Briefings in Bioinformatics, 2024, bbae504
https://doi.org/10.1093/bib/bbae504
Problem Solving Protocol

MSlocPRED: deep transfer learning-based identification of multi-label mRNA subcellular localization

Yun Zuo[1], Bangyi Zhang[1], Wenying He[2], Yue Bi[3], Xiangrong Liu[4], Xiangxiang Zeng[5], Zhaohong Deng[1,*]

[1]School of Artificial Intelligence and Computer Science, Jiangnan University, No. 1800 Lihu Avenue, Binhu District, Wuxi 214000, China
[2]School of Artificial Intelligence, Hebei University of Technology, 5340 Xiping Road, Beichen District, Tianjin 300130, China
[3]Department of Biochemistry and Molecular Biology and Biomedicine Discovery Institute, Monash University, Wellington Rd, Clayton VIC 3800, Australia
[4]Department of Computer Science and Technology, National Institute for Data Science in Health and Medicine, Xiamen Key Laboratory of Intelligent Storage and Computing, Xiamen University, 422 Siming South Road, Siming District, Xiamen City, Fujian 361005, China
[5]School of Information Science and Engineering, Hunan University, Yuelu District, Changsha 410012, China

[*]Corresponding authors. Yun Zuo, School of Artificial Intelligence and Computer Science, Jiangnan University, No. 1800 Lihu Avenue, Binhu District, Wuxi 214000, China. E-mail: zuoyun@jiangnan.edu.cn; Zhaohong Deng, School of Artificial Intelligence and Computer Science, Jiangnan University, No. 1800 Lihu Avenue, Binhu District, Wuxi 214000, China. E-mail: dengzhaohong@jiangnan.edu.cn

Abstract

Subcellular localization of messenger ribonucleic acid (mRNA) is a universal mechanism for precise and efficient control of the translation process. Although many computational methods have been constructed by researchers for predicting mRNA subcellular localization, very few of these computational methods have been designed to predict subcellular localization with multiple localization annotations, and their generalization performance could be improved.

In this study, the prediction model MSlocPRED was constructed to identify multi-label mRNA subcellular localization. First, the preprocessed Dataset 1 and Dataset 2 are transformed into the form of images. The proposed MDNDO–SMDU resampling technique is then used to balance the number of samples in each category in the training dataset. Finally, deep transfer learning was used to construct the predictive model MSlocPRED to identify subcellular localization for 16 classes (Dataset 1) and 18 classes (Dataset 2). The results of comparative tests of different resampling techniques show that the resampling technique proposed in this study is more effective in preprocessing for subcellular localization. The prediction results of the datasets constructed by intercepting different NC end (Both the 5' and 3' untranslated regions that flank the protein-coding sequence and influence mRNA function without encoding proteins themselves.) lengths show that for Dataset 1 and Dataset 2, the prediction performance is best when the NC end is intercepted by 35 nucleotides, respectively. The results of both independent testing and five-fold cross-validation comparisons with established prediction tools show that MSlocPRED is significantly better than established tools for identifying multi-label mRNA subcellular localization. Additionally, to understand how the MSlocPRED model works during the prediction process, SHapley Additive exPlanations was used to explain it. The predictive model and associated datasets are available on the following github: https://github.com/ZBYnb1/MSlocPRED/tree/main.

Keywords: subcellular localization; deep transfer learning; sequence analysis; interpretable analysis

Introduction

In the intricate process of eukaryotic development, the precise subcellular localization of messenger ribonucleic acid (mRNA) is a fundamental and essential regulatory mechanism, with profound implications for protein synthesis [1]. Eukaryotic cells exhibit a dynamic distribution of mRNA, often found in diverse cellular compartments, particularly in complex organisms where it orchestrates multiple functions [2–6]. mRNA localization is governed by intricate biological rules; yet, any disruption can trigger severe health consequences, including cancers, spinal muscular atrophy, Alzheimer's disease, and neurological disorders [7–10]. Consequently, a comprehensive understanding of mRNA's localization machinery is of paramount biological importance. However, given the time-consuming nature of traditional biochemical methods, there is a pressing need for the development of a computationally efficient and accurate predictive tool to streamline this process [11–16]. In recent years, an increasing number of machine learning algorithms have been employed to predict mRNAs and non-coding RNA's (ncRNAs') subcellular localization [17–20]. In 2021, Wang et al.'s method, DM3Loc, utilized one-hot encoding as input features and employed Convolutional Neural Network (CNN) and multi-head attention to predict multiple labels for six positions [21]. Li's team, also in 2021, presented SubLocEP, which took into account additional features and used a weighted aggregation of single-layer models [22]. In 2022, Bi et al. introduced Clarion that combined sequence information and prior label knowledge, using multiple binary classifiers to achieve multi-label predictions [23]. In 2023, Yuan et al. developed RNAlight, which assembled k-mers into sequence feature maps and targeted a more general approach [24]. In the same year, Wang et al. introduced DeepmRNALoc, employing a two-stage feature extraction strategy in a deep learning neural network [25].

Received: May 29, 2024. **Revised:** August 19, 2024. **Accepted:** September 30, 2024

Although a great deal of research work has been carried out and many computational tools have been constructed for subcellular localization prediction, there are still many limitations: (i) they are often designed for single-position prediction, while in reality, multiple positions may be relevant. (ii) Converting multi-position classification into multiple binary problems can be computationally expensive. (iii) The generalization and accuracy of current models are not optimal, and their feature extraction methods may be limited or unreliable. To address these limitations, a computational tool needs to be developed that can handle multi-position prediction more efficiently, consider more sophisticated feature extraction techniques, and improve overall performance and generalizability. With this in mind, this study aims to establish a multi-label computational tool that can be directly used to predict multi-label subcellular localizations. The flowchart of the prediction model MSlocPRED developed in this study is depicted in Fig. 1. Figure 1 mainly consists of the following parts: initially, Dataset 1 and Dataset 2 are divided into a training set and a testing set (data preprocessing) according to a 9:1 ratio. Subsequently, the multi-dimensional normal distribution–similarity of Mahalanobis distances (MDNDO–SMDU) resampling algorithm is employed to balance the quantity of each class. The balanced data are then converted into a format suitable for transfer learning, that is, all data from Dataset 1 and Dataset 2 are transformed into images using a Python script. This script iterates over text files in a specified directory, reads the text content line by line, and renders each line into separate images. The generated image has a transparent background and is cropped according to the size of the text to ensure a compact display effect. The image files are saved in the corresponding directories using the same naming conventions as the source text files. It should be noted that in this step, we only convert the NC-termini of the sequences (the first 35 and last 35 nucleotides). After completing the above steps, the data will be sent to the optimized AlexNet transfer learning model for training. Finally, once the network training is complete, the testing set is input into the predictor MSlocPRED, yielding test results and computing various metrics for multi-label classification to assess the performance of network.

Materials and methods

Benchmark dataset

It is crucial to build a reliable benchmark dataset in order to develop predictive models with good generalization performance and statistical significance. All mRNA subcellular localization datasets used in this study were sourced from the RNALocate database (a resource for RNA subcellular localization analysis). RNALocate v1.0 (version 1.0, updated in February 2020) [26] integrated GenBank (https://www.ncbi.nlm.nih.gov/genbank/) [27], and the mRNA sequence data in the FASTA format were obtained from the National Center for Biotechnology Information in February 2020 (https://www.ncbi.nlm.nih.gov/sites/batchentrez). RNALocate v2.0 (version 2.0, updated in June 2021) [28] integrated RNA subcellular localization data from five databases (CSCD) [29], EVmiRNA [30], exoRBase [31], PomBase [32] and TAIR [33]. To comprehensively validate the effectiveness of the predictive model constructed in this study, we constructed predictive models using two benchmark datasets collected from the RNALocate v1.0 and RNALocate v2.0 databases. The specific descriptions of the two benchmark datasets are as follows:

The benchmark dataset collected based on the RNALocate v1.0 [26], namely Dataset 1

Considering that Wang et al. [21] collected data on seven subcellular localizations [nucleus, exosome, cytosol, cytoplasm, ribosome, membrane, endoplasmic reticulum (ER)] based on the RNALocate v1.0 database in 2021, they constructed DM3Loc [21] to identify multiple subcellular localizations and compared its performance with that of four advanced methods. In order to objectively compare the predictive performance of the model constructed in this study, the benchmark dataset constructed by Wang et al. [21] was used as Dataset 1. The detailed steps for constructing training and testing datasets for Dataset 1 are as follows:

(i) For the seven subcellular localization data collected, since an mRNA can localize at multiple departments and we are considering seven departments, a total of 128 (2^7) possible combinations of mRNA localizations exist in theory, removing categories with less than 230 sequences and duplicate sequences. Finally, the dataset is divided into 16 categories:

$$\begin{cases} data_1^1 = data_1^1(0100000), data_2^1 = data_2^1(0100001) \\ data_3^1 = data_3^1(0100100), data_4^1 = data_4^1(0101000) \\ data_5^1 = data_5^1(1100000), data_6^1 = data_6^1(1100001) \\ data_7^1 = data_7^1(1100100), data_8^1 = data_8^1(1101000) \\ data_9^1 = data_9^1(1101010), data_{10}^1 = data_{10}^1(1101100) \\ data_{11}^1 = data_{11}^1(1101101), data_{12}^1 = data_{12}^1(1101110) \\ data_{13}^1 = data_{13}^1(1111000), data_{14}^1 = data_{14}^1(1111010) \\ data_{15}^1 = data_{15}^1(1111100), data_{16}^1 = data_{16}^1(1111110) \end{cases} \tag{1}$$

In which, one-hot was used to represent the localization categories, they are in the following order: nucleus, exosome, cytosol, cytoplasm, ribosome, membrane, and ER. For example, 0100000 means this sequence has the exosome annotation, and 0100001 means this sequence has the exosome and ER annotations. The meanings of the other 14 equations are similar. (ii) After division, the 16 categories of data were obtained. (iii) The 16 classes of data obtained were randomly divided into two parts: the training dataset and the testing dataset. Where 90% of the data in each class is used as training data, the rest of the data is used as test data. (iv) In order to test the importance of the N-terminal and C-terminal of subcellularly localized sequences, the N- and C-termini of the preprocessed sequences were intercepted to lengths of 20–45, interval of 5, that is, the sequence lengths are 40, 50, 60, 70, 80, and 90, respectively. Table 1 lists the specific quantities for each category in the training and testing sets of Dataset 1, along with the corresponding labels.

Based on the benchmark dataset collected by clarion [23] for messenger ribonucleic acid subcellular localizations from the RNALocate v2.0 [28] (Dataset 2)

For the nine subcellular localization data collected from Clarion (exosome, nucleus, nucleoplasm, chromatin, cytoplasm, nucleolus, cytosol, membrane, and ribosome), the detailed steps for constructing the training and test datasets are as follows:

(i) For the nine subcellular localization training and test data collected, since an mRNA can localize at multiple compartments and we are considering nine compartments, a total of 512(2^9) possible combinations of mRNA localizations exist in theory, the categories with less than 300 sequences and duplicated sequences were removed and finally the training and test datasets were

