRNA identification, the performance of Lncident is better than CPAT, CNCI and PLEK, only slightly inferior to CPC (See Table 3). However, as an alignment-based tool, CPC aligns all the sequences against the whole reference database. It is not surprising that CPC can obtain the best result when the sequences are selected from the “transcripts file” or “lncRNA file”of the database.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table 3. The performances on human species of GENCODE database | | | | |
| Tools | **Sensitivity** | **Specificity** | **Accuracy** | **F-measure** |
| CPC (2007) | **0.9885** | **0.9985** | **0.9935** | **0.9934** |
| CPAT (2013) | 0.9501 | 0.9608 | 0.9555 | 0.9552 |
| CNCI (2013) | 0.9734 | 0.9095 | 0.9414 | 0.9433 |
| PLEK (2014) | 0.9407 | 0.9448 | 0.9427 | 0.9426 |
| Lncident | 0.9763 | 0.9742 | 0.9752 | 0.9752 |

Additionally, we also assesed these tools with another human datasets [4] which contains 4,000 lncRNAs and 4,000 mRNAs in test set. These lncRNA sequences were selected from the “genome file” and some of them (1,186 lncRNAs) were not included in the ready made “lncRNA file”. We also re-trained Lncident with 8,300 lncRNAs and 20,000 mRNAs. Considering that CPAT and PLEK can be trained by users, we also evaluates these two tools by training their models with the same training set used in Lncident. The training and testing datasets have no transcripts from the same genes and only the transcripts from every gene are collected in training set. Since CPC and CPAT provided web interface, stand-alone and web server version are both tested. The performances are shown in Table 4.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 4. The performances on a new human dataset | | | | | | |
| Tools | **Sensitivity** | **Specificity** | **Accuracy** | **F-measure** | **MCC** | **Kappa** |
| CPC.local | 0.6613 | **1** | 0.8306 | 0.7961 | 0.7028 | 0.6612 |
| CPC.web | 0.6625 | 0.9992 | 0.8309 | 0.7966 | 0.7028 | 0.6618 |
| CPAT.local | 0.9333 | 0.9802 | 0.9568 | 0.9557 | 0.9145 | 0.9135 |
| CPAT.web | 0.9535 | 0.9673 | 0.9604 | 0.9601 | 0.9208 | 0.9208 |
| CNCI | 0.9702 | 0.9157 | 0.9430 | 0.9445 | 0.8873 | 0.8860 |
| PLEK | **0.9952** | 0.8918 | 0.9435 | 0.9463 | 0.8918 | 0.8870 |
| CPAT.train\* | 0.9160 | 0.9848 | 0.9504 | 0.9486 | 0.9029 | 0.9008 |
| PLEK.train\* | 0.7622 | 0.9507 | 0.8565 | 0.8416 | 0.7260 | 0.7130 |
| Lncident | 0.9535 | 0.9795 | **0.9665** | **0.9661** | **0.9333** | **0.9330** |
| CPC and CPAT were tested on stand-alone version and web server. \*The suffix of “train” meansCPAT and PLEK with the new-trained model. | | | | | | |

From Table 4, Lncident shows the most balanced result among these tools compared with other tools. The results of local version and web server fluctuate slightly because the web server may be trained with different genome assembly. We also generated a synthetic dataset based on this test set. The new dataset were generated by utilizing the function “syn()” of R package “synthpop” [8]. The default parameters were selected, and synthesizing method was “classification and regression tress (CART)”. The sensitivity of Lncident is sensitivity is 1.0000 and specificity of Lncident is 0.9990.

(2). Comprehensive comparisons for lncRNA identification of invertebrate species:

According your suggestion, CPAT and PLEK were re-trained using the dataset of *C. elegans*. The training set for *C. elegans* contains 22,929 coding sequences and 22,929 ncRNAs. The test set contains 2,000 coding sequences and 2,000 ncRNAs. When assessing the performances of these tools on *S. cerevisiae,* all these datasets were used as training set to train the models for lncRNA identification of *S. cerevisiae* . The results are listed in Table 5.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 5. Comparisons of *C. elegans* | | | | | | | | | | | | |
| Tools | **Sensitivity** | | **Specificity** | | **Accuracy** | | **F-measure** | | **MCC** | | **Kappa** | |
| CPC.web | | 0.9990 | | **0.9895** | | **0.9942** | | **0.9942** | | **0.9885** | | **0.9884** | |
| CPAT.web | | 0.9910 | | 0.9187 | | 0.9548 | | 0.9564 | | 0.9120 | | 0.9097 | |
| CNCI | | 0.9938 | | 0.7744 | | 0.8841 | | 0.8955 | | 0.7873 | | 0.7681 | |
| PLEK | | 0.9987 | | 0.4526 | | 0.7256 | | 0.7845 | | 0.5387 | | 0.4513 | |
| Lncident\* | | **1** | | 0.9545 | | 0.9772 | | 0.9778 | | 0.9555 | | 0.9545 | |
| CPAT.train\*\* | | **0.9995** | | 0.9950 | | **0.9972** | | **0.9973** | | **0.9945** | | **0.9945** | |
| PLEK.train\*\* | | 0.9795 | | 0.9950 | | 0.9872 | | 0.9872 | | 0.9746 | | 0.9745 | |
| Lncident.train\*\* | | 0.9950 | | **0.9975** | | 0.9962 | | 0.9962 | | 0.9925 | | 0.9925 | |
| For tools with default models, Lncident presented the best result among the alignment-free methods. Both Lncident and CPAT outperformed CPC by utilizing new-trained model. \*Lncident with model trained on human. \*\*The suffix of “train” means the tools with model trained on *C. elegans*. | | | | | | | | | | | | |

