**Methods**

**Dataset**

Human lncRNA and protein-coding transcripts were downloaded from GENCODE release 30 [36]. After removing the transcripts whose length is less than 200nt, we obtained 29,698 lncRNAs and 75,153 protein-coding transcripts, which are from 15,855 lncRNA genes and 18,909 protein-coding genes, from which we randomly selected 23,000 lncRNAs and 23,000 protein-coding transcripts to construct the training dataset (namely Human training dataset). In the remaining transcripts, we randomly selected 6,000 lncRNAs and 6,000 protein-coding transcripts to construct the testing dataset (namely Human testing dataset).

Due to the fact of that the transcripts in Human transcript testing dataset may be from the same genes with the transcripts in Human transcript training dataset, which may cause the “memory” effect. To investigate the “memory” effect, we split the GENCODE annotation gene-wise into training and testing datasets. That is, for each lncRNA gene (or protein-coding gene), we used the corresponding transcript of the lncRNA gene (or protein-coding gene) with max-length to represent this lncRNA gene (or protein-coding gene). Then, we obtained 15,855 lncRNA transcripts and 18,909 protein-coding transcripts, from which we randomly selected 12,000 lncRNA transcripts and 12,000 protein-coding transcripts to construct the training dataset (namely Human gene-wise training dataset), and all the remain transcripts were used to build the testing dataset (namely Human gene-wise testing dataset). In addition, we also built another dataset (namely Human non-gene-wise training dataset) by randomly selected 12,000 lncRNA transcripts and 12,000 protein-coding transcripts from Human training dataset.

We also built other 11 cross-species testing datasets, in which mouse lncRNAs and protein-coding transcripts were downloaded from GENCODE release M20 [36], and other 10 cross-species (e.g., arabidopsis, chicken, Bos taurus, C.elegans, chimpanzee, frog, fruit fly, gorilla, pig, and zebrafish) testing dataset were downloaded from RefSeq v94 [37]. The statistics of all datasets are listed in Additional file 5.

**LncRNA\_Mdeep**

LncRNA\_Mdeep mainly consists of the following phases: 1) Extract the OFH feature and *k*-mer feature from transcript sequences, and use one-hot encoding strategy to encode the transcript sequences; 2) Build two DNN models and a CNN model to mine the high-level representations from OFH feature, *k*-mer feature, and one-hot encoding of transcript sequences, respectively; 3) Fuse the learned representations (namely OFH\_DNN descriptor, *k*-mer\_DNN descriptor and One-hot\_CNN descriptor) to represent the transcript sequences; 4) Feed three descriptors into a DNN to distinguish lncRNAs from protein coding transcripts. The overview of lncRNA\_Mdeep is illustrated in Figure 1.

**Feature extraction and one-hot encoding**

Given a transcript sequence  with *L* nucleotides, where  denotes the first nucleotide,  denotes the second nucleotide, and so on. We extract two kinds of features from the transcript sequences to convert them into vectors.

The first one is the OFH feature which consists of the length and coverage of the open reading frame (ORF), a Fickett score, and a Hexamer score. We first calculated the length and the coverage of ORF, which is identified as the longest reading frame in three forward frames starting with a start codon and ending with a stop codon, then obtained the Fickett score *SF* from literature [38]. Fickett score *SF* can be attained by calculating the percentage compositions of A, C, G, T and their position values (i.e., Apos, C*pos*, G*pos*, and T*pos*) with the formula , where B1, B2, B3 denote the occurrence number of nucleotide B (i.e., A, C, G and T) in first, second, and third position of the codon, respectively, and transforming these eight values to a TESTCODE score. The Hexamer score *SH* was calculated by using the formula, where and (*i* = 1,2,…,4096) represent in-frame coding and non-coding hexamer frequency, respectively. Finally, the OFH feature can be represented as, where *lORF* denotes the length of ORF, *SF* denotes the Fickett score, and *SH* denotes the Hexamer score.

The second feature is the *k*-mer frequency feature denoted as, where  is the occurrence frequency of *k* neighboring bases in the transcript sequence.

One-hot encoding translates the A, T, C, G characters into a binary vector of (1,0,0,0), (0,1,0,0), (0,0,1,0) and (0,0,0,1), respectively. Therefore, the transcript with length *L* is denoted as a 4 × L matrix, i.e.,, whereis the corresponding binary vector of *ith* nucleotide in the transcript.

**High-level abstract representations**

Two DNN models and a CNN model were built to learn the hidden high-level abstract representations from different input modalities. DNN model consists of an input layer, multiple hidden layers and an output layer, which is used to model high-level abstractions in input data with a deep architecture composed of multiple non-linear transformation [39, 40]. We built two DNN models for inputs of OFH feature and *k*-mer feature, respectively. Then, the OFH\_DNN descriptor and *k*-mer\_DNN descriptor were obtained from the last hidden layers of two DNNs, which denote the final representations of OFH feature and *k*-mer feature, respectively.

Furthermore, a CNN model was built to learn the hidden high-level abstract representations from one-hot encoding of transcript sequences. CNN model consists of convolution layer, batch normalization, rectified linear unit, and pooling [41].

  (1)

where *R* is the output vector of convolutional module; *X* is the input vector; , and *M* are the parameters of batch normalization and convolution layers. Since the input of CNN requires fixed-length input, we set a parameter of *maxlen* to make the one-hot encoding of transcript sequence  to be a 4 × *maxlen* matrix.

  (2)

The output of CNN is defined as the one-hot\_CNN descriptor, which denotes the high-level abstract representations of raw transcript sequences.

**Multimodal framework**

To distinguish the lncRNAs from protein-coding transcripts, we concatenated the OFH\_DNN descriptor, k-mer\_DNN descriptor, and one-hot\_CNN descriptor, then fed them into a DNN to predict the probability of the input transcript sequence to be a lncRNA. There are two steps to train our lncRNA\_Mdeep. The first step is training a DNN for OFH feature, a DNN for *k*-mer feature, and a CNN for one-hot encoding, respectively. The parameters in two DNNs and CNN architectures are trained using the labeled data. The second step is learning the parameters of the DNN for final classification and processing a fine-tuning for renew all parameters in the whole multimodal framework.

**Evaluation metrics**

The following metrics of accuracy (ACC), sensitivity (*Sn*), specificity (*Sp*), and Matthew’s correlation coefficient (MCC) are used to measure the performance of lncRNA\_Mdeep.

  (3)

  (4)

  (5)

  (6)

where TP and TN are the number of correctly predicted lncRNAs and protein-coding transcripts, respectively; FP and FN are the number of incorrectly predicted lncRNAs and protein-coding transcripts, respectively.