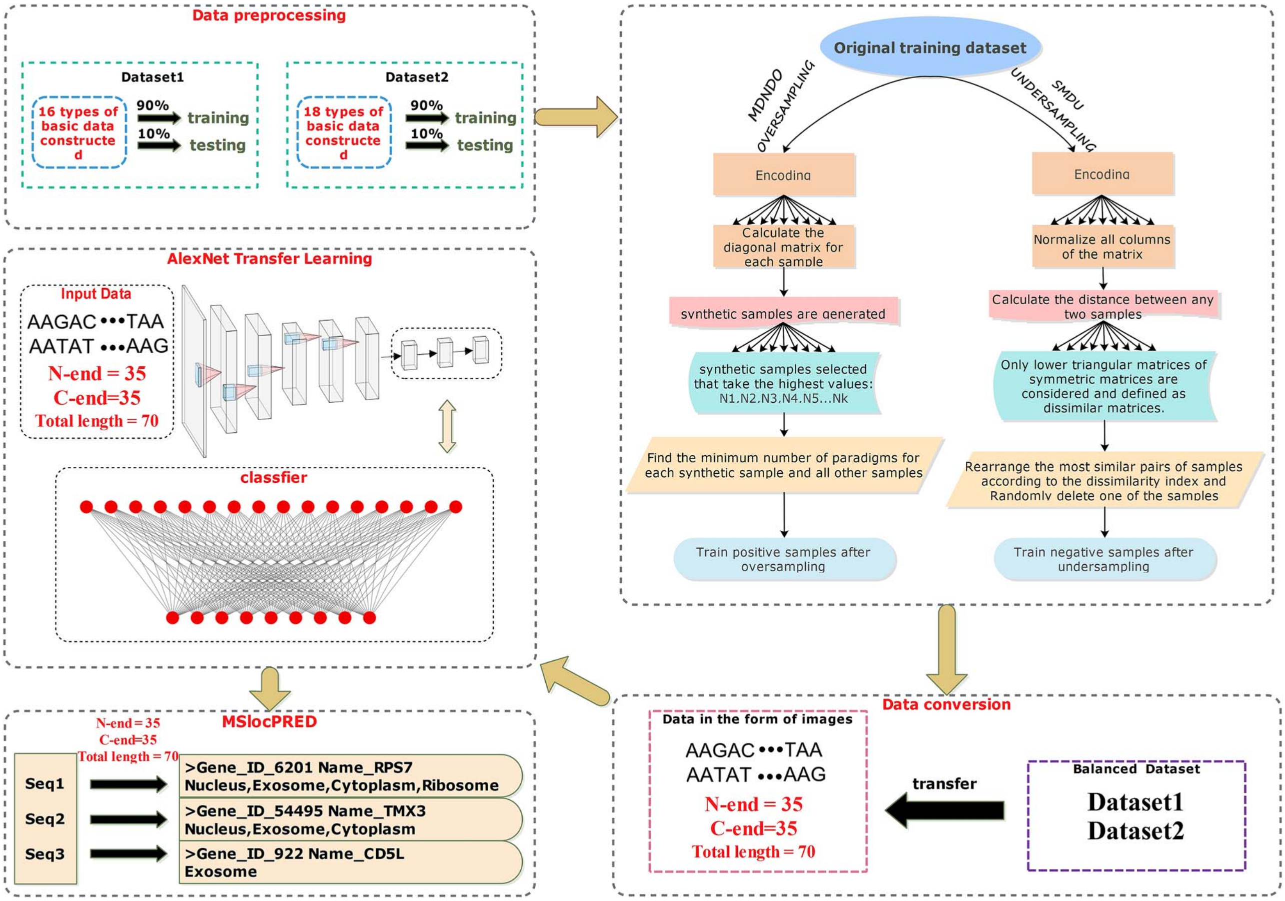
Data preprocessing
Dataset1
16 types of basic data constructe d
90%
training
10%
testing
Dataset2
18 types of basic data constructe d
90%
training
10%
testing
Original training dataset
MDNDO OVERSAMPLING
SMDU UNDERSAMPLING
Encoding
Calculate the diagonal matrix for each sample
synthetic samples are generated
synthetic samples selected that take the highest values: N1.N2.N3.N4.N5...Nk
Find the minimum number of paradigms for each synthetic sample and all other samples
Train positive samples after oversampling
Encoding
Normalize all columns of the matrix
Calculate the distance between any two samples
Only lower triangular matrices of symmetric matrices are considered and defined as dissimilar matrices.
Rearrange the most similar pairs of samples according to the dissimilarity index and Randomly delete one of the samples
Train negative samples after undersampling
AlexNet Transfer Learning
Input Data
AAGAC •••TAA
AATAT •••AAG
N-end = 35
C-end=35
Total length = 70
classfier
MSlocPRED
N-end=35
C-end=35
Total length=70
Seq1
Seq2
Seq3
>Gene_ID_6201 Name_RPS7
Nucleus,Exosome,Cytoplasm,Ribosome
>Gene_ID_54495 Name_TMX3
Nucleus,Exosome,Cytoplasm
>Gene_ID_922 Name_CD5L
Exosome
Data conversion
Data in the form of images
AAGAC •••TAA
AATAT •••AAG
N-end = 35
C-end=35
Total length = 70
transfer
Balanced Dataset
Dataset1
Dataset2

Figure 1. The framework diagram of the prediction model MSlocPRED constructed in this paper.

Table 1. Content of Dataset 1

Classes	Label	Total counts	Train set	Test set	Classes	Label	Total counts	Train set	Test set
1	0100000	3603	3243	360	9	1101010	1032	929	103
2	0100001	240	216	24	10	1101100	1677	1509	168
3	0100100	242	205	23	11	1101101	224	202	22
4	0101000	574	517	57	12	1101110	714	643	71
5	1100000	1662	1496	166	13	1111000	879	791	88
6	1100001	266	239	24	14	1111010	337	303	34
7	1100100	586	527	59	15	1111100	430	387	43
8	1101000	1909	1718	191	16	1111110	233	210	23

divided into 18 categories, respectively:

$$\begin{cases} data_1^2 = data_1^2(100000000), data_2^2 = data_2^2(010000000) \\ data_3^2 = data_3^2(000010000), data_4^2 = data_4^2(110000000) \\ data_5^2 = data_5^2(100100000), data_6^2 = data_6^2(100000100) \\ data_7^2 = data_7^2(100000001), data_8^2 = data_8^2(110010000) \\ data_9^2 = data_9^2(110000100), data_{10}^2 = data_{10}^2(110000110) \\ data_{11}^2 = data_{11}^2(111101000), data_{12}^2 = data_{12}^2(111100100) \\ data_{13}^2 = data_{13}^2(111101100), data_{14}^2 = data_{14}^2(111100110) \\ data_{15}^2 = data_{15}^2(111100101), data_{16}^2 = data_{16}^2(111101110) \\ data_{17}^2 = data_{17}^2(111101101), data_{18}^2 = data_{18}^2(111101111) \end{cases} \tag{2}$$

In which,we use one-hot encoding to represent the localization categories, they are in the following order: exosome, nucleus, nucleoplasm, chromatin, cytoplasm, nucleolus, cytosol, membrane, and ribosome. For example, 100000000 means this sequence has the exosome annotation; 100100000 means this sequence has the exosome and chromatin reticulum annotations; the meanings of the other 16 equations follow in this manner. (ii) The 18 class training datasets and testing datasets were obtained after division. (iii) In order to test the importance of the N-terminal and C-terminal of subcellular localized sequences, the N-terminal and C-terminal of the preprocessed sequences were intercepted to lengths of 20–50, interval of 5, that is, the sequence lengths are 40, 50, 60, 70, 80, 90, and 100, respectively. Table 2 lists the specific quantities for each category in the training and testing sets of Dataset 2, along with the corresponding labels.

Resampling methods

Due to the extreme imbalance of the training dataset constructed in this study, the ratio of 16 training samples for Dataset 1 is as follows: 3243: 216: 205: 517: 1496: 239: 527: 1718: 929: 1509: 202:

Table 2. Content of Dataset 2

Classes	Label	Total counts	Train set	Test set	Classes	Label	Total counts	Train set	Test set
1	100000000	7506	6774	732	10	110000110	393	349	44
2	010000000	3365	3037	328	11	111101000	394	359	35
3	000010000	1911	1715	196	12	111100100	580	527	53
4	110000000	1291	1154	137	13	111101100	1730	1557	173
5	100100000	353	319	34	14	111100110	497	448	49
6	100000100	1143	1016	127	15	111100101	493	445	48
7	100000001	475	422	53	16	111101110	1884	1688	196
8	110010000	392	345	47	17	111101101	1320	1190	130
9	110000100	647	587	60	18	111101111	1139	1034	105

643: 791: 303: 387: 210; the ratio of 18 training samples for Dataset 2 is as follows: 6774: 3037: 1715: 1154: 319: 1016: 422: 345: 587: 349: 359: 527: 1557: 448: 445: 1688: 1190: 1034. Therefore, this study proposed the MDNDO–SMDU resampling algorithm to reduce the ratio of training samples for Dataset 1 (16 categories) and Dataset 2 (18 categories). We calculate the number of various types of samples after sampling according to Equation (3):

$$N = round\left(\frac{n_1 + n_2 + \cdots + n_k + \cdots + n_t}{t}\right) \tag{3}$$

Among them, $n_k$represents the number of the k class samples, $k = 1, 2, 3, \cdots, t$. For Dataset 1, t=16; Dataset 2, t=18. For each class of training samples with a quantity less than N, use undersampling based on the similarity of Mahalanobis distances (SMDU) to remove similar or redundant samples, and an oversampling algorithm based on multi-dimensional normal distribution (MDNDO) to synthesize samples with the same distribution. The MDNDO–SMDU resampling algorithm proposed in this study is described as follows:

Multi-dimensional normal distribution oversampling algorithm

The MDNDO oversampling algorithm was proposed in our previous research for synthesizing multi-label lysine sequences. Given the effectiveness of the MDNDO oversampling algorithm in predicting post translational modification sites in proteins, this study utilizes the MDNDO oversampling algorithm to synthesize training samples for each class with fewer than N. The main idea of oversampling algorithm based on multi-dimensional normal distribution (MDNDO) used in this study is to synthesize the$M_k$samples that follow the same normal distribution, and the $M_k$computation formula is shown in Equation (4):

$$M_k = N - n_k, k = 1, 2, 3, \cdots, t \tag{4}$$

In which n_k is the sample number of the k class, $k = 1, 2, 3, \cdots, t$. The specific steps of the MDNDO oversampling technique are shown below:

The first step: for the truncated nucleotides at the NC end, convert the four nucleotides (A, C, G, T) into numerical vectors, namely:

$$A - 1, C - 2, G - 3, T - 4 \tag{5}$$

The second step: suppose $\mathbf{X}^k = \left[\mathbf{x}_1^k, \mathbf{x}_2^k, \cdots, \mathbf{x}_i^k, \cdots, \mathbf{x}_{n_k}^k\right]'$, $k = 1, 2, \cdots, t$ is all samples in the k class, where $\mathbf{x}_i^k$ is the i sample in the k class, n_k is the total number of samples in the k class, and d is the dimensionality. $\mathbf{XX} = \left[\mathbf{X}^1, \mathbf{X}^2, \cdots, \mathbf{X}^{k-1}, \mathbf{X}^{k+1}, \cdots, \mathbf{X}^t\right]'$ denotes all samples in the other classes except the k class, $\mathbf{X}^t = \left[\mathbf{x}_1^t, \mathbf{x}_2^t, \cdots, \mathbf{x}_i^t, \cdots, \mathbf{x}_{n_t}^t\right]'$ denotes all samples in the t class, n_t is the total number of samples in the t class. When the i sample in the k class is synthesized using MDNDO, $\mathbf{x}_i^k$ denotes the mean of the data to be generated, and σ denotes the autocorrelation matrix (correlation coefficient matrix) of x_i^k. Then ${\sigma_i}^2$ can be expressed in the form of equation (6):

$$ {\sigma_i}^2 = \begin{bmatrix} \beta\left(x_{i,1}^t\right)^2 & 0 & \cdots & 0 \\ 0 & \beta\left(x_{i,2}^t\right)^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \beta\left(x_{i,d}^t\right)^2 \end{bmatrix}_{d\times d} \tag{6}$$

Where the p row and q column element of ${\sigma_i}^2$ is denoted as:

$$ {\sigma_i}^2\,(p,q) = \begin{cases} 0 & , \ p \neq q \\ \beta\left(x_{i,p}^t\right)^2 & , \ p = q \end{cases}, \quad p, q = 1, 2, \cdots, d \tag{7}$$

β is the probability factor that is taken as 0.05 in this study.

The third step: Assuming that t_p samples are generated for each sample in the k class, for $n_k \times t_p$ synthetic samples generated from all samples in the k class, all synthetic samples are evaluated using indicator $\mathbf{I}$. The indicator for the k synthetic sample is:

$$\mathbf{I}_i = \min_{1\leq j\leq m} \sqrt{\left\| \mathbf{x}_i^k - \mathbf{x}_j^l \right\|^2}, i = 1, 2, \cdots, n_k \times t_p \tag{8}$$

Where $l = 1, 2, \cdots, k-1, k+1, \cdots, t$, $m = n1 + n2 + \cdots + k - 1 + k + 1 + \cdots + t$.The $n_k \times t_p$ metrics calculated from Equation (8) are rearranged in ascending order, so that the M_k synthetic samples selected are the M_k samples that take the highest values in $\mathbf{I}$ (i.e. those considered to be the furthest from the l class according to the metric defined in Equation (8)).