From Table 5, CPC had the best performance for the sequences selected from the database. For other tools on human model, Lncident presented the best result. When CPAT, PLEK and Lncident were re-trained with *C. elegans* dataset, both Lncident and CPAT were better than CPC. CPAT showed the best result. However, users have to determine a cutoff for CPAT, and the best cutoffs of different species vary widely. For example, the cutoff of human is 0.364, mouse is 0.440, while the best cutoff for *C. elegans* is 0.850 in our research. Lncident is only 0.001 lower than CPAT on accuracy, and no cutoff needs to be chosen by users. In addition, Lncident provided the specific model for invertebrate species

Table 6 shows the performances on *S. cerevisiae*. Lncident was the best tool among the alignment-free tools. Among the tools tested with default model, Lncident was still better than CPAT (web interface), CNCI and PLEK.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 6. Comparisons of *S. cerevisiae* | | | | | | | | | | | | | |
| Tools | | **Sensitivity** | | **Specificity** | | **Accuracy** | | **F-measure** | | **MCC** | | **Kappa** | |
| CPC.web | 0.9734 | | **0.9687** | | **0.9697** | | **0.9327** | | **0.9145** | | **0.9132** | |
| CPAT.web | **1** | | 0.7980 | | 0.8416 | | 0.7316 | | 0.6785 | | 0.6304 | |
| CNCI | 0.9758 | | 0.6760 | | 0.7407 | | 0.6190 | | 0.5377 | | 0.4598 | |
| PLEK | 0.9903 | | 0.5507 | | 0.6456 | | 0.5468 | | 0.4491 | | 0.3407 | |
| Lncident\* | **1** | | 0.8620 | | 0.8918 | | 0.7996 | | 0.7578 | | 0.7295 | |
| CPAT.train\*\* | **1** | | 0.9487 | | 0.9597 | | 0.9147 | | 0.8942 | | 0.8886 | |
| PLEK.train\*\* | 0.9758 | | 0.8053 | | 0.8421 | | 0.7274 | | 0.6682 | | 0.6262 | |
| Lncident.train\*\* | **1** | | **0.9680** | | **0.9749** | | **0.9451** | | **0.9312** | | **0.9289** | |
| Lncident presented the best result on *S. cerevisiae* dataset. \*Lncident with model trained on human. \*\*The suffix of “train” means the tools with model trained on *C. elegans*. | | | | | | | | | | | | | |

For the tools with new-trained model, Lncident was the only one outperformed CPC. Lncident model for invertebrata achieved the best results and Lncident model for human presented the best performance among the alignment-free tools. CPC needs several hours to complete the prediction process. As we just discussed, the performance of CPAT largely depends on the different cutoff values for different species. The performance of CPAT in Table 6 was obtained by utilizing the model trained on *C. elegans* (best cutoff 0.850). However, the cutoff values for different species are diverse dramatically and cannot be applied to other species directly. We even cannot guarantee that 0.850 is the most appropriate cutoff for *S. cerevisiae*. In practice, for some species, there maybe are not sufficient information to determine the best cutoff value. When we use the model trained by other species, Lncident presents the better result.

(3). As you suggested, Matthews’ correlation coefficient (MCC) and Kappa value have been added in Table 6.

**Concern 3:** Third, please use your trained SVM classifiers and other methods to screen the entire human genome or other genomes of species that were not included in your training data. Report the number of newly-discovered lncRNA sequences and the running time. This experiment can be used to evaluate if Lncident is in fact appropriate for a large-scale analyses. Lastly, please make sure the predictions from the R program and the web server are identical and easy to understand.

