




TACOS: a novel approach for accurate prediction of cell-specific long noncoding RNAs subcellular localization

Young-Jun Jeon , Md. Mehedi Hasan , Hyun Woo Park, Ki Wook Lee and Balachandran Manavalan 

Corresponding authors: Young-Jun Jeon, Department of Integrative Biotechnology, College of Biotechnology & Bioengineering, Sungkyunkwan University, Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do 16419, Republic of Korea. Tel: +82-31-290-7862; Fax: +82-31-290-7870; E-mail: jeon2020@skku.edu; Balachandran Manavalan, Computational Biology and Bioinformatics Lab, Department of Integrative Biotechnology, College of Biotechnology & Bioengineering, Sungkyunkwan University, Seobu-ro, Jangan-gu, Suwon-si, Gyeonggi-do 16419, Republic of Korea. Tel: +82-31-290-7860; Fax: +82-31-290-7870; E-mail: bala2022@skku.edu

Abstract

Long noncoding RNAs (lncRNAs) are primarily regulated by their cellular localization, which is responsible for their molecular functions, including cell cycle regulation and genome rearrangements. Accurately identifying the subcellular location of lncRNAs from sequence information is crucial for a better understanding of their biological functions and mechanisms. In contrast to traditional experimental methods, bioinformatics or computational methods can be applied for the annotation of lncRNA subcellular locations in humans more effectively. In the past, several machine learning-based methods have been developed to identify lncRNA subcellular localization, but relevant work for identifying cell-specific localization of human lncRNA remains limited. In this study, we present the first application of the tree-based stacking approach, TACOS, which allows users to identify the subcellular localization of human lncRNA in 10 different cell types. Specifically, we conducted comprehensive evaluations of six tree-based classifiers with 10 different feature descriptors, using a newly constructed balanced training dataset for each cell type. Subsequently, the strengths of the AdaBoost baseline models were integrated via a stacking approach, with an appropriate tree-based classifier for the final prediction. TACOS displayed consistent performance in both the cross-validation and independent assessments compared with the other two approaches employed in this study. The user-friendly online TACOS web server can be accessed at <https://balalab-skku.org/TACOS>.

Keywords: long noncoding RNAs, stacking strategy, tree-based algorithms, feature extraction, sequence analysis, bioinformatics

Introduction

RNA is one of the main components in the central dogma of molecular biology and plays vital roles in different biological processes [1]. Early sequencing data have shown that >80% of the mammalian genome is transcribed into noncoding regions, whereas only a smaller portion is transcribed into protein coding RNAs [2, 3]. Owing to the advancements in sequencing technologies and bioinformatics analysis, an increasing number of noncoding RNAs (ncRNAs) have been identified, including circular RNAs, long ncRNAs (lncRNAs) and small ncRNAs [4–6]. ncRNAs are the largest constituent of the transcriptome

that do not possess functional open reading frames, but play potent roles in several biological processes, including disease pathogenesis [1]. Genomes undergo extensive transcription, resulting in thousands of lncRNAs that are >200 nucleotides in length and do not undergo translation to become functional proteins. They function as decoys, enhancer RNAs, guide, scaffold, signal and short peptides [7, 8]. lncRNAs can modulate chromatin function, affect signaling mechanisms, alter the stability of cytoplasmic mRNAs translation and regulate the assembly and function of nuclear bodies based on their localization and interactions with other biological

Young-Jun Jeon is an assistant professor in the Department of Integrative Biotechnology, Sungkyunkwan University, Republic of Korea. His research team's overarching goal is to identify biomarkers and drug resistance mechanisms in cancer using cell biology and NGS-based approaches.

Md. Mehedi Hasan is currently a postdoctoral researcher at the Tulane Center for Aging and Department of Medicine, Tulane University, USA. Prior to his current position, he worked at the Japan Society for the Promotion of Science International PD fellow at Kyushu Institute of Technology, Japan. He has also worked as a researcher at the Chinese University of Hong Kong. His primary research interests include protein structure prediction, machine learning, data mining, computational biology and functional genomics.

Hyun Woo Park is a graduate of Young-Jun Jeon's laboratory at Sungkyunkwan University. He is interested in developing bioinformatics and machine learning pipelines for identifying prognostic biomarkers for disease, as well as characterizing chemoresistant mechanisms using cancer cell biology.