The forth step: Use the reverse numerical conversion of nucleotides to convert nucleotides represented by 1 to 4 to four kinds of nucleotides, namely:

$$1 - A, 2 - C, 3 - G, 4 - T \tag{9}$$

Finally, the training samples synthesized using MDNDO are combined with the original n_k training samples, and finally N training samples can be obtained for training the correlation

Table 3. The pseudo-code of MDNDO oversampling

```
Algorithm 1: MDNDO oversampling
Input : The$k$class of original training samples:$\mathbf{x}_1^k, \mathbf{x}_2^k, \cdots, \mathbf{x}_i^k, \cdots, \mathbf{x}_{n_k}^k, k = 1, 2, \cdots, t$, the number of samples$n_k$
Output : Synthesized$M_k$training samples$Q_1^k, Q_2^k, Q_3^k, \cdots, Q_{M_k}^k$
 1. Encoding training samples$\mathbf{x}_1^k, \mathbf{x}_2^k, \cdots, \mathbf{x}_i^k, \cdots, \mathbf{x}_{n_k}^k$ by using equation (5)
 2. For EACH$i$:$1 \le i \le n_k$ /*Calculate the diagonal matrix of covariance generated for each sample in the$k$class */
 3. DO $c = \mathbf{x}_i^k; b = size\,(c, 2)\,; z = zeros\,(b, b)\,; m = zeros\,(2{*}b, 1)\,;$
 4.     For EACH$j_1$:$1 \le j_1 \le size\,(c, 2)$
 5.        For EACH $j_2$:$1 \le j_2 \le size\,(c, 2)$
 6.        IF $j_1 == j_2$
 7.           DO$z\,(j_1, j_2) = 0.05{*}\,(c\,(j_1))\,\hat{}2; t\,(:, i) = \mathbf{diag}(z); t_1\,\{i\} = \mathbf{diag}\,(t\,(:, i))\,;$
 8.        END IF
 9.        END FOR
10.     END FOR
11. END FOR
12. For EACH$i$:$1 \le i \le n_k$ /*For each sample,$kp$synthetic samples are generated */
13. DO $p_1\,\{i\} = \left[mvnrnd\left(\mathbf{x}_i^k, sqrt\,(t_1\,\{1, i\})\,, kp\right)\right];$
14. END FOR
15. For EACH$i$:$1 \le i \le n_k$ /* Find the minimum number of paradigms for each synthetic sample and all other samples */
16.     For EACH$j_1$:$1 \le j_1 \le kp$
17.        For EACH$k_1$:$1 \le k_1 \le n_1 + n_2 + \cdots + n_{k-1} + n_{k+1} + \cdots + n_t$
18.           $nn_1\,(k_1) = sqrt\left(\mathbf{norm}\left(p_1\,\{1, i\}\,(j_1, :) - \mathbf{x}_{k_1}^l\right)\hat{}2\right);/{*}l = 1, 2, \cdots, k-1, k+1, \cdots, t{*}/$
19.        END FOR
20.           $m_1\,(i, j_1) = \mathbf{min}\,(nn_1)\,;$
21.     END FOR
22. END FOR
23. The$n_k \times k_p$metrics calculated from 15–17 are rearranged in ascending order
24. The$M_k$synthetic samples selected that take the highest values
25. RETURN $Q_1^k, Q_2^k, Q_3^k, \cdots, Q_{M_k}^k$
```

prediction model. The pseudo-code of MDNDO (i.e. Algorithm 1) is shown in Table 3.

Similarity of Mahalanobis distances undersampling algorithm

Similarity-based undersampling was firstly proposed by Cateni et al., and its basic idea is to calculate the Euclidean distance between any two samples in each class, select the top N pairs of samples with the smallest result from the obtained lower triangular matrix, and randomly select one sample from each pair to delete, thus achieving the goal of deleting similar samples. Given the effectiveness of the Similarity-based undersampling algorithm, in our previous research, we proposed Kmeans similarity-based undersampling to remove redundant and similar non-carbonylated samples.

Consider two flaws that exist in the Euclidean distance: (i) it does not consider that different variables (dimensions) vary on different scales. For example, y_1 and y_2 represent lengths, and the difference between using 'centimeters' as the unit of measure and using 'meters' as the unit of measure is very large. They are really the same value, it is just the difference in units that causes the results of the Euclidean distance calculations to vary dramatically. (ii) The correlation between the variables was not considered. If the correlation between two variables (dimensions) is very strong, the Euclidean distance does not capture the correlation. Whereas Mahalanobis distance with the help of the idea of normalization of the unitary case, solving for the distance adds the inverse of the covariance matrix of y_1, y_2, so that variables (dimensions) with greater variance correspond to smaller weights, and the contribution of two highly correlated variables (dimensions) to the Mahalanobis distance is smaller than the contribution of two variables with relatively low correlation. In view of this, in this study, we calculate the Mahalanobis distance between any two samples in each class, select the top N_g pairs of samples with the smallest distance, and randomly select one sample from each pair to delete. The detailed steps are as follows:

The first step: suppose that $\mathbf{x}^g = \mathbf{x}_1^g, \mathbf{x}_2^g, \cdots, \mathbf{x}_i^g, \cdots, \mathbf{x}_{n_g}^g$ represent all training samples in the g class, in which $\mathbf{x}^g, g = 1, 2, \cdots, t$, d is the dimensionality of the extracted features, and n_g is the number of all samples in the g class. Normalize all columns of the matrix $\mathbf{x}^g$ and get a transformed matrix w, where the element $w_{i,h}$ of the i row and h column can be represented as:

$$w_{i,h} = \frac{\mathbf{x}_{i,h}^g}{\max_{1 \le j \le n_g}\left\{\mathbf{x}_{j,h}^g\right\}}, i = 1, 2, \cdots, n_g, h = 1, 2, \cdots, d \qquad (10)$$

The second step: compute the Mahalanobis distance between every two rows of the transformed matrix w to get a symmetrical square distance matrix D_1, where the element $d\,(p, q)$ of the matrix D_1 can be represented as:

$$d\,(p, q) = \sqrt{(w^p - w^q)\,S^{-1}(w^p - w^q)'} \qquad (11)$$

Where w^p and w^q represented vectors consisting of all elements of the p and q rows of the matrix w, respectively. S^{-1} is the inverse of the covariance of w^p, w^q.

The third step: it was clear that the elements located on the main diagonal of the symmetrical square distance matrix D_1 are zero. Because the matrix D_1 was a symmetry matrix, only the lower triangle of the matrix D_1 was considered in the below, and D_1 was defined as a dissimilarity matrix.

The fourth step: The smaller the element $d\,(p, q)$, the more 'similar' the samples w^p and w^q. The pairs of samples were rearranged based on this similarity index. For the most similar

Table 4. The pseudo-code of SMDU undersampling

```
Algorithm 2: SMDU undersampling
Input : The$g$class of original training samples:$\mathbf{x}^g = \mathbf{x}_1^g, \mathbf{x}_2^g, \cdots, \mathbf{x}_i^g, \cdots, \mathbf{x}_{n_g}^g, g = 1, 2, \cdots, t$, the number of samples$n_g$
Output: After undersampling$N$training samples$Q_1^g, Q_2^g, Q_3^g, \cdots, Q_N^g$
Encoding training samples$\mathbf{x}_1^g, \mathbf{x}_2^g, \cdots, \mathbf{x}_i^g, \cdots, \mathbf{x}_{n_g}^g$by using equation (5)
For EACH$i$:$1 \le i \le n_g$ /* Normalize all columns of the matrix$\mathbf{x}^g \in \mathbb{R}^{n \times d}$and get a transformed matrix$w$*/
  For EACH$k$:$1 \le k \le size\left(\mathbf{x}^g, 2\right)$
      $a\left(i, k\right) = \max\left(\mathbf{x}^g\left(:, k\right)\right);$
    IF $a\left(i, k\right) == 0$
      $w\left(i, k\right) = \mathbf{x}^g\left(i, k\right) / n_g$
    ELSE
      $w\left(i, k\right) = \mathbf{x}^g\left(i, k\right) / \max\left(\mathbf{x}^g\left(:, k\right)\right)$
    END IF
  END FOR
END FOR
/*Calculate the distance between any two samples*/
$D_1 = pdist\left(w, \text{"}mahal\text{"}\right); D = squareform\left(D_1\right);$
For EACH$i$:$1 \le i \le n_g$
  For EACH$j$:$1 \le j \le n_g$
    IF$j >= i$
      $D\left(i, j\right) = 0;$
    END IF
  END FOR
END FOR
$D_1 = nonzeros(D);$
$D_2 = sort\left(D_1, 'ascend'\right);$
Select the$N_g$most similar couples of samples
Eliminate one sample from the$N_g$most similar couples
Obtain$N = n_g - N_g$training samples$Q_1^g, Q_2^g, Q_3^g, \cdots, Q_N^g$after undersampling
RETURN $Q_1^g, Q_2^g, Q_3^g, \cdots, Q_N^g$
```

pairs, we randomly selected one sample to eliminate, thus preserving the original class distribution without significant loss of information. The pseudo-code of SMDU (i.e. Algorithm 2) is shown in Table 4.

AlexNet transfer learning

Transfer learning allows models developed for one task to be reused as a starting point for models for another task, and saves the significant computational and time resources needed to train neural networks. In this study, the AlexNet model trained at the Computer Vision Challenge via the ImageNet dataset was used to transfer the application to the subcellular localization dataset for retraining (fine-tuning). The major steps are shown below:

(1) Each sample from the training and test data is first transformed into the form of an image and then fed into the input layer of the AlexNet network.

(2) Then load the trained network and resize the dataset image to the same size as the network. The newly loaded data is not needed for a 1000-category classification task, so the last three layers of AlexNet must be targeted for readjustment to the new classification problem: (i) extract all layers except the last three; (ii) the extracted layers are transferred to the new task and the last three original layers are replaced with a fully connected layer, a soft max layer, and a classification output layer; (iii) Configure the new fully-connected layer parameters based on our new data as fullyConnectedLayer (class, ‘WeightLearnRateFactor’, 20, ‘BiasLearnRateFactor’, 20). (iv) Set the hyperparameters for model training respectively as ops = trainingOptions (‘sgdm’, ‘InitialLearnRate’, 0.0001, ‘ValidationData’, augimdsTest, ‘Plots’, ‘training-progress’, ‘MiniBatchSize’, 4, ‘MaxEpochs’, 3, ‘ValidationPatience’, Inf, ‘Verbose’, false);

(3) Initialize the weights of the output layer to random values, but keep the weights of the other layers the same as the originally trained weights.

(4) Start training on the subcellular localization dataset.

(5) Select the optimal model based on five-fold cross-validation and five evaluation indicators (Aiming, Coverage, Accuracy, Absolute_True, Absolute_False).

After modifying the input and output terminals, the final network structure diagram of AlexNet is illustrated in Fig. 2. Figure 2a depicts the modified input end, Fig. 2b shows the overall architecture of AlexNet used for transfer learning, and Fig. 2c presents the classifier modified for use in this study, where the input of the classifier should correspond to the number of features, and the output should be the number of classes. To ensure the reproducibility of the experimental results, this study provides a detailed description of the hardware and software environments used during the experimental process, as well as the experimental settings. The hardware configuration includes an Intel Gold 6226R processor and an NVIDIA RTX 4090 GPU equipped with 24GB of video memory, which are used to handle large datasets and complex computational demands. All experiments were conducted in the MATLAB environment, utilizing its capabilities for matrix computation and visualization to perform data analysis and model experiments. Experimental parameter settings included setting the number of iterations to 5490 for Dataset 1 and 9639 for Dataset 2, and a learning rate of 0.0001, to ensure that the model adequately learns the data features and achieves stable training performance.

In order to select the optimal transfer learning model, this study also compared AlexNet with several traditional neural networks used in the field of computer vision (VGG16, GoogLeNet, ResNet-50) on two datasets, with the comparative results

Table 5. Comparison of transfer learning effect applied to different networks on Dataset 1

Name	Aiming	Coverage	Accuracy	Absolute_True	Absolute_False
VGG16	0.6910	0.7420	0.5433	0.0901	0.3164
GoogLeNet	**0.8033**	0.6055	0.5006	**0.1226**	0.3186
ResNet-50	0.7467	0.6910	0.5343	0.1137	0.3121
AlexNet	0.7378	**0.7611**	**0.5676**	0.1120	**0.2935**

Table 6. Comparison of transfer learning effect applied to different networks on Dataset 2

Name	Aiming	Coverage	Accuracy	Absolute_True	Absolute_False
VGG16	**0.6963**	0.4373	0.3573	0.1059	0.3543
GoogLeNet	0.6652	0.6043	0.4445	0.1156	0.3328
ResNet-50	0.6381	0.6042	0.4470	**0.1393**	0.3315
AlexNet	0.6636	**0.6586**	**0.4756**	0.1151	**0.3163**

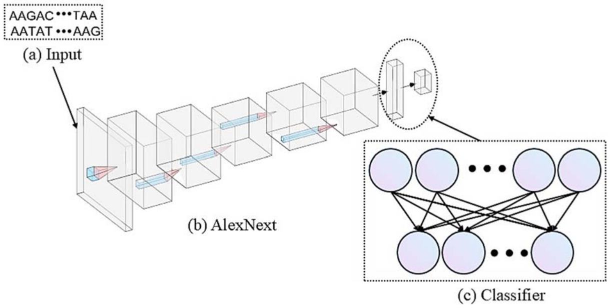

Figure 2. Transfer learning network.

presented in Table 5 and Table 6. For Dataset 1, AlexNet exhibited superior performance in terms of Accuracy (0.5676) and Coverage (0.7611), and its Absolute False (0.2935) was the lowest among the four networks, demonstrating its superior learning capabilities and good generalization performance. On Dataset 2, despite a general increase in error rates across all networks, AlexNet maintains the lowest Absolute_False (0.3163), while continuing to lead in Accuracy (0.4756) and Coverage (0.6586), further demonstrating its stability and adaptability across different settings. Moreover, although GoogLeNet achieved the highest Aiming (0.8033) in Dataset 1 and ResNet-50 exhibited the best Absolute_True value (0.1393) in Dataset 2. However, in terms of overall performance, AlexNet displays a more balanced and consistent performance across both datasets. These results indicate that AlexNet not only maintains high Accuracy and Coverage, but also effectively controls Absolute_False, making it the preferred network architecture for performing transfer learning.