**Response:** Thank you for this comment.

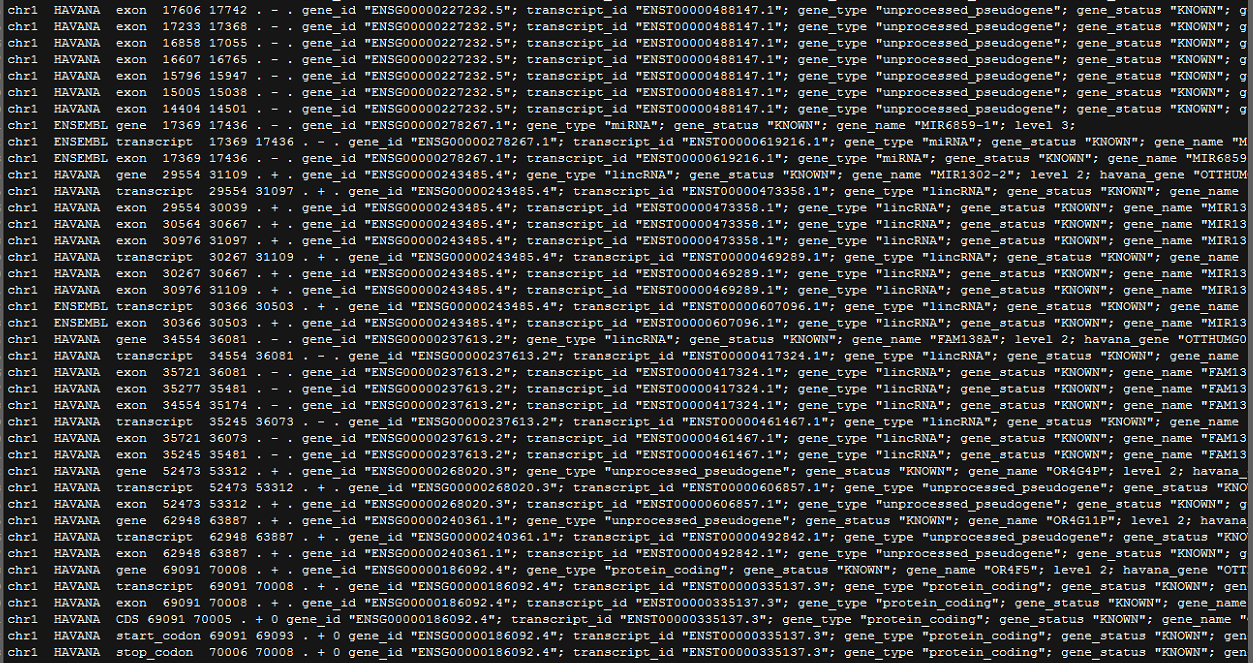
1. The annotated and separated sequences can be obtained from a genome .fasta file and a main annotation file (.gtf or .gff3 file). However, the genome is generally the sequences assembled together. We applied Lncident to screen the whole human genome to identify the lncRNAs by the following steps: (a) Determination of length of sliding window determination; (b) Training new model; (c) Construction of the test set; (d) Splitting the genome into separated sequences using sliding windows; (e) Mapping onto the whole genome with .gtf annotation file and (f) Evaluation. The details of each step are given as follows:
2. Utilizing sliding window to search the whole genome to identify whether the region in the sliding window is lncRNA. First of all, we need to determine the length of sliding window. We selected sliding window with size of 100nt, 300nt and 500nt and with overlap of 50nt, 150nt, 250nt to scan the whole whole genome.
3. The lncRNAs and CDs sequences, obtained from GENCODE [7] and Ensembl [12] database, were split according to the different lengths of the sliding windows. The total numbers of windows for lncRNA and mRNA were listed in Table 7.

|  |  |  |
| --- | --- | --- |
| **Table 7.** The total windows (sub-sequences) of two class sequences from Ensembl | | |
|  | **sub-sequences of lncRNA** | **sub-sequences of CDs** |
| window of 100nt | 294,512 | 1,210,116 |
| window of 300nt | 106,692 | 437,581 |
| window of 500nt | 70,238 | 283,120 |

From all these sub-sequences, 25,000 lncRNAs sub-sequences and 25,000 CDs sub-sequences were selected to build the training sets. We built three models based on three different lengths of the windows.

1. We then chose chromosome 22 as the test set. We scanned the whole sequence and split the sequence into sub-sequences. If one window includes “N” more than 50%, this window will be abandoned.
2. We predicted these three test sets by utilizing the three trained models. The running time were calculated and listed in Table 8. This process was run on a Linux server with 32 Intel CPUs (1,999.913 MHz); eight cores per socket and 128 GB memory.

|  |  |  |
| --- | --- | --- |
| **Table 8.** The total windows (sub-sequences) of chromosome 22 of human genome | | |
|  | **Valid windows** | **Running time** |
| window of 100nt | 391,600 | 12h 08min 18s |
| window of 300nt | 130,539 | 03h 44min 09s |
| window of 500nt | 78,326 | 02h 05min 18s |

1. All the windows were mapped onto the original genome with annotation file in order to evaluate the performance. The figure below is human genome’s .gtf annotation file. Since the sliding window method can only predict the sequences roughly, one window will be regarded as “protein-coding” if the start position of this window is between the annotated protein-coding gene’s start position (column 4) and end position (column 5). Similarly, we labelled one window as “True Positive” if this window is not in these ranges.
2. The performances of different models are listed in Table 9.

|  |  |  |
| --- | --- | --- |
| **Table 9.** The performances of Lncident with three model | | |
| **Model** | **Sensitivity** | **Specificity** |
| window 100nt | 0.8809 | 0.1323 |
| window 300nt | 0.7802 | 0.2161 |
| window 500nt | 0.8008 | 0.1634 |

We also conducted this experiment on CPAT and PLEK. Nonetheless, as stated in the manual of PLEK，training a new model of PLEK is a very time-consuming process. For human dataset 2 in our updated evaluation, it takes 3 days to train a new human model. Similar to the evaluation for lncRNA-ID [13], the default model of PLEK was selected to compare with other tools because of high time-consuming training process of PLEK. Due to the limited time of revision, we have not enough time to train another three models of PLEK. Thus, we here only re-trained CPAT with the same datasets as Lncident, while PLEK was selected as the default model. The result of PLEK can also help us to assess whether a traditional tool can handle the task of predicting lncRNAs on the scale of genome. The results were displayed in Table 10.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table 10.** The performances of CPAT.train and PLEK | | | | |
| **Model** | **CPAT.train** | | **PLEK** | |
| Sensitivity | Specificity | Sensitivity | Specificity |
| window 100nt | 0.8726 | 0.1545 | 1 | 0 |
| window 300nt | 0.9259 | 0.1025 | 0.9998 | 0.0002 |
| window 500nt | 0.9435 | 0.0824 | 0.9975 | 0.0037 |