Ki Wook Lee is a PhD student at Young-Jun Jeon's laboratory at Sungkyunkwan University. His research interests include developing bioinformatics pipelines and machine learning-based models for identifying prognostic biomarkers in solid tumors and neurodegenerative disorders and conducting cell biology experiments to identify drug resistance mechanisms in solid tumors.

Balachandran Manavalan is an assistant professor at the Department of Integrative Biotechnology, Sungkyunkwan University, Republic of Korea. He is also an associate member of the Korea Institute for Advanced Studies, Republic of Korea. His research interests include artificial intelligence, bioinformatics, machine learning, big data and functional genomics.

Received: March 1, 2022. Revised: May 23, 2022. Accepted: May 24, 2022

© The Author(s) 2022. Published by Oxford University Press.

This is an Open Access article distributed under the terms of the Creative Commons Attribution-NonCommercial License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact journals.permissions@oup.com

macromolecules. Several of these functions affect gene expression in diverse biological and pathophysiological pathways such as in cancer, immune responses and neuronal diseases [9].

As evidenced, the human genome constitutes >16 000 lncRNA genes [10]; however, other sources have predicted >100 000 lncRNAs [11]. To date, only a small portion of lncRNAs have been characterized; hence, it is necessary to identify the functional characterization of others [9, 12]. Furthermore, the versatile functions of lncRNAs are determined by their subcellular localizations. Hence, understanding the cellular localizations of lncRNAs will help decipher their potential molecular mechanisms.

In situ hybridization (ISH) is a popular technique to identify the cellular localization of candidate lncRNAs using labeled complementary oligonucleotide probes [13, 14]. Single-molecule fluorescence ISH is the gold standard method, in which multiple probes are utilized to amplify the fluorescent signal for the detection of target RNAs, which are present in low levels [15]. In contrast, fluorescent *in situ* RNA sequencing method provides *in situ* information at high-throughput levels [16]. Spatially resolved transcript amplicon readout mapping is another method in which three-dimensional locational information on RNA expression in intact tissue samples is provided [17]. As the identified lncRNAs outnumber those with known localizations, it is necessary to implement rapid, efficient and cost-effective computational methods to assist in their identifications.

To date, only a few computational approaches have been developed to predict lncRNA subcellular localization across tissues/cell lines [18–22]. Cao et al. [19] proposed a predictor named lncLocator, which was developed based on the RNALocate database [23], to determine five localizations. This predictor adopts k-mer frequency information features with random forest (RF), support vector machine (SVM) and an autoencoder. To construct a balanced training model dataset, lncLocator utilized the synthetic minority oversampling technique [24]. In 2018, Gudenas and Wang [20] developed DeepLncRNA, which predicts lncRNA subcellular localization directly from transcript sequences. They analyzed 93 strand-specific RNA-seq samples from multiple cell types by extracting k-mer frequencies using deep neural networks. Su et al. [22] developed iLoc-lncRNA by incorporating 8-tuple nucleotide features into the general Pseudo k-tupler composition (PseKNC) using SVM. Recently, Ahmad et al. [18] developed Locate-R by extracting k-mer features using a deep local SVM to classify four locations. Recently, Lin et al. developed lncLocator 2.0 [21] as a cell-line-specific subcellular localization predictor using an interpretable deep-learning approach. Among the existing predictors, lncLocator 2.0, is the only cell-specific predictor available, but it has sufficient room for improvement. To develop a machine learning (ML)-based predictor, it is necessary to devise appropriate encoding approaches to represent the lncRNA sequence fragments surrounding subcellular localization across tissues/cell lines.