Five kinds of multi-label evaluation indicators

For the two benchmark datasets collected and processed in this study (Dataset 1, Dataset 2), Dataset 1: we have a total of 13 135 training samples, the 7 subcellular localization data were specifically divided into 16 categories, of which 3243 are labelled with exosome annotation, the other 15 categories have two or more labels. Dataset 2: there are 22 966 training samples, the nine subcellular localization data were specifically divided into 18 categories, among them, 6774 with exosome annotation; 3037 with nucleus, 1715 with cytoplasm, the other 15 categories have two or more labels. Therefore, in the current study we are dealing with a multi label system according to Chou's formula [34], in order to evaluate the predictive performance of MSlocPRED, the evaluation criterion for a multi-label system can be defined as follows:

$$\text{Aiming} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\|}{\left\|\mathbb{L}_i^*\right\|}\right) \tag{12}$$

$$\text{Coverage} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\|}{\left\|\mathbb{L}_i\right\|}\right) \tag{13}$$

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\|}{\left\|\mathbb{L}_i \cup \mathbb{L}_i^*\right\|}\right) \tag{14}$$

$$\text{Absolute} - \text{True} = \frac{1}{n}\sum_{i=1}^{n}\Delta\left(\mathbb{L}_i, \mathbb{L}_i^*\right) \tag{15}$$

$$\text{Absolute} - \text{False} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\left\|\mathbb{L}_i \cup \mathbb{L}_i^*\right\| - \left\|\mathbb{L}_i \cap \mathbb{L}_i^*\right\|}{M}\right) \tag{16}$$

where n is the total number of samples, M is the total number of labels in the system, ∪ and ∩ denote 'union' and 'intersection' in set theory, $\|\|$ is the operator that operates on one of the sets to calculate the number of elements, $\Delta\,()$ denotes an operator that operates on a subset of them to determine whether all their subset elements are equal. $\mathbb{L}_i$ is the subset of all labels observed experimentally for the i sample, $\mathbb{L}_i^*$ is the subset of all predicted labels for the i sample, and

$$\sum_{i=1}^{n}\Delta\left(\mathbb{L}_i, \mathbb{L}_i^*\right) = \begin{cases} 1, & \textit{all labels in } \mathbb{L}_i \textit{ are the same as} \\ & \textit{the corresponding labels in } \mathbb{L}_i \\ 0, & \textit{otherwise} \end{cases} \tag{17}$$

In a multi-label system, (i) 'Aiming' or 'Precision' denotes the average ratio of predicted labels agreeing with true labels; (ii) 'Coverage' or 'Recall' indicates the average ratio of true labels covered by predicted labels; (iii) 'Accuracy' indicates the average ratio of correctly predicted labels to the total number of labels (including correctly and incorrectly predicted labels, as well as those true labels that were omitted during the prediction process); (iv) 'Absolute-True' or 'Subset-Accuracy' indicates the average ratio of predicted labels that are exactly the same as the true labels; and (v) 'Absolute-False' or 'Hamming loss' indicates the average ratio of the inconsistency between the predicted label and the true label to the total number of categories and samples. Obviously, in a multi-label system, when the values of Aiming, Coverage, Accuracy and Absolute-True are higher, and the value of Absolute-False is lower, the performance of the constructed model is better.

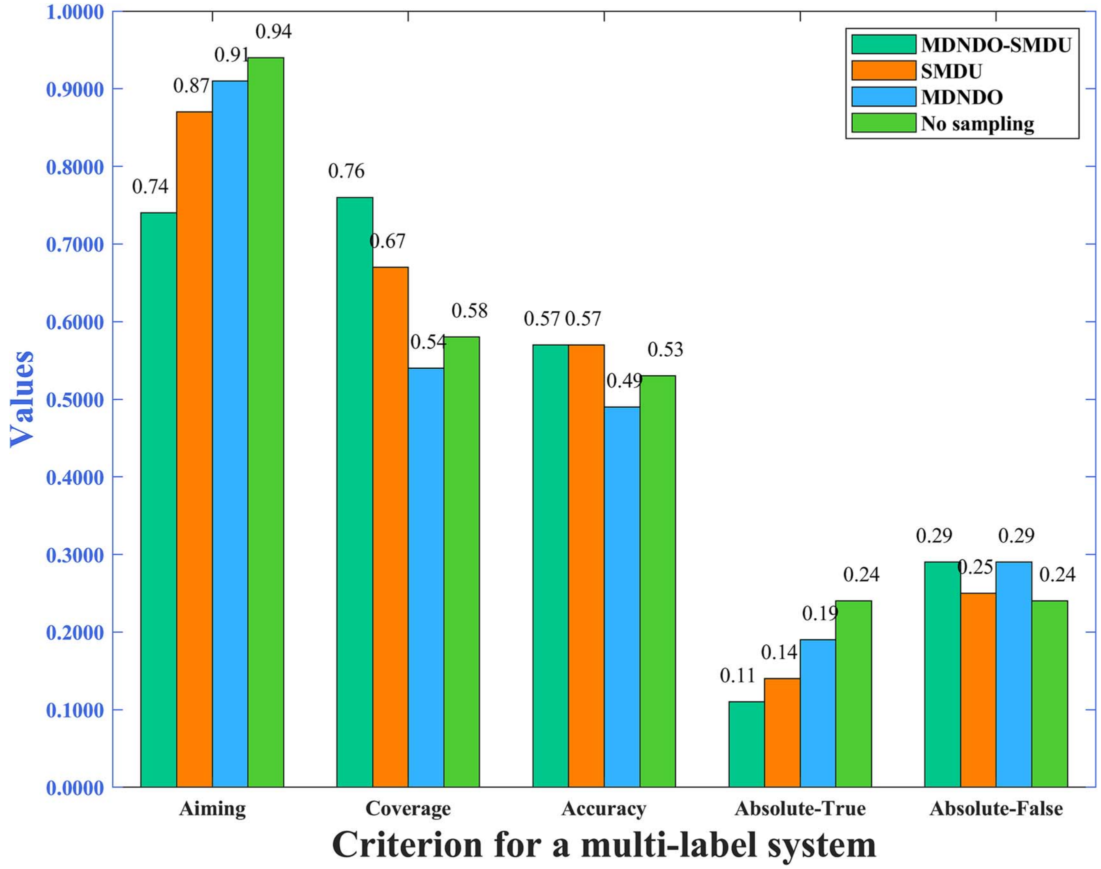

Figure 3. Five-fold cross-validation results of different resampling methods for Dataset 1.

Results and discussion

Preprocessing training data based on multi-dimensional normal distribution–similarity of Mahalanobis distances resampling technique

Since both training datasets constructed in this study are extremely unbalanced, for Dataset 1, the ratio of 16 classes of training samples is 3243: 216: 205: 517: 1496: 239: 527: 1718: 929: 1509: 202: 643: 791: 303: 387: 210. The number of training data for classes 2, 3, 4, 6, 7, 11, 12, 13, 14, 15, and 16 are less than 800 ($train_2^1$: 216, $train_3^1$: 205, $train_4^1$: 517, $train_6^1$: 239, $train_7^1$: 527, $train_{11}^1$: 202, $train_{12}^1$: 643, $train_{13}^1$: 791, $train_{14}^1$: 303, $train_{15}^1$: 387, $train_{16}^1$: 210), therefore, this study synthesized$N$samples using the algorithm proposed in a previous study on MDNDO for these 11 classes of training data. Assuming n_2, n_3, n_4, n_6, n_7, n_{11}, n_{12}, n_{13}, n_{14}, n_{15} and n_{16} are the number of samples for $train_2^1$, $train_3^1$, $train_4^1$, $train_6^1$, $train_7^1$, $train_{11}^1$, $train_{12}^1$, $train_{13}^1$, $train_{14}^1$, $train_{15}^1$ and $train_{16}^1$, respectively, the number of samples generated for these 11 classes is $N = M - n_i, i \in \{2, 3, 4, 6, 7, 11, 12, 13, 14, 15, 16\}$,

where $M = round\left(\frac{n_1+n_2+n_3+\cdots+n_{16}}{16}\right)$ is the number of training samples for each class after sampling, and $round\,()$ represents rounding. In this study, five-fold cross validation was used to train prediction models. For classes 2, 3, 4, 6, 7, 11, 12, 13, 14, 15, and 16 training samples, $round\left(\frac{4}{5}M\right)$ samples were used as training data for each fold in five-fold cross validation, $round\left(\frac{1}{5}M\right)$ samples were used as test data, and $round\,()$ represents rounding. For the training data of categories 1, 5, 8, 9, and 10, since their numbers are 3243, 1496, 1718, 929, and 1509, respectively, in contrast to the other 11 classes of training data, which are much larger in number, the SMDU under-sampling algorithm was used to calculate the Mahalanobis distance of any two samples in each category (categories 1, 5, 8, 9, and 10), select the$N_1 = n_i - M, i \in \{1, 5, 8, 9, 10\}$pairs of samples that have the smallest distance (i.e. the most similar pair of samples), and randomly delete one of them to achieve the purpose of removing the most similar samples.

For Dataset 2, since the number of training data for classes 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 17, and 18 are less than 1200, $N_i = M - n_i, i \in \{4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 15, 17, 18\}$ samples are synthesized for these 13 classes of training data using the MDNDO over-sampling algorithm,

where $M = round\left(\frac{n_1+n_2+n_3+\cdots+n_{18}}{18}\right)$ is the number of training samples for each class after sampling, and $round\,()$ denotes rounding. For the remaining 5 classes, the SMDU under-sampling algorithm is utilized to select the $N_i = n_i - M, i \in \{1, 2, 3, 13, 16\}$ pairs of samples with the smallest distance (i.e. the most similar pairs of samples) and randomly deletes one of them, thus achieving the purpose of deleting the most similar samples.

Effectiveness of multi-dimensional normal distribution–similarity of Mahalanobis distances resampling technique

In order to objectively evaluate the performance of the prediction model MSlocPRED constructed in this study, the MDNDO–SMDU resampling technique is compared with the training dataset constructed by three methods, namely, utilizing only MDNDO over-sampling, utilizing only the SMDU under-sampling algorithm, and no sampling. The results of the five-fold cross-validation correlation are shown in Figs 3 and 4.