1. The prediction of genome is totally different from prediction of transcripts, thus, three new-trained models of Lncident used for genome prediction are also different with Lncident we proposed in our original manuscript. In this revised version we calculated the running time on the human dataset 2 containing 4,000 mRNAs and 4,000 lncRNAs. This test was performed on a home-used PC, which can evaluate whether Lncident can handle the prediction task without the limitation of hardware configuration. It is hard to calculate precise running time of testing on web server, thus, we can only obtain a rough approximation for CPC and Lncident. For the detailed information, please refer to Table 11.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table 11.** Running time on human dataset | | | |
| **Tools** | | **Running Time** | |
| **Stand-alone** | **Web server\*** |
| CPC | | 5 day 4h 37min 43s | About 1 day |
| CPAT | | 16s |  |
| CNCI | | 37min 53s |  |
| PLEK | | 3min 04s |  |
| Lncident | | 3min 01s | About 12min |
| *Test on: Intel® Core™ i7-2600 CPU @ 3.40 GHz; 8 GB RAM* | | |
| The test dataset contains 4,000 mRNAs and 4,000 lncRNAs.\*It is hard to calculate preciserunning time of testing on web server, thus, we can only obtain a rough approximation. We omitted the result of CPAT web server, because only file size less than 10 MB is allowed, and when we split our file, we found the average error is too serious to accept. | | |

In our original manuscript, we introduced Lncident as follows:

“Lncident is much faster than Coding-Potential Calculator and Coding-Non-Coding Index.” (in “Abstract”)

“Lncident can complete the identification process in several minutes while CPC (web server) needs dozens of hours and CNCI need tens of minutes.” (in “Introduction”)

According to out experiment, out description is accurate.

CPAT is the most efficient tool mainly because CPAT based on logistic regression which is much faster than SVM. Since CPC is an alignment-based tool, it needs lots of time to align the sequences against the reference database. The script of Lncident is written in R which is slower than Python, but the speed of Lncident is comparable to PLEK (written in Python), and much faster than CPC and CNCI.

1. The database NONCODE [14] now have collected 487,164 non-coding transcripts of various species. Actually, the non-coding sequences provided by database NONCODE are those sequences whose predicted results obtained from CNCI (See <http://www.noncode.org/introduce.php>). Lncident outperforms CNCI with higher accuracy and is more than 10 times faster than CNCI as well. Hence, Lncident is capable of processing massive-scale analysis.
2. The web interface of Lncident is developed based on Lncident’s R package, which provides identical results with the ones of stand-alone version.

Other minor modifications please refer to our revised manuscript.

Thank you again for the review of our paper.

**Review #2**

**Concern 1:** page 5, the second line "The Figure 2 shows that the differences between the distribution of lncRNAs and CDs are significant." is not clear. How do you calculate significance? How do you determine significance?

**Response:** Thank you for this comment. The word “significant” is a rhetoric expression more than a statistical term. To avoid this ambiguity, we have replaced this word with “obvious”. The original *k*-mer scheme is based on unbalanced distribution of nucleotides. From Fickett Score [15][16] to codon bias, all these methods show satisfying results (Accuracy of Fickett Score: 0.750) using the k-mer nucleotide frequencies of lncRNAs and protein-coding sequences. In recent years, besides CPAT (2013) [1], CNCI (2013) [9] and PLEK (2014) [2], several novel tools such as lncRNA-MFDL (2015) [6], LncRNApred (2016) [5] and FEElnc (2016) [17] all selected *k*-mer scheme as features, which indicates that the *k*-mer frequencies have strong discriminative power. Since lncRNA-MFDL and lncRNApred provided no script; FEElnc was not published when we developed Lncident, we did not compare Lncident with these tools. However, according to ROC curve on different species, Lncident displayed the best results among alignment-free methods. In this revised version, more comparisons were conducted in order to comprehensively evaluate Lncident. For detailed information, please refer to our revised manuscript.

**Concern 2:** p8, the first paragraph. I am not sure the point made is fair here. Although other methods might contain identical sequences in their pre-trained model, they are not trained based on data set collected by authors. However, Lncident is trained and tested on the collected data set. In protein function prediction field, a fair data set should not contain any identical or even similar sequence (sequence identity >= 30%). I would suggest authors split data set into two parts: easy one (with similar or identical sequences) and difficult one (non-identical sequences). Authors could report the performance of those two sets. Lncident might outperform CPC in difficult one since the later method is based on sequence similarity.

**Response:** Thank you for this comment. Inspired by your comments, we have made a major modification in the section of “evaluation” in this revised version.

1. Lncident compared with CPC and other tools on the datasets selected from GENCODE database are remained as the original version (can be regarded as dataset easy to predict), but considering that CPC will definitely present an excellent result when the sequences are selected from the database, we evaluated these tools on new human datasets provided in [13]. Figure 1 shows the result of this new comparison.