In this study, we developed a Tree-based Algorithm for Cell-specific long non-coding RNA Subcellular location (TACOS) for the accurate detection of cell-specific human lncRNA subcellular locations, an overview of which is shown in Figure 1. First, we constructed a balanced training datasets for each of the 10 different cell types, including A549, GM12878, H1 human embryonic stem cell line (HESC), HeLa.S3 (HELA), Hep G2 (HEPG), HT1080, HUVEC, NHEK, SK.MEL.5 (SKMEL) and SK.N.SH (SKNS). Utilizing a balanced training dataset for each cell type, we tested six different tree-based classifiers [RF, extremely gradient boosting (XGB), AdaBoost (AB), gradient boosting (GB), light GB (LGB) and extremely randomized tree (ERT)] using 10 different feature descriptors (which cover composition and physicochemical properties) and identified appropriate classifier-based baseline models. Subsequently, we integrated these 10 baseline models predicted probability values with an appropriate tree-based classifier through stacking strategy to make the final prediction. Notably, TACOS is the first application of tree-based algorithms employed for identifying cell-specific lncRNA subcellular locations. TACOS will be able to assist experimentalists in identifying novel lncRNA locations and elucidating their functions on a larger scale.

Materials and methods

Dataset construction

To develop a prediction model based on sequence information, lncRNA nucleotide sequences and localization information are required. Recently, Lin et al. [21] recently constructed a high-quality dataset based on nucleotide sequences with variable lengths obtained from the GENCODE project [25] and localization information obtained from lncATLAS [26]. In order to determine the location of lncRNA, the authors used the cytoplasm/nucleus relative concentration index (CNRCI) for different cell types and determined that if CNRCI is >1, the lncRNA is located within the cytoplasm, and if it is <-1, it is located within the nucleus. These data are available through the following link: <https://github.com/Yang-J-LIN/lncLocator2>. For each cell type, they generated nonredundant datasets by applying the CD-HIT [27] threshold of 0.8, which means that none of the sequences shared >80% sequence identity. Specifically, the total dataset was divided into 8/1/1 sets and used as train/dev/test sets, where train and dev sets were used for parameter optimization and model building, and the test sets were used to evaluate the model.

We utilized the same sequence and respective classification information in the current study with the following modifications: (i) the train and dev datasets were combined to generate new training dataset for each cell type, resulting in a greater proportion of negative samples than positive samples. By utilizing an imbalanced dataset, any classifier will ultimately introduce class biases during cross-validation/training. (ii) To avoid such circumstances, we considered all positive samples and an equal number of negative samples randomly selected

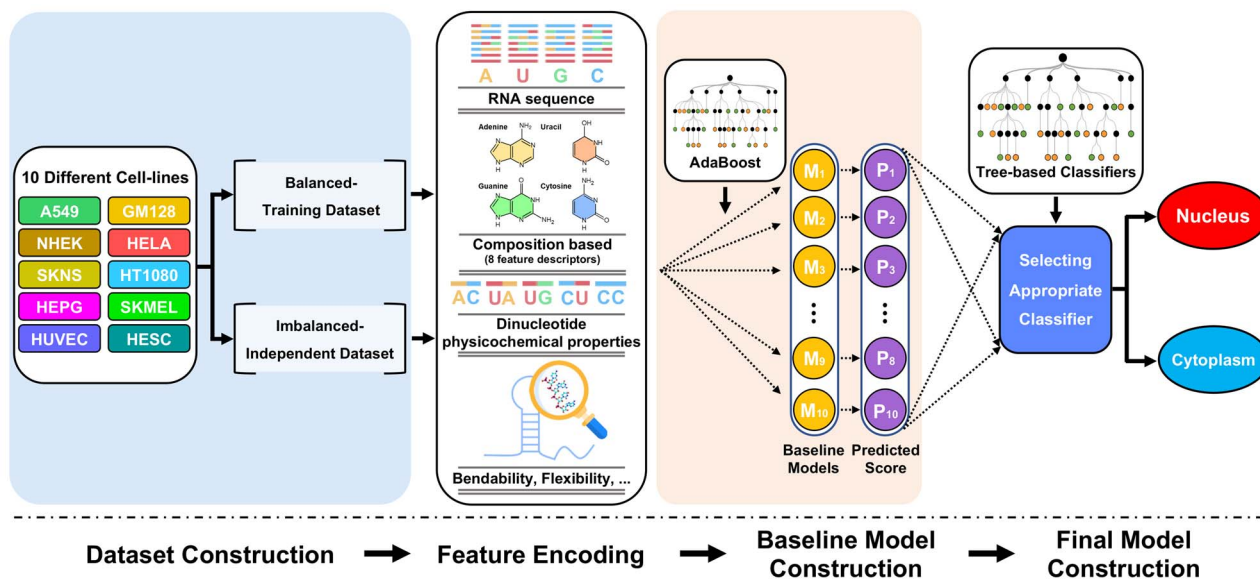


Figure 1. An overview of TACOS. It involves the following steps: dataset construction, feature extraction, baseline model construction and final model construction.