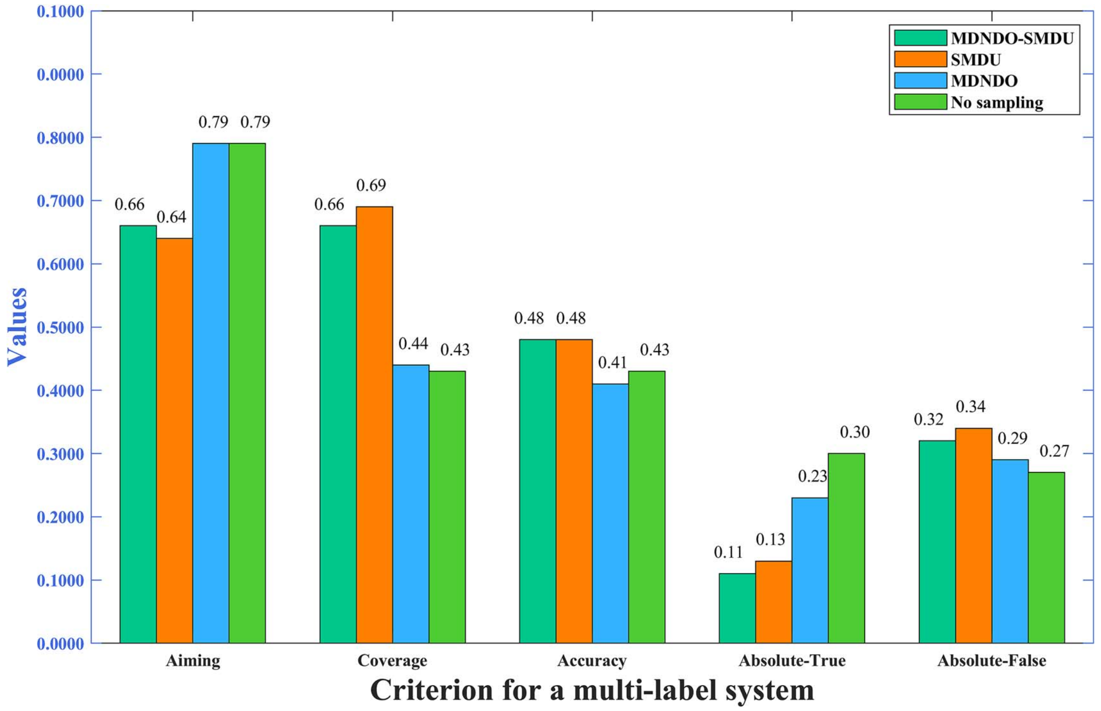

Figure 4. Five-fold cross-validation results of different resampling methods for Dataset 2.

From the five evaluation metrics and the specific prediction labels in Figs 3 and Fig. 4, it can be observed that when no sampling is performed (i.e. the original training dataset), due to the extreme difference in the number of samples between the large and small classes, the results of the trained model predictions are skewed towards the large class (take the first fold as an example, for Dataset 1: all were predicted to be Category 1, except for 20 were predicted to be Category 8 for the 2619 test sets. For Dataset 2: 4390 out of 4589 test data were predicted to be class 1). Although the predictors Aiming, Coverage, Accuracy, Absolute-True, and Absolute-False reached 0.94, 0.58, 0.53, 0.24, 0.24 and 0.79, 0.43, 0.43, 0.30, 0.27 for the two datasets, respectively, this model is an invalid model. When using only SMDU under-sampling, it is obvious that due to the imbalance of samples across different categories, most categories (especially those with a small number) were not tested. When only using MDNDO over-sampling to synthesize subcellular localization data, it has a certain effect. When using MDNDO–SMDU for resampling, it can be found that among categories containing two or more positions, most categories only have one position undetected.

After testing two training datasets (Dataset 1 and Dataset 2) using different sampling methods, the results show significant differences in category prediction. In order to clearly present these results, using the first fold of the five-fold cross-validation as an example, two tables (Tables 7 and 8) summarize the performance of different sampling methods on various categories.

Table 7 displays the test results on Dataset 1 using three sampling methods (MDNDO, SMDU, and MDNDO–SMDU). It is evident from the table that the MDNDO sampling method achieved the highest number of correct predictions for Category 1, totaling 2126, while this number significantly decreased to 381 when using the SMDU method. For other categories, the MDNDO–SMDU method showed notable advantages in certain categories, such as Category 10 with 1325 correct predictions, while the SMDU method performed better in Category 13 with 580 correct predictions. It is particularly noteworthy that the application of MDNDO could predict five categories, SMDU predicted three categories, and MDNDO–SMDU could predict up to eleven categories. This indicates that the MDNDO–SMDU sampling method not only learns features more comprehensively and completely but also significantly increases the number of category predictions, which is of significant importance for practical applications.

Table 8 shows the test results on Dataset 2. Similar to Dataset 1, the MDNDO sampling method performed exceptionally in Category 1 with 5134 correct predictions, while the SMDU method had a relatively high number of correct predictions in Category 13, totaling 1422. The MDNDO–SMDU method had the highest number of correct predictions in Category 13, amounting to 2045, and also showed excellent performance in Category 5 with 906 correct predictions. Additionally, the MDNDO–SMDU sampling method could predict 12 categories on Dataset 2, while the MDNDO and SMDU methods could only predict 7 and 5 categories, respectively. The application of the MDNDO–SMDU method again confirms its significant advantages in comprehensively and completely learning data features, playing a crucial role in enhancing classification performance.

It should be noted that for multi-label classification tasks, if the model can effectively predict some correct labels but not completely predict all labels correctly, the model is actually effective. Therefore, judging the effectiveness of a model based solely on average accuracy and complete correctness is neither scientific nor fair. Consequently, we have proposed a set of metrics to assess multi-label classification systems: Partial label accuracy MR_j.

$$MR_j = \frac{\sum_{k=j}^{\max} P_k}{\sum_{k=j}^{\max} C_k} \tag{18}$$

In this definition, P_k represents the total number of predicted samples at computation level k, where the true labels match the

Table 7. Different sampling algorithms predict the number of classes in the first fold of the five-fold cross-validation on Dataset 1

Class	MDNDO	SMDU	MDNDO–SMDU	Class	MDNDO	SMDU	MDNDO–SMDU
1	**2126**	381	33	9	0	0	4
2	0	0	**342**	10	20	703	**1325**
3	111	0	**348**	11	0	0	**83**
4	0	0	**4**	12	0	0	0
5	0	0	0	13	0	**580**	0
6	**464**	0	0	14	0	0	**53**
7	0	0	**18**	15	0	0	**4**
8	**851**	0	403	16	0	0	0

Table 8. Different sampling algorithms predict the number of classes in the first fold of the five-fold cross-validation on Dataset 2

Class	MDNDO	SMDU	MDNDO–SMDU	Class	MDNDO	SMDU	MDNDO–SMDU
1	**5134**	784	407	10	50	0	**121**
2	**106**	5	1	11	**339**	0	103
3	100	112	**130**	12	0	0	0
4	0	0	0	13	0	1422	**2045**
5	0	0	**906**	14	0	0	0
6	0	0	**125**	15	0	0	**441**
7	**482**	0	217	16	0	0	65
8	0	0	**29**	17	0	0	0
9	55	0	0	18	0	**591**	0

Table 9. The influence of different sampling algorithms on prediction results on Dataset 1

Method	MR_1	MR_2	MR_3	MR_4
None	100.00%	24.66%	24.49%	6.58%
MDNDO	100.00%	30.74%	17.57%	4.87%
SMDU	100.00%	67.83%	65.34%	20.88%
MDNDO–SMDU	100.00%	**86.40%**	**74.97%**	**44.89%**

Table 10. The influence of different sampling algorithms on prediction results on Dataset 2

Method	MR_1	MR_2	MR_3	MR_4
None	93.17%	0.00%	0.00%	0.00%
MDNDO	95.14%	16.6%	10.38%	5.54%
SMDU	94.57%	73.62%	**74.48%**	**75.09%**
MDNDO–SMDU	**96.57%**	**81.54%**	66.67%	68.68%

predicted labels, and the number of correctly predicted labels (considering only labels marked as '1' among all true labels) is exactly k. Meanwhile, C_k denotes the total number of samples at the same level k where the true label count equals k (also considering only labels marked as '1' among all true labels). The value of $\mathbf{max}$ is the highest number of labels marked as '1' occurrences within the sample categories in the current dataset.

For example, when calculating the data for Dataset 1 at level $k=4$, with the value of $\mathbf{max}$ is 6 (i.e. the sample in Dataset 1 with the most label marked as '1' is: 1111110), the numerator of interest would be $P_4 + P_5 + P_6$, where P_4 is the total number of samples where the predicted labels match the true labels and the number of matching labels is exactly 4; P_5 and P_6 are defined similarly. The corresponding denominator would be $C_4 + C_5 + C_6$, where C_4 is the total number of samples in the true dataset with exactly 4 labels marked as '1'. C_5 and C_6 are defined similarly. The numerator encompasses all samples with the following labels: 1101010, 1101100, 1101101, 1101110, 1111000, 1111010, 1111100, and 1111110. Notably, for samples with more than 4 labels marked as '1' (such as 1101101, 1101110, 1111010, 1111100, 1111110), only the cases where exactly 4 labels match are considered. The corresponding denominator includes the total number of samples with the true labels: 1101010, 1101100, 1101101, 1101110, 1111000, 1111010, 1111100, and 1111110. This can then be substituted into the formula as follows: $MR_4 = \frac{P_4+P_5+P_6}{C_4+C_5+C_6}$.

By applying this method, the partial label matching rate MR_j can be effectively calculated, providing insights into the performance of classification models in scenarios with imbalanced datasets. In this study, four indicators MR_1, MR_2, MR_3, MR_4 are used to evaluate the effectiveness of the multi-label classification model before and after the application of the sampling methods. Where MR_1 denotes the probability of at least one label being correctly predicted, that is, statistical analysis is conducted for categories containing one or more locations and so on for others.

The results of the five-fold cross validation calculation are shown in Tables 9 and 10, and the MDNDO–SMDU resampling algorithm demonstrates significant superiority in multiple key indicators. On Dataset 1, an MR_1 of 100% was achieved by all sampling algorithms, indicating that each method was capable of predicting at least one correct label. However, MDNDO–SMDU outperformed other sampling methods with MR_2, MR_3, and MR_4 scores of 86.40%, 74.97%, and 44.89%, respectively. It is worth noting that SMDU follows closely behind with scores of 67.83%, 65.34% and 20.88% for MR_2, MR_3, and MR_4, respectively. These results demonstrate the significant advantages of MDNDO–SMDU in enhancing data balance and improving the robustness of the classification model. On Dataset 2, MDNDO–SMDU also exhibited superior performance, achieving the best scores for MR_1 and MR_2 at 96.57% and 81.54%, respectively. In terms of MR_3 and MR_4, SMDU held a slight advantage, reaching 74.48% and 75.09%, respectively. Therefore, the MDNDO–SMDU resampling algorithm demonstrated a significant overall advantage in handling

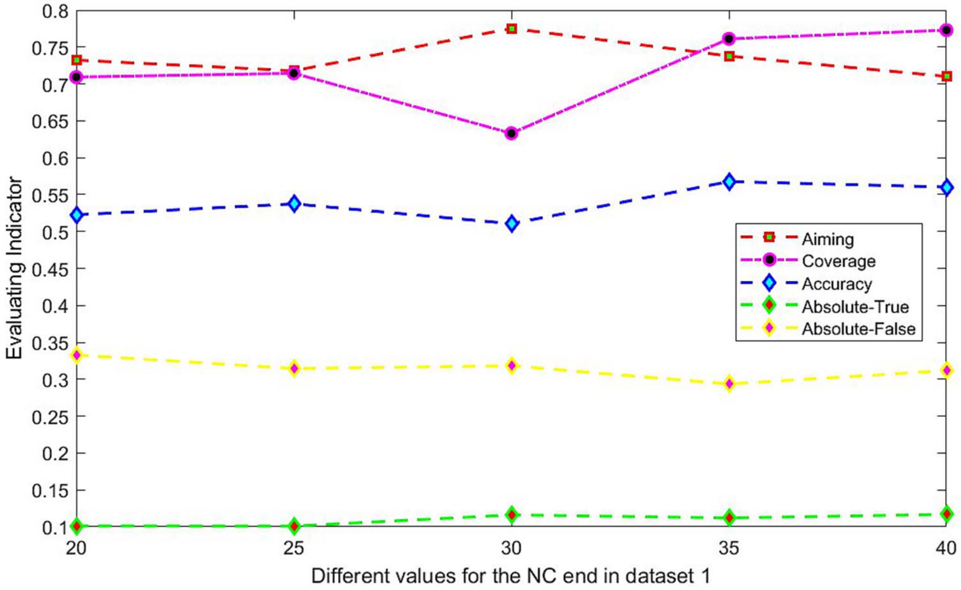

Figure 5. Prediction results of intercepted sequences with different values at the NC end on Dataset 1.

complex and imbalanced datasets. This method not only excelled in classification accuracy but also achieved comprehensive improvements across various evaluation metrics such as partial label matching rates. MDNDO–SMDU provides a more effective approach to addressing complex classification tasks by effectively combining the advantages of oversampling and undersampling, showcasing its potential value in learning data features.

Comparison of predictive performance between datasets constructed with different lengths on N-terminal and C-terminal

In molecular biology, the N-terminus (5′ end) and C-terminus (3′ end) of mRNA are considered critical regions for regulating mRNA function. These termini often contain various regulatory elements, such as protein-binding sites and stability control elements, which play crucial roles in mRNA subcellular localization and translation efficiency. Notably, the 5′ untranslated region (N-UTR) and the 3′ untranslated region (C-UTR) of mRNA have been found to contain regulatory elements that interact with specific proteins within the cell, thereby guiding mRNA transport to specific cellular regions [35, 36].