The results of web server and stand-alone version of CPC and CPAT show slight differences, thus, both web server and local version were tested. For CPAT and PLEK, two tools can be trained by users, we trained these tools with the same training set as Lncident. The training set contains 8,300 lncRNAs and 20,000 mRNAs while the test set contains 4,000 lncRNAs and 4,000 mRNAs. Only one transcript from each gene is collected in training set. In addition, the training and testing datasets have no transcripts from the same genes. From the table 1, we can find that CPC, as an alignment-based tool, has a strong tendency to misclassify lncRNAs as protein-coding transcripts. Compared with the performances of CPAT.train and PLEK.train, PLEK is more sensitive to imbalanced training sets than CPAT. Lncident has the most balanced and satisfying performance.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Table 1. The performances on human dataset 2 | | | | | | |
| Tools | **Sensitivity** | **Specificity** | **Accuracy** | **F-measure** | **MCC** | **Kappa** |
| CPC.local | 0.6613 | **1** | 0.8306 | 0.7961 | 0.7028 | 0.6612 |
| CPC.web | 0.6625 | 0.9992 | 0.8309 | 0.7966 | 0.7028 | 0.6618 |
| CPAT.local | 0.9333 | 0.9802 | 0.9568 | 0.9557 | 0.9145 | 0.9135 |
| CPAT.web | 0.9535 | 0.9673 | 0.9604 | 0.9601 | 0.9208 | 0.9208 |
| CNCI | 0.9702 | 0.9157 | 0.9430 | 0.9445 | 0.8873 | 0.8860 |
| PLEK | **0.9952** | 0.8918 | 0.9435 | 0.9463 | 0.8918 | 0.8870 |
| CPAT.train\* | 0.9160 | 0.9848 | 0.9504 | 0.9486 | 0.9029 | 0.9008 |
| PLEK.train\* | 0.7622 | 0.9507 | 0.8565 | 0.8416 | 0.7260 | 0.7130 |
| Lncident | 0.9535 | 0.9795 | **0.9665** | **0.9661** | **0.9333** | **0.9330** |
| Lncident displayed a satisfying overall performance. CPC and CPAT were tested on stand-alone version and web server. \*The suffix of “train” meansCPAT and PLEK with the new-trained model. | | | | | | |

1. For *C. elegans* and *S. cerevisiae*, we conducted a more comprehensive comparison. PLEK and CPAT were re-trained with the same dataset. Both default model and new trained model were tested in this updated manuscript. The training set of *C. elegans* for CPAT, PLEK and Lncident contains 22,929 coding sequences and 22,929 ncRNAs. The test set contains 2,000 coding sequences and 2,000 ncRNAs. Both two datasets were used to construct the test set to evaluate the different tools.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 2. Comparisons of *C. elegans* | | | | | | | | | | | | |
| Tools | **Sensitivity** | | **Specificity** | | **Accuracy** | | **F-measure** | | **MCC** | | **Kappa** | |
| CPC.web | | 0.9990 | | **0.9895** | | **0.9942** | | **0.9942** | | **0.9885** | | **0.9884** | |
| CPAT.web | | 0.9910 | | 0.9187 | | 0.9548 | | 0.9564 | | 0.9120 | | 0.9097 | |
| CNCI | | 0.9938 | | 0.7744 | | 0.8841 | | 0.8955 | | 0.7873 | | 0.7681 | |
| PLEK | | 0.9987 | | 0.4526 | | 0.7256 | | 0.7845 | | 0.5387 | | 0.4513 | |
| Lncident\* | | **1** | | 0.9545 | | 0.9772 | | 0.9778 | | 0.9555 | | 0.9545 | |
| CPAT.train\*\* | | **0.9995** | | 0.9950 | | **0.9972** | | **0.9973** | | **0.9945** | | **0.9945** | |
| PLEK.train\*\* | | 0.9795 | | 0.9950 | | 0.9872 | | 0.9872 | | 0.9746 | | 0.9745 | |
| Lncident.train\*\* | | 0.9950 | | **0.9975** | | 0.9962 | | 0.9962 | | 0.9925 | | 0.9925 | |
| \*Lncident with model trained on human. \*\*The suffix of “train” means the tools with model trained on *C. elegans*. | | | | | | | | | | | | |

The sequences were selected from database Ensembl [12], and CPC achieved the best result among tools with default model. For other alignment-free tools, Lncident presented the most balanced and satisfying result. Both Lncident and CPAT are better than CPC after re-training the models.

The test set of *S. cerevisiae* contains 413 non-coding sequences and 1,500 coding sequences.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Table 3. Comparisons of *S. cerevisiae* | | | | | | | | | | | | | |
| Tools | | **Sensitivity** | | **Specificity** | | **Accuracy** | | **F-measure** | | **MCC** | | **Kappa** | |
| CPC.web | 0.9734 | | **0.9687** | | **0.9697** | | **0.9327** | | **0.9145** | | **0.9132** | |
| CPAT.web | **1** | | 0.7980 | | 0.8416 | | 0.7316 | | 0.6785 | | 0.6304 | |
| CNCI | 0.9758 | | 0.6760 | | 0.7407 | | 0.6190 | | 0.5377 | | 0.4598 | |
| PLEK | 0.9903 | | 0.5507 | | 0.6456 | | 0.5468 | | 0.4491 | | 0.3407 | |
| Lncident\* | **1** | | 0.8620 | | 0.8918 | | 0.7996 | | 0.7578 | | 0.7295 | |
| CPAT.train\*\* | **1** | | 0.9487 | | 0.9597 | | 0.9147 | | 0.8942 | | 0.8886 | |
| PLEK.train\*\* | 0.9758 | | 0.8053 | | 0.8421 | | 0.7274 | | 0.6682 | | 0.6262 | |
| Lncident.train\*\* | **1** | | **0.9680** | | **0.9749** | | **0.9451** | | **0.9312** | | **0.9289** | |
| Lncident presented the best result on *S. cerevisiae* dataset. \*Lncident with model trained on human. \*\*The suffix of “train” means the tools with model trained on *C. elegans*. | | | | | | | | | | | | | |

From Table 2 and Table 3, CPC shows the best performance. For other tools with human model, Lncident has the best result. When CPAT, PLEK and Lncident are re-trained with the same datasets, only CPAT slightly outperformed Lncident on *C. elegans*. However, CPAT, logistic regression model, needs to choose cutoff value by users for the new trained model. The cutoff values of different species are varies comparatively great which makes the cutoff cannot be applied to other species directly. For example, the optimal cutoff value for human species is 0.364. For mouse species the the optimal cutoff value is 0.44, but the optimal cutoff value is 0.850 *C. elegans* in our experiment. Not all the species have sufficient data used to determine the best cutoff. For *S. cerevisiae*, we can not guarantee that the cutoff obtained from only 413 ncRNAs in database can achieve a satisfying result. When we utilize the model trained on other species, Lncident displayed better results. Lncident is only 0.001 lower than CPAT on accuracy, but no cutoff needs to be chosen by users. In addition, Lncident has the best overall performances.

**Concern 3:** p2, the last sentence should give the detailed performance of CPC.

**Response:** Thank you for this comment. We have made the suggested changes.

**Concern 4:** p3, the first paragraph might summary running time analysis into a table.

**Response:** Thank you for this comment. We added running time analysis in the “Evaluation” section. The test dataset contains 4,000 mRNAs and 4,000 lncRNAs. It is hard to calculate precise running time of testing on web server, thus, we can only obtain a rough approximation for CPC and Lncident. Benefited from logistic regression model, CPAT is the most efficiency tool. CPC is the slowest tool because it is an alignment-based tool. The script of Lncident is written in R which is slower than Python, but the speed of Lncident is comparable to PLEK (which is written in Python), and much faster than CPC and CNCI.

|  |  |  |
| --- | --- | --- |
| **Table 4. Running time on human dataset** | | |
| **Tools** | **Running Time** | |
| **Stand-alone** | **Web server\*** | |
| CPC | 5 day 4h 37min 43s | About 1 day | |
| CPAT | 16s |  | |
| CNCI | 37min 53s |  | |
| PLEK | 3min 04s |  | |
| Lncident | 3min 01s | About 12min | |
| *Test on: Intel® Core™ i7-2600 CPU @ 3.40 GHz; 8 GB RAM* | | |

**Concern 5:** p4, For "Features Extraction" section, I would suggest reform variable k into italic face.

**Response:** Thank you for this comment. In this revised version, we checked the whole manuscript and all the *k* have been transformed into italic format.

**Concern 6:** p5, the sentence "The Supplementary Figure S1 is somewhat different from Figure 2," is not clear. Authors should describe the difference Supp. Fig1 and Fig2 in detail, i.e., the bar increase or decrease in Supp. Fig1. **Response:** Thank you for this comment. It is useful to describe the differences. Providing some specific examples can make our manuscript precise and to the point. Hence, we have changed the expression as:

“We can notice that the Supplementary Figure S1 is somewhat different from the Figure 2. In Figure 2, the log2-ratios of bases combination such as AC, AG, ATC and ATG are all positive numbers while in Figure S1 are all negative. In addition, the log2-ratio of TAA nearly reaches 1.0 in Figure 2 while in Figure S1 even less than 0.5. There are many other obvious differences between these two figures, which indicates that …”

**Concern 7:** p5, the sentence "which indicates that the untranslated regions of protein-coding transcripts can affect the performance of the classifier." is not clear neither. Does it improve or worsen the performance?

**Response:** Thank you for this comment. The *k*-mer scheme calculates the composition frequencies of nucleotides. However, the untranslated regions (UTRs) can be regard as non-coding sequences, which decreases the “protein-coding probability” for classifier. Thus, the classifier may predict some mRNAs as lncRNAs. To avoid the ambiguous expression in our manuscript, we have added an explanation in the modified version:

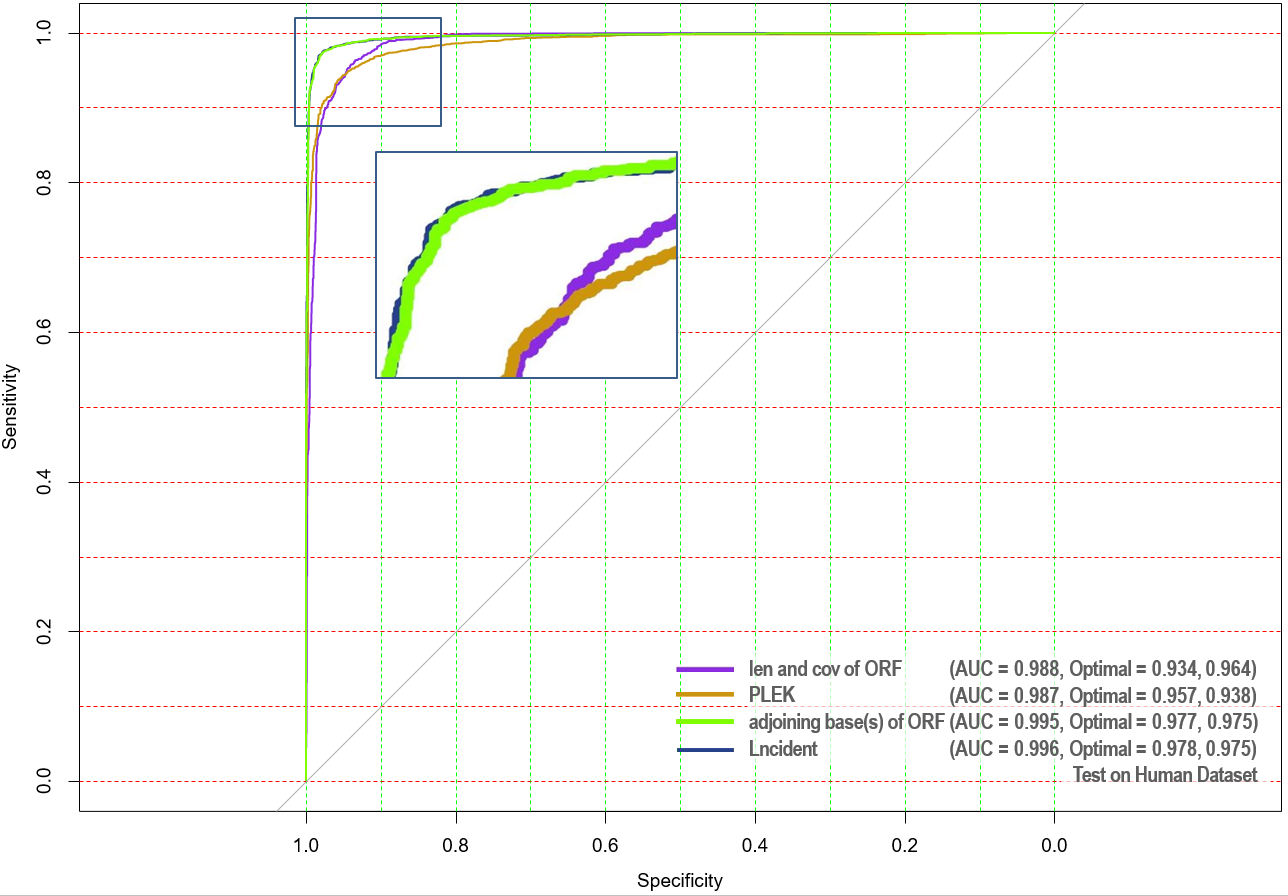
“…the untranslated regions of protein-coding transcripts indeed affect the performance of the classifier. UTRs make protein-coding gene like non-coding sequence, which may lead to a wrong classification for protein-coding sequence.”

**Concern 8:** p5, authors could explain a bit of "the *k*-mer scheme of PLEK." **Response:** Thank you for this comment. We have provided a brief explanation of improved *k*-mer scheme of PLEK in revised version:

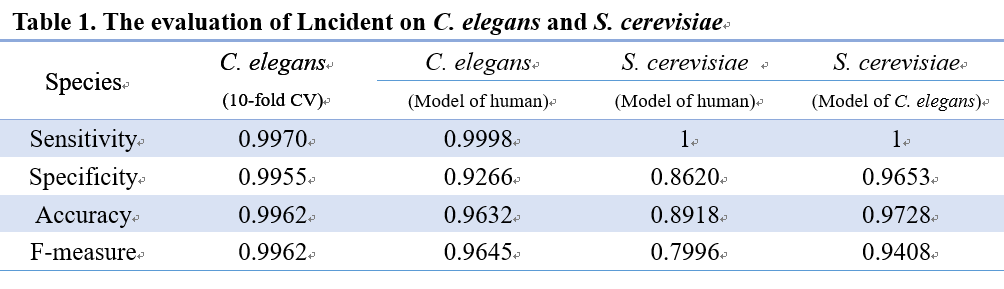
“PLEK calculates the nucleotides compositions on the whole sequence, and *k*-mer frequencies are also multiplied with weights in order to scale the data.”

**Concern 9:** p6, which species is used in the Figure 3?

**Response:** Thank you for this comment. The specie in Figure 3 is human. To make our manuscript more explicit, we modified this figure by adding the specie information at the right bottom. Now the new figure is as follows:

****

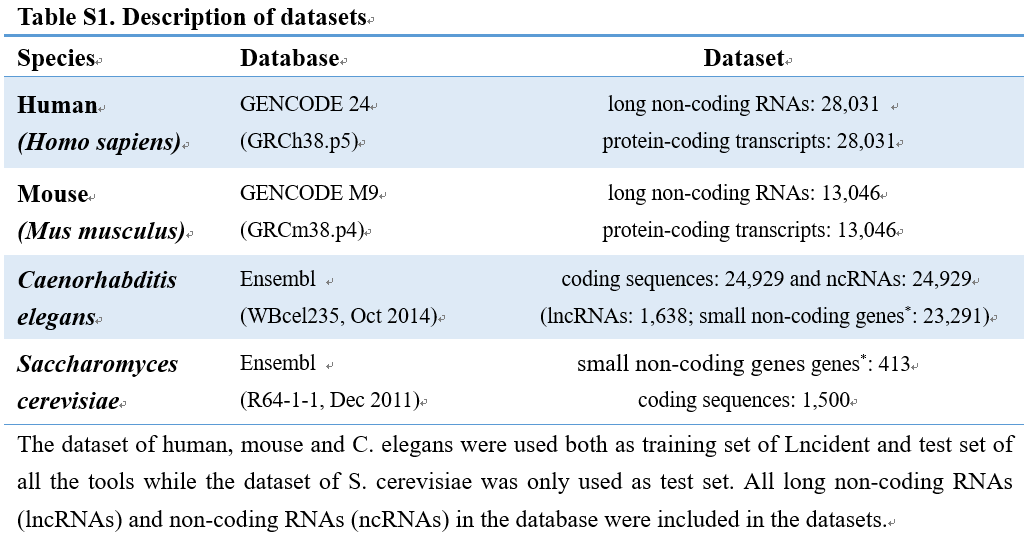
**Concern 10:** p9, it is surprising that the F-measure of the last column, 0.9404, is worse than the one of the third column, 0.9645.