To further emphasize the importance of the N-terminus (5′ end) and C-terminus (3′ end) of mRNA in predicting subcellular localization, several relevant studies have been reviewed. Yan et al. identified that the N-terminal and C-terminal regions of RNA sequences exhibit significant biological activity [37]. Meer et al. (2012) [38] and Bergalet et al. [39] have also emphasized the importance of the N-terminal and C-terminal untranslated regions (N-UTR and C-UTR), pointing out that regulatory elements within these regions have a significant impact on mRNA subcellular localization. The study of DM3Loc further corroborates these findings, employing a multi-head self-attention mechanism to predict mRNA localization across various subcellular regions. The results demonstrated that DM3Loc outperforms existing methods in overall performance and provides biological insights into RNA-binding protein motifs and key signals [21]. These studies collectively support our approach of analyzing the N-terminus and C-terminus of mRNA, facilitating a more comprehensive understanding of its behavior and function within the cell.

To elucidate whether subcellular localized sequences of N-terminal and C-terminal nucleotides affect the predictive performance of the model, for subcellular localization sequences in the training dataset, we intercepted their N-terminal and C-terminal nucleotides from 20 to 40 for Dataset 1 and from 20 to 50 for Dataset 2 (i.e. the length of each sample after interception ranged from 40 to 80 for Dataset 1 and from 40 to 100 for Dataset 2), and the interval is 5. Figure 5 and Fig. 6 give the results of the five-fold cross-validation performance comparison for different NC end-value intercept sequences. Based on the comparison results we can see that the model constructed using the dataset constructed with 35 nucleotides intercepted from each of the N and C terminals achieved the best prediction performance with 73.78%, 76.11%, 56.76%, 11.20%, and 0.2935 for Aiming, Coverage, Accuracy, Absolute-False, and Absolute-True for Dataset 1, and 66.36%, 65.86%, 47.56%, 11.51%, and 0.3163 for Aiming, Coverage, Accuracy, Absolute-False, and Absolute-True for Dataset 2. From the comparison results, it can be concluded that the predictor was influenced to a certain extent by the length of the sequence.

Ultimately, to ensure that our model achieves optimal predictive performance, we extracted sequences of length 35 from both the N-terminus and C-terminus (resulting in a total length of 70) from Dataset 1 and Dataset 2 as input for MSlocPRED during training and testing. It is important to note that after the model is trained, the predictor must also be provided with sequences of 35 nucleotides from both the N-terminus and C-terminus, totaling 70 nucleotides, as input during the prediction phase.

Comparing MSlocPRED with existing methods and tools

For the prediction performance of MSlocPRED, we firstly compared it with the other four predictors in different aspects on Dataset 1. Since the original training datasets were friendly offered by DM3Loc, RNATracker, mRNALoc, and iLoc-mRNA, MSlocPRED was compared with these methods using the five-fold cross-validation according to the results listed in their works. We used the original six types of subcellular localization data provided by DM3Loc, adjusted the model parameters constructed in this study to binary classification. The area under the receiver

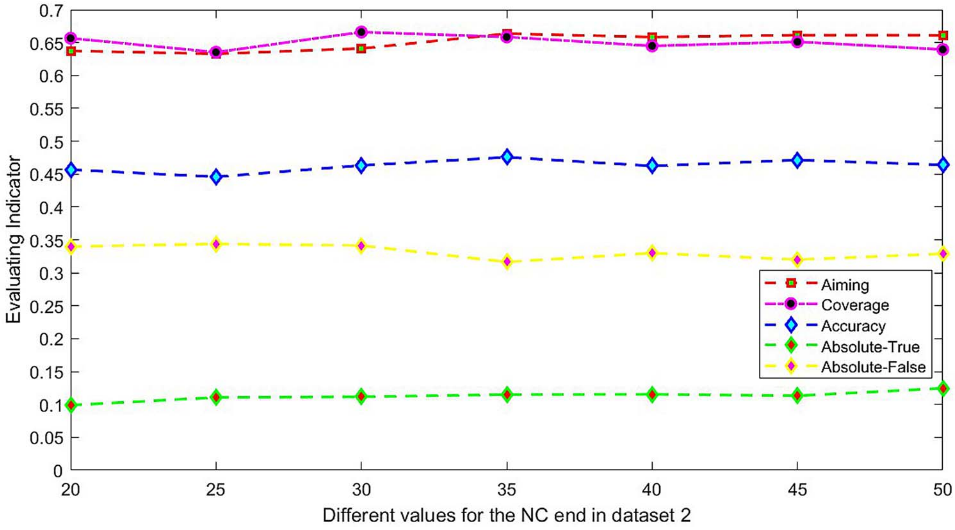

Figure 6. Prediction results of intercepted sequences with different values at the NC end on Dataset 2.

Table 11. The five-fold cross-validation comparison with existing methods on Dataset 1

Compartment	Method	AUC	APR	MCC
Nucleus	DM3Loc	**0.7725**	**0.8765**	**0.3859**
	RNATracker	0.7531	0.8601	0.345
	mRNALoc	0.6075	0.7655	0.1501
	iLoc-mRNA	0.5186	0.7200	0.0516
	MSlocPRED	0.7038	0.7586	0.3778
Exosome	DM3Loc	0.7233	0.9964	0.0736
	RNATracker	**0.7533**	**0.997**	0
	mRNALoc	0.4065	0.9887	−0.0294
	iLoc-mRNA	/	/	/
	MSlocPRED	0.6097	0.6290	**0.1700**
Cytosol	DM3Loc	0.7406	0.3193	0.2872
	RNATracker	0.7331	0.3176	0.1383
	mRNALoc	0.4529	0.1177	−0.0134
	iLoc-mRNA	0.531	0.1339	0.0253
	MSlocPRED	**0.9164**	**0.9312**	**0.6831**
Ribosome	DM3Loc	0.7589	0.5478	0.355
	RNATracker	0.7447	0.5365	0.2697
	mRNALoc	/	/	/
	iLoc-mRNA	0.7940	0.6634	0.3899
	MSlocPRED	**0.8585**	**0.8893**	**0.6145**
Membrane	DM3Loc	0.7558	0.4472	0.3115
	RNATracker	0.7386	0.4051	0.1927
	mRNALoc	/	/	/
	iLoc-mRNA	/	/	/
	MSlocPRED	**0.8917**	**0.9121**	**0.6388**
ER	DM3Loc	0.6981	0.2502	0.2048
	RNATracker	0.6265	0.1880	0
	mRNALoc	0.3729	0.1402	−0.1479
	iLoc-mRNA	0.8100	0.5702	0.3762
	MSlocPRED	**0.9275**	**0.9404**	**0.7205**

operating characteristic (ROC), the precision-recall (PR) curves, and the Matthews correlation coefficient (MCC) were used to evaluate their performance and these methods were compared. As can be seen in the Table 11, as far as AUC is concerned, compared to the existing DM3Loc, RNATracker, mRNALoc, and iLoc-mRNA, the impacts of the MSlocPRED are generally increased by 17.58% to 46.35% for cytosol, 6.45% to 11.38% for ribosome, 13.59% to 15.31% for membrane, and 11.75% to 55.46% for ER. In terms of MCC, except for Nucleus, our method obtained the best results in the other five compartments. DM3Loc achieves the best prediction performance in Nucleus, and our method obtains the second best. From the results can also be concluded that the predictor was greatly influence by the selected sample. Additionally, the best performance also indicated that deep transfer learning is more suitable than traditional machine learning algorithms for multi-label subcellular localization recognition.

To further demonstrate the effectiveness of the MSlocPRED predictor, we also compared MSlocPRED to accessible web-servers

Table 12. Performance comparison between MSlocPRED and other state-of-art tools on independent test dataset

Dataset	Method	Aiming	Coverage	Accuracy	Absolute-true	Absolute-false
Dataset 1	DM3Loc	0.9593	0.6537	0.6203	0.3310	**0.1902**
	Clarion	**0.9818**	0.4808	0.4808	0.2426	0.2665
	MSlocPRED	0.8001	**0.8003**	**0.6251**	0.2262	0.2127
Dataset 2	DM3Loc	0.7605	0.5327	**0.4855**	0.2650	**0.2338**
	Clarion	**0.7845**	0.4175	0.4168	**0.2823**	0.2683
	MSlocPRED	0.6060	**0.6780**	0.4503	0.2093	0.3338

Table 13. Summary of subcellular localization predictors

Type	Tool	Subcellular localization	Benchmark dataset size	Encoding scheme	Classifier	AUC	APR	MCC
Single-label	RNATracker	Cyt, ER, Ins, Mem, Mito, Nuc	11 373 (Dataset 1) 13 860 (Dataset 2)	One-hot RNA secondary structure	CNN LSTM Attention	0.7249	0.5507	0.1576
	iLoc-mRNA	Cyp, Cyt, Den, ER, Exo, Mito, Nuc, Rib	4901	K-mer	SVM	0.6634	0.5219	0.2108
	mRNALoc	Cyp, ER, ECR, Mito, Nuc	14 909	PseKNC	SVM	0.4600	0.5030	−0.0102
Multi-label	DM3Loc	Cyt, ER, Exo, Mem, Nuc, Rib	17 870	One-hot	CNN Attention	0.7415	0.5729	0.2697
	MSlocPRED	Exo, Nuc, NP, Chr, Cyp, No, Cyt, Mem, Rib	14 608 (Dataset 1) 22 966 (Dataset 2)	Convert to images	AlexNet	**0.8179**	**0.8434**	**0.5341**

Note: cytosol: Cyt; endoplasmic reticulum: ER; insoluble: Ins; membranes: Mem; mitochondrial: Mito; nuclear: Nuc; cytoplasm: Cyp; dendrite: Den; exosome: Exo; mitochondrion: Mito; nucleus: Nuc; ribosome: Rib; extracellular region: ECR; nucleoplasm: NP; chromatin: Chr; nucleolus: No.

DM3Loc and Clarion on independent test datasets (Dataset 1 and Dataset 2). As indicated in the Table 12, when tested using the independent test datasets (i.e. 35 nucleotides from each of the NC ends) after preprocessing in this paper, in terms of Coverage, for Dataset 1 and Dataset 2, MSlocPRED was broadly improved by 14.66%–31.95% and 14.53%–26.05%, respectively. For the four assessment metrics of Aiming, Accuracy, Absolute-False and Absolute-True, although the MSlocPRED results are lower than those of DM3Loc and Clarion, a look at the specific test labels reveals the following: the DM3Loc predicted all the test samples (Dataset 1 and Dataset 2) to be Exosome, and a few samples were predicted as both Nucleus or Nucleus and Cytoplasm. Clarion predicted almost all of the test data as Exosome, and a few samples were not predicted at all. For Dataset 1, the number of completely correct predictions by MSlocPRED were 495 for class 1, 40 for class 2, 3 for class 4, 18 for class 5, 17 for class 6, 1 for class 7, 87 for class 8, 71 for class 9, 566 for class 10, 87 for class 12, 71 for class 13, and 3 for class 15, respectively. For Dataset 2, the number of completely correct predictions by MSlocPRED were 1093 for class 1, 53 for class 2, 91 for class 3, 28 for class 5, 2 for class 6, 50 for class 8, 1 for class 10, 2 for class 11, 185 for class 13, 523 for class 16, 399 for class 17, and 120 for class 18, respectively. For the other categories that were not completely predicted correctly, analyzing the labels tested reveals that for categories containing two and more positions, most of them can be predicted for one or two positions, even though they were not completely predicted. The experimental results showed that the recognition of multi-labeled subcellular localization after the NC-terminal interception of fragments was effective using the model MSlocPRED constructed in this study.

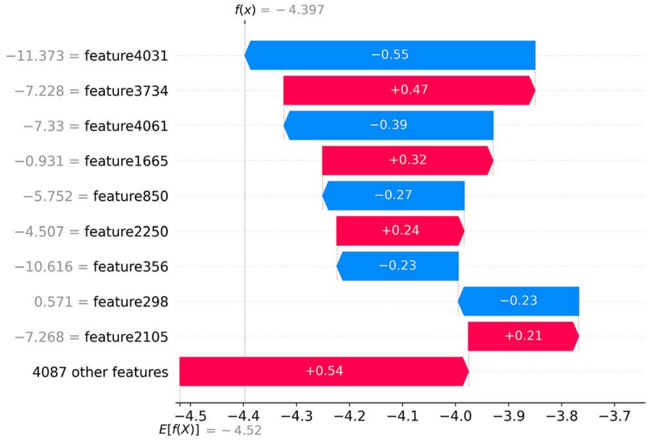

Figure 7. Waterfall chart for category2 in Dataset 1.