**Response:** Thank you this comment. This is our original table:

The F-measure, 0.9408, in the last column is the performance of *S. cerevisiae*, while 0.9645 in the second column is of *C. elegans*. We all know that training a specific model for a specific species usually can achieve the better result. From the second column, the performance of predicting *C. elegans* sequences by utilizing model trained on human dataset is acceptable (F-measure = 0.9645). It seems that there is no need to train a new model for *C. elegans*. Nonetheless, when we applied the model trained on human dataset to other species, the performances are unsatisfying (the third column). A new model for *C. elegans was* trained to extend the application scope of Lncident. The performances of *C. elegans* and *S. cerevisiae* are enhanced. In short, the second and the third columns indicate the necessity of training a new model; the first and the second, the third and the forth columns display the improvements of the performances by utilizing the new model. However, the last column and the second column are the performances of different species.

**Concern 11:** Expect F-measure, the performance of the last column is better than the one of the third column.

**Response:** Thank you for your comments. This is the original table (Table S1) about the descriptions of dataset:



The total number of the sequences of *C. elegans* in Ensembl database is 24,929 non-coding RNAs (ncRNAs) and 30,939 coding sequences (CDs). To build a balanced training dataset, we only selected 24,929 CDs and this dataset is used to assess the performance of model trained on human as well.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Table 5. Details of performances on two dataset | | | | | | | |
| Species | **TP** | **FN** | **FP** | **TN** | **Sensitivity** | **Specificity** | **F-measure** |
| *C. elegans*  Model for Human | 24,925 | 4 | 1,829 | 23,100 | 0.9998 | 0.9266 | 0.9645 |
| *S. cerevisiae*  Model for *C. elegans* | 413 | 0 | 52 | 1448 | 1 | 0.9653 | 0.9408 |

Other minor modifications please refer to our revised manuscript.

Thank you again for undertaking the review of our paper.

## Reference

1. Wang L, Park HJ, Dasari S, et al. CPAT: Coding-potential assessment tool using an alignment-free logistic regression model. Nucleic Acids Res. 2013; 41:1–7

2. Li A, Zhang J, Zhou Z. PLEK: a tool for predicting long non-coding RNAs and messenger RNAs based on an improved k-mer scheme. BMC Bioinformatics 2014; 15:311

3. Kong L, Zhang Y, Ye ZQ, et al. CPC: Assess the protein-coding potential of transcripts using sequence features and support vector machine. Nucleic Acids Res. 2007; 35:345–349

4. Achawanantakun R, Sun Y. BIOINFORMATICS LncRNA-ID : Long non-coding RNA IDentification using balanced random forests. 2014; 31:1–8

5. Pian C, Zhang G, Chen Z, et al. LncRNApred: Classification of Long Non-Coding RNAs and Protein-Coding Transcripts by the Ensemble Algorithm with a New Hybrid Feature. PLoS One 2016; 11:e0154567

6. Fan X-N, Zhang S-W. lncRNA-MFDL: identification of human long non-coding RNAs by fusing multiple features and using deep learning. Mol. Biosyst. 2015; 11:892–897

7. Derrien T, Johnson R, Bussotti G, et al. The GENCODE v7 catalogue of human long non-coding RNAs : Analysis of their structure , evolution and expression. Genome Res. 2012; 22:1775–1789

8. Nowok B, Raab GM, Snoke J, et al. synthpop: Generating Synthetic Versions of Sensitive Microdata for Statistical Disclosure Control. 2016; https://cran.r-project.org/package=synthpop

9. Sun L, Luo H, Bu D, et al. Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts. Nucleic Acids Res. 2013; 41:e166–e166

10. Achawanantakun R, Chen J, Sun Y, et al. Sequence analysis LncRNA-ID : Long non-coding RNA IDentification using balanced random forests. 2015; 1–9

11. Sun L, Liu H, Zhang L, et al. lncRScan-SVM: A Tool for Predicting Long Non-Coding RNAs Using Support Vector Machine. PLoS One 2015; 10:e0139654

12. Yates A, Akanni W, Amode MR, et al. Ensembl 2016. 2016; 44:710–716

13. Achawanantakun R, Chen J, Sun Y, et al. LncRNA-ID: Long non-coding RNA IDentification using balanced random forests. Bioinformatics 2015; 31:3897–3905

14. Zhao Y, Li H, Fang S, et al. NONCODE 2016: an informative and valuable data source of long non-coding RNAs. Nucleic Acids Res. 2015; 44:1–6

15. Fickett JW. Recognition of protein coding regions in DNA sequences. Nucleic Acids Res. 1982; 10:5303–5318

16. Fickett JW, Tung C-SS. Assessment of protein coding measures. Nucleic Acids Res. 1992; 20:6441–6450

17. Wucher V, Legeai F, Hedan B, et al. FEELnc: A tool for Long non-coding RNAs annotation and its application to the dog transcriptome. bioRxiv 2016;