As shown in Table 13, a comparison of the current state-of-the-art subcellular localization models is summarized, covering aspects such as subcellular positions, the sizes of benchmark dataset, encoding schemes, and classifiers, along with average metrics of AUC, APR, and MCC for each position. For models unable to predict certain cellular compartments, the denominators of these metrics have been appropriately adjusted to ensure fairness in the comparison process. The proposed prediction model, MSlocPRED, employs mature techniques and algorithms from the field of image processing, transforming data into image form. Compared to traditional methods, the use of images as input data allows complex data relationships to be displayed visually, making pattern recognition and feature extraction more intuitive. For the input image data, feature extraction is performed using a modified AlexNet, and classification is conducted through transfer learning. This approach is particularly valuable in bioinformatics, where slight variations in experimental conditions or new experimental setups often lead to changes in data distribution. Utilizing a pre-trained image recognition network provides robust feature extraction capabilities, reduces the risk

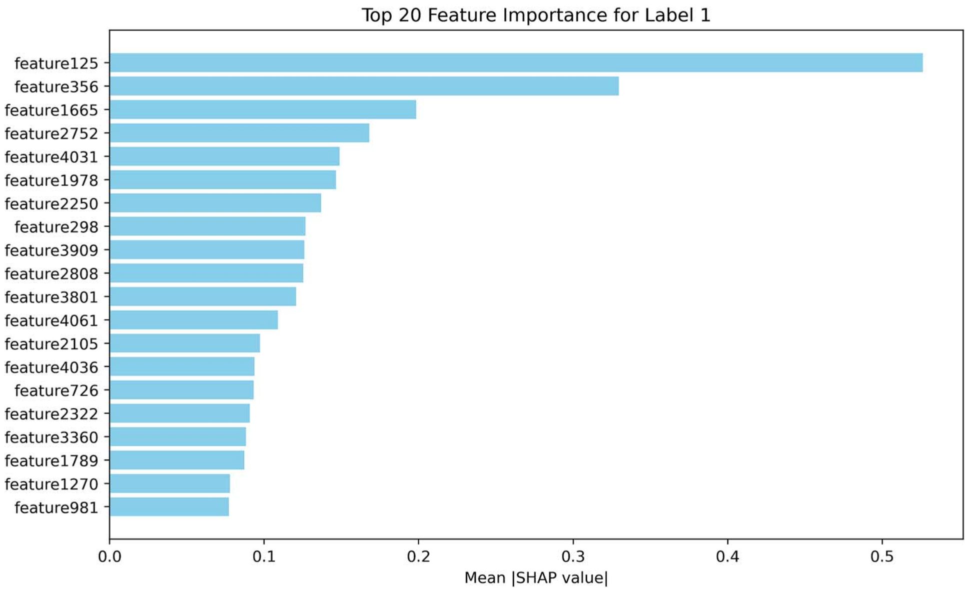

Figure 8. Bar chart for category2 in Dataset 1.

of overfitting, and enhances generalization performance of the model.

Ultimately, the average AUC, APR, and MCC of MSlocPRED reached 0.8179, 0.8434, and 0.5341, respectively, representing improvements of 0.0930, 0.2927, and 0.3233 over the highest average AUC and APR (0.7249 and 0.5507 for RNATracker) and MCC (0.2108 for iLoc-mRNA) observed in single-label methods. Compared to the most advanced multi-label method, DM3Loc (with AUC, APR, and MCC of 0.7415, 0.5729, and 0.2697, respectively), the increases were 0.0764, 0.2705, and 0.2644. Experimental results demonstrate that MSlocPRED exhibits superior performance and robustness compared to both single-label and multi-label methods. The innovative combination of this new encoding method and transfer learning techniques shows significant advantages over traditional methods in various aspects.

Model interpretability analysis based on SHapley Additive exPlanations values

Overview of SHapley Additive exPlanations

The complexity and numerous parameters of deep learning models pose significant challenges in understanding their internal workings. To address this issue, SHapley Additive exPlanations (SHAP), a machine learning interpretability approach grounded in game theory, is used. SHAP values are assigned to individual features, effectively quantifying their impact on the model's prediction outcomes, thereby facilitating a deeper understanding of the model's decision-making process. Consequently, we employed the SHAP library to generate waterfall charts, bar graphs, and bee swarm plots for the in-depth analysis of the 4096 features present in the 'fc7' layer of the AlexNet network. Our objective was to interpret the influence of these features on the prediction outcomes. To illustrate this, we selected a single label from each of the two datasets to clearly demonstrate the feature contributions to the prediction results.

Feature analysis for Dataset 1

For the second class in Dataset 1, the SHAP algorithm was employed to generate waterfall plots, bee swarm plots, and feature bar charts, providing a comprehensive analysis of the prediction results.

The waterfall chart in Fig. 7 illustrates the cumulative effect of each feature on the prediction score for Category 2 in Dataset 1. Positive SHAP values, such as those for Feature3734, Feature1665, and Feature2250, significantly boost the prediction score. In contrast, features like Feature4031 and Feature4061 have negative SHAP values, indicating they reduce the prediction score.

Figure 8 presents a bar chart that visually represents the contributions of the top-ranking features, emphasizing their magnitudes. Feature125 emerges as the most impactful, contributing a value of 0.51, followed closely by Feature 356 with a contribution of 0.31. This chart effectively highlights the relative importance of these key features in the prediction process.

A bee swarm plot provides a comprehensive view of SHAP values for all features, showcasing their distribution and influence on the prediction outcome. Figure 9 illustrates this, highlighting how most features contribute positively. However, it also reveals discernible outliers with substantial negative impacts on the prediction score, indicating the nuanced nature of feature interactions.

Feature analysis for Dataset 2

In Dataset 2, the analysis of category 8 mirrors previous findings. The cumulative impact of key features, such as Feature2608, Feature373, and Feature1012, significantly enhances the prediction score, as illustrated in Fig. 10. Conversely, negative contributions are demonstrated by features like Feature2357 and Feature1135, resulting in a decrease in the prediction score.

Similar to the analysis in Dataset 1, Fig. 11 further quantifies the feature contributions in Dataset 2 for category8. Here, Feature2608 stands out with the highest positive SHAP value of 0.215, clearly demonstrating its paramount significance in driving the prediction outcome. The comprehensive view of the feature impact in Dataset 2 for category8 is offered by Fig. 12, the bee swarm plot. It showcases a diverse distribution of SHAP values, illustrating the intricate interplay between features and their influence on the prediction. This plot highlights the significance of both high and low SHAP values in appreciating the complexity of the prediction process.

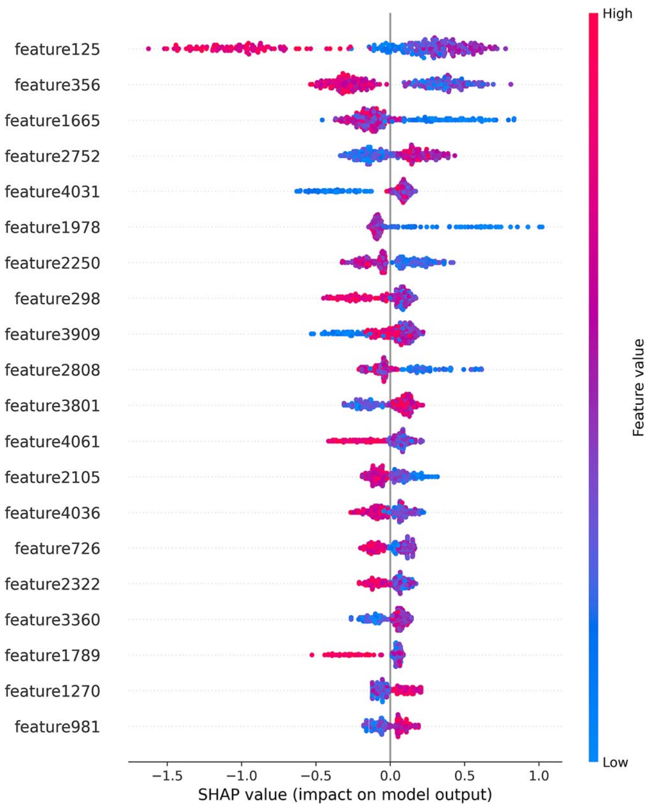

Figure 9. Bee swarm plot for category2 in Dataset 1.

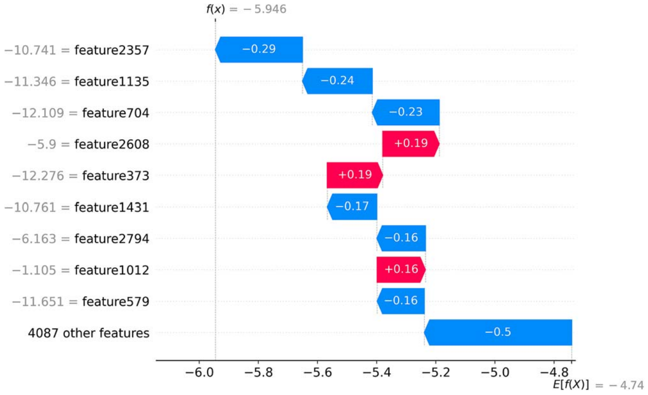

Figure 10. Waterfall chart for category8 in Dataset 2.

Conclusion and outlook

In this study, the multi-label prediction model MSlocPRED was constructed to identify multi-label mRNA subcellular localizations for seven subcellular localizations in Dataset 1 and nine subcellular localizations in Dataset 2. The preprocessed datasets were transformed into image formats, and the MDNDO–SMDU resampling technique was used to balance the number of samples in each category. Deep transfer learning was subsequently employed to develop the MSlocPRED predictive model. Comparative tests among various resampling techniques demonstrated that the proposed MDNDO–SMDU method is more effective for preprocessing subcellular localizations. The prediction performance was optimal when the NC end was intercepted by 35 nucleotides for both datasets. Independent testing and five-fold cross-validation showed that MSlocPRED significantly outperforms established tools for identifying multi-label mRNA subcellular localizations. SHAP values were used to explain the prediction process of MSlocPRED.

Although the MSlocPRED model demonstrates strong performance in predicting seven and nine subcellular localizations, there remains room for improvement in its accuracy and generalization capabilities. One of the limitations of the model is its inability to cover other important localization sites or achieve a higher accuracy, which may affect the applicability and comprehensiveness of the model in certain biological contexts. Future research will explore more themes related to mRNA, including extending the model to predict additional critical localization sites and further optimizing the prediction algorithms and processing techniques to enhance accuracy. Additionally, the possibility of developing customized models for specific biological applications deserves further investigation. Beyond transfer learning, other approaches such as meta-learning and contrastive learning will also be explored. The focus will not only be on learning strategies themselves but also on designing targeted machine learning schemes based on specific task challenges such as data imbalance, incomplete feature extraction, and the effectiveness of sampling methods. Subsequent studies will continue to examine how to integrate these learning strategies with multi-label subcellular localization methods. Through these explorations, we hope to provide more comprehensive and precise solutions for mRNA subcellular localization research.

Key Points

- Subcellular localization of mRNAs is a universal mechanism for precise and efficient control of the translation process. However, up till the present moment, most prediction methods have been designed for single-label subcellular localization, ignoring the mutual information between multi-label localizations. With this in mind, this study established a multi-label computational tool, MSlocPRED, which can be directly used to predict multi-label mRNA subcellular localization.
- MDNDO–SMDU resampling technique was firstly proposed and incorporated to reduce the proportion of the original training samples.
- Inputting NC-terminal interception of fragments directly as a picture into the convolutional neural network, and the predictive model was constructed using deep transfer learning to identify subcellular localization.
- The model interpretability analysis based on SHAP values showed that the prediction model constructed in this study, MSlocPRED, was effective.

Funding

This work is supported by the National Natural Science Foundation of China [62302198]; Natural Science Foundation of Jiangsu Province of China [BK20231035]; Fundamental Research Funds for the Central Universities [JUSRP124014]; National Key Research and Development Program of China [2021YFE010178]; National Natural Science Foundation of China [62176105]; Hong Kong Research Grants Council [PolyU152006/19E].

Conflict of interest: None declared.

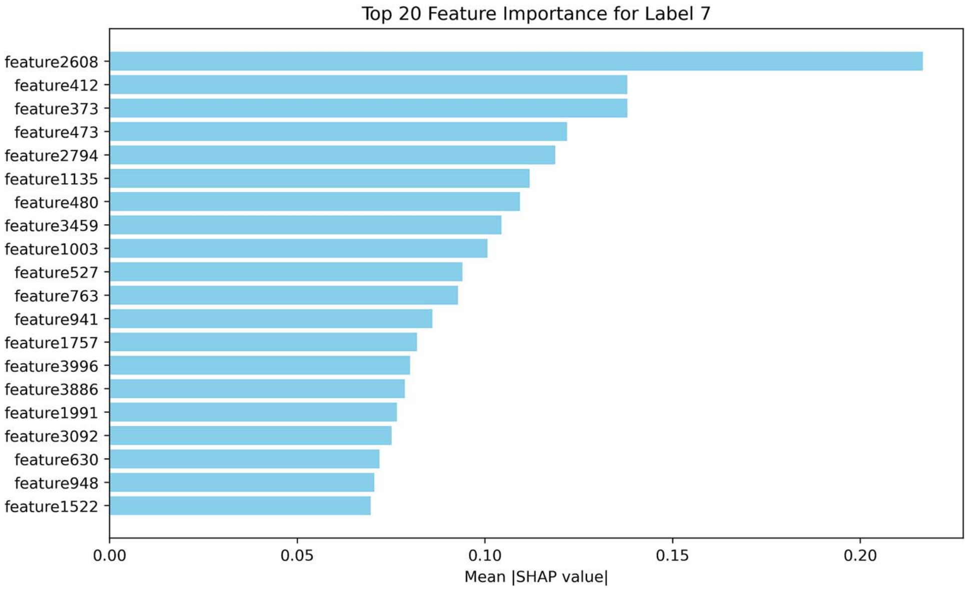

Figure 11. Bar chart for category8 in Dataset 2.

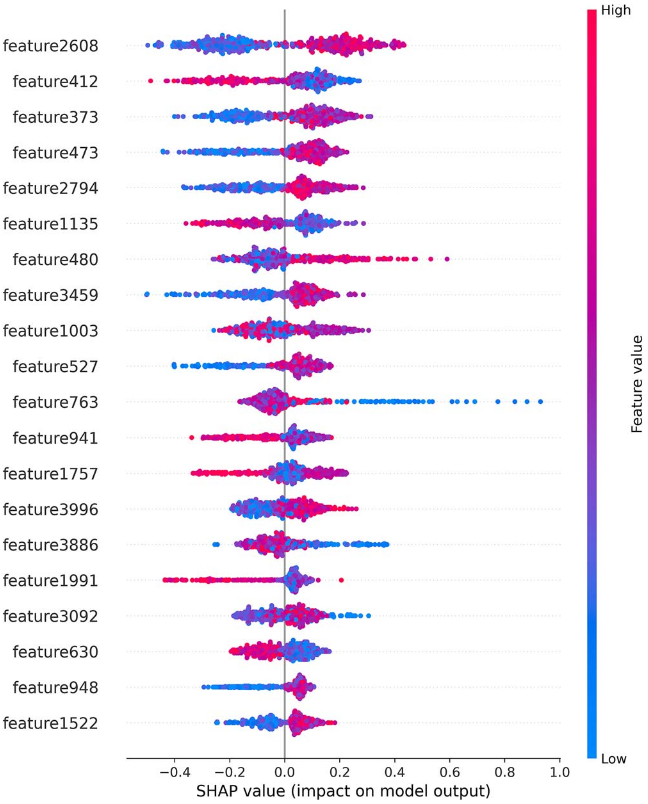

Figure 12. Bee swarm plot for category8 in Dataset 2.

Data availability

The predictive model MSlocPRED and associated datasets are available at: https://github.com/ZBYnb1/MSlocPRED/tree/main.

References

1. Buxbaum AR, Haimovich G, Singer RH. In the right place at the right time: visualizing and understanding mRNA localization. *Nat Rev Mol Cell Biol* 2015;**16**:95–109. https://doi.org/10.1038/nrm3918.
2. Lashkevich KA, Dmitriev SE. mRNA targeting, transport and local translation in eukaryotic cells: from the classical view to a diversity of new concepts. *Mol Biol* 2021;**55**:507–37. https://doi.org/10.1134/S0026893321030080.
3. Ross J. mRNA stability in mammalian cells. *Microbiol Rev* 1995;**59**:423–50. https://doi.org/10.1128/mr.59.3.423-450.1995.
4. Wang R, Jiang Y, Jin J. *et al.* DeepBIO: an automated and interpretable deep-learning platform for high-throughput biological sequence prediction, functional annotation and visualization analysis. *Nucleic Acids Res* 2023;**51**:3017–29. https://doi.org/10.1093/nar/gkad055.
5. Cheng H, Rao B, Liu L. *et al.* PepFormer: end-to-end transformer-based siamese network to predict and enhance peptide detectability based on sequence only. *Anal Chem* 2021;**93**:6481–90. https://doi.org/10.1021/acs.analchem.1c00354.
6. Li H, Pang Y, Liu B. BioSeq-BLM: a platform for analyzing DNA, RNA, and protein sequences based on biological language models. *Nucleic Acids Res* 2021;**49**:e129. https://doi.org/10.1093/nar/gkab829.
7. Bhatti GK, Khullar N, Sidhu IS. *et al.* Emerging role of non-coding RNA in health and disease. *Metab Brain Dis* 2021;**36**:1119–34. https://doi.org/10.1007/s11011-021-00739-y.
8. Chin A, Lécuyer E. RNA localization: making its way to the center stage. *Biochim Biophys Acta Gen Subj* Nov 2017;**1861**:2956–70. https://doi.org/10.1016/j.bbagen.2017.06.011.
9. Nussbacher JK, Tabet R, Yeo GW. *et al.* Disruption of RNA metabolism in neurological diseases and emerging therapeutic interventions. *Neuron* 2019;**102**:294–320. https://doi.org/10.1016/j.neuron.2019.03.014.
10. Liu B, Gao X, Zhang H. BioSeq-Analysis2.0: an updated platform for analyzing DNA, RNA and protein sequences at sequence level and residue level based on machine learning approaches. *Nucleic Acids Res* 2019;**47**:e127. https://doi.org/10.1093/nar/gkz740.
11. Liu Y, Shen X, Gong Y. *et al.* Sequence alignment/map format: a comprehensive review of approaches and applications. *Brief Bioinform* 2023;**24**:bbad320. https://doi.org/10.1093/bib/bbad320.
12. Zhu H, Hao H, Yu L. Identifying disease-related microbes based on multi-scale variational graph autoencoder embedding Wasserstein distance. *BMC Biol* 2023;**21**:294. https://doi.org/10.1186/s12915-023-01796-8.

13. Wei L, Xing P, Shi G. *et al.* Fast prediction of protein methylation sites using a sequence-based feature selection technique. *IEEE/ACM Trans Comput Biol Bioinform* 2019;**16**:1264–73. https://doi.org/10.1109/TCBB.2017.2670558.
14. Zhou H, Wang H, Tang J. *et al.* Identify ncRNA subcellular localization via graph regularized k-local hyperplane distance nearest neighbor model on multi-kernel learning. *IEEE/ACM Trans Comput Biol Bioinform* 20222022;**19**:3517–29. https://doi.org/10.1109/TCBB.2021.3107621.
15. Ding YJ, Tiwari P, Guo F. *et al.* Shared subspace-based radial basis function neural network for identifying ncRNAs subcellular localization. *Neural Netw* 2022;**156**:170–8. https://doi.org/10.1016/j.neunet.2022.09.026.
16. Wang Y, Zhai Y, Ding Y. *et al.* SBSM-pro: support bio-sequence machine for proteins *arXiv preprint*, p. arXiv:2308.10275. 2023.
17. Zhang ZY, Zhang Z, Ye X. *et al.* A BERT-based model for the prediction of lncRNA subcellular localization in Homo sapiens. *Int J Biol Macromol* 2024;**265**:130659. https://doi.org/10.1016/j.ijbiomac.2024.130659.
18. Sun Z-J, ZHANG Z-Y, YANG Y-H. *et al.* Towards a better prediction of subcellular location of long non-coding RNA. *Front Comput Sci* 2022;**16**:165903.
19. Zhang ZY. *et al.* iLoc-miRNA: extracellular/intracellular miRNA prediction using deep BiLSTM with attention mechanism. *Brief Bioinform* 2022;**23**:1–10. https://doi.org/10.1093/bib/bbac395.
20. Li H, Liu B. BioSeq-Diabolo: biological sequence similarity analysis using Diabolo. *PLoS Comput Biol* 2023;**19**:e1011214. https://doi.org/10.1371/journal.pcbi.1011214.
21. Wang D, Zhang Z, Jiang Y. *et al.* DM3Loc: multi-label mRNA subcellular localization prediction and analysis based on multi-head self-attention mechanism. *Nucleic Acids Res* 2021;**49**:e46. https://doi.org/10.1093/nar/gkab016.
22. Li J, Zhang L, He S. *et al.* SubLocEP: a novel ensemble predictor of subcellular localization of eukaryotic mRNA based on machine learning. *Brief Bioinform* 2021;**22**:1–11. https://doi.org/10.1093/bib/bbaa401.
23. Bi Y, Li F, Guo X. *et al.* Clarion is a multi-label problem transformation method for identifying mRNA subcellular localizations. *Brief Bioinform* 2022;**23**:1–12. https://doi.org/10.1093/bib/bbac467.
24. Yuan GH, Wang Y, Wang GZ. *et al.* RNAlight: a machine learning model to identify nucleotide features determining RNA subcellular localization. *Brief Bioinform* 2023;**24**:1–13. https://doi.org/10.1093/bib/bbac509.
25. Wang S, Shen Z, Liu T. *et al.* DeepmRNALoc: a novel predictor of eukaryotic mRNA subcellular localization based on deep learning. *Molecules* 2023;**28**:2284. https://doi.org/10.3390/molecules28052284.
26. Zhang T, Tan P, Wang L. *et al.* RNALocate: a resource for RNA subcellular localizations. *Nucleic Acids Res* 2017;**45**:D135–d138. https://doi.org/10.1093/nar/gkw728.
27. Sayers EW, Cavanaugh M, Clark K. *et al.* GenBank. *Nucleic Acids Res* 2019;**47**:D94–d99. https://doi.org/10.1093/nar/gky989.
28. Cui T, Dou Y, Tan P. *et al.* RNALocate v2.0: an updated resource for RNA subcellular localization with increased coverage and annotation. *Nucleic Acids Res* 2022;**50**:D333–d339. https://doi.org/10.1093/nar/gkab825.
29. Xia S, Feng J, Chen K. *et al.* CSCD: a database for cancer-specific circular RNAs. *Nucleic Acids Res* 2018;**46**:D925–d929. https://doi.org/10.1093/nar/gkx863.
30. Liu T, Zhang Q, Zhang J. *et al.* EVmiRNA: a database of miRNA profiling in extracellular vesicles. *Nucleic Acids Res* 2019;**47**:D89–d93. https://doi.org/10.1093/nar/gky985.
31. Li S, Li Y, Chen B. *et al.* exoRBase: a database of circRNA, lncRNA and mRNA in human blood exosomes. *Nucleic Acids Res* 2018;**46**:D106–d112. https://doi.org/10.1093/nar/gkx891.
32. Lock A, Rutherford K, Harris MA. *et al.* PomBase 2018: user-driven reimplementation of the fission yeast database provides rapid and intuitive access to diverse, interconnected information. *Nucleic Acids Res* 2019;**47**:D821–d827. https://doi.org/10.1093/nar/gky961.
33. Berardini TZ, Reiser L, Li D. *et al.* The Arabidopsis information resource: making and mining the “gold standard” annotated reference plant genome. *Genesis* 2015;**53**:474–85. https://doi.org/10.1002/dvg.22877.
34. Chou KC. Some remarks on predicting multi-label attributes in molecular biosystems. *Mol Biosyst* 2013;**9**:1092–100. https://doi.org/10.1039/c3mb25555g.
35. Jambhekar A, Derisi JLJR. Cis-acting determinants of asymmetric, cytoplasmic RNA transport. *RNA* 2007;**13**:625–42. https://doi.org/10.1261/rna.262607.
36. Martin KC, Ephrussi AJC. mRNA localization: gene expression in the spatial dimension. *Wiley Interdisciplinary Reviews: RNA* 2009;**136**:719–30. https://doi.org/10.1016/j.cell.2009.01.044.
37. Zichao Y, Eric L, Mathieu BJB. Prediction of mRNA subcellular localization using deep recurrent neural networks. 2019;**35**:i333–42.
38. Meer EJ, Wang DO, Kim S. *et al.* Identification of a cis-acting element that localizes mRNA to synapses. *Proc Natl Acad Sci U S A* 2012;**109**:4639–44.
39. Bergalet J, Lécuyer E. The functions and regulatory principles of mRNA intracellular trafficking. *Adv Exp Med Biol* 2014;**825**:57–96.