Table 1. A statistical summary of the cell-specific dataset employed in this study

Cell types	Training		Independent	
	Positive	Negative	Positive	Negative
A549	4523	4523	432	1682
GM12878	3130	3130	365	5841
HELA	3356	3356	317	2653
HESC	5961	5961	677	3869
HT1080	4505	4505	459	1499
HUVEC	3739	3739	492	6038
HEPG	4129	4129	511	5300
NHEK	3439	3439	483	4025
SKNS	4587	4587	506	5984
SKMEL	3593	3593	408	3829

from the original samples. (iii) The leftover negative samples from the new training dataset were considered as negative samples for the independent dataset and supplemented with the same positive samples from the test set. A statistical summary of the dataset employed in this study is provided in the Table 1, where the training samples comprise equal numbers of positives and negatives. In contrast, an independent dataset that replicates the actual scenario contains an imbalanced data.

Feature extraction

Feature extraction is one of the most critical steps in constructing an ML model. In general, multiple feature encodings [28–31] should be explored on a single dataset, rather than exploring on collection of specific encodings. In this study, we investigated 10 different encodings for each cell type and assessed their ability to distinguish positive samples from negative samples. A brief description of each encoding calculation is provided below.

Composition of k -spaced nucleic acid pairs (CKSNAP)

The CKSNAP algorithm calculates the frequency of dinucleotides separated by k nucleic acids (k was set to 3). As an example, if k is equal to 0, it generates 16 0-spaced dinucleotide pairs ('CU', 'CA', 'GC', 'GA', 'GG', 'GU', 'CG', 'UG', 'CC', 'AA', 'AU', 'AG', 'AC', 'UG', 'UU', 'UA'). The feature vector is defined as:

$$\text{CKSNAP} = \left(\frac{R_{CG}}{S}, \frac{R_{AU}}{S}, \frac{R_{UU}}{S}, \frac{R_{CA}}{S}, \dots, \frac{R_{AA}}{S} \right)_{16} \quad (1)$$

For each descriptor, the value corresponds to the dinucleotide in the given sequence. From the given sequence, the frequency of dinucleotide mn is represented by R_{mn} , and the sum of 0-spaced dinucleotides is represented by S . Here, k is set in the range of 0–3 with an interval of 1 that generates a 64D feature vector.

KC is a combination of Kmer and other features. (i) Kmer: Kmer encoding determines the number of possible nucleotides or nucleotide pairs that are present in a given sequence. Previous studies have provided mathematical formulations for Kmer calculations [32, 33]. Setting Kmer > 4 , one can generate many features and suffer from dimensional disasters. We set Kmer = 1 (monomer), 2 (dimer), 3 (trimer) and 4 (tetramer) to avoid irrelevant and redundant information. Finally, all of these Kmers were combined, resulting in 340-D ($= 4 + 16 + 64 + 256$) features for the given input sequence.

(ii) Other features, including Z-cure, GC content, AUGC ratio and GC skew, are mathematically represented as follows:

$$\text{curve} = \begin{cases} X = (F_A + F_G) - (F_C + F_U) \\ Y = (F_A + F_C) - (F_G + F_U) \\ X = (F_A + F_U) - (F_G + F_C) \end{cases} \quad (2)$$

$$GC = \frac{F_A + F_C}{F_A + F_U + F_G + F_C} \quad (3)$$

$$AUGC = \frac{F_A + F_U}{F_G + F_C} \quad (4)$$

$$GCskew = \frac{F_G - F_C}{F_G + F_C} \quad (5)$$

where F_X represents the frequency of the nucleotide X. In the end, KC combined Kmer and other features to produce a 346D feature vector.

Dinucleotide physicochemical properties (DPCP)

Using 21 physicochemical properties of RNA listed in the iLearn package (excluding one of the free energies) [28], we computed the DPCP as follows:

$$DPCP = f(m) \times RNA_PCP(X_m)_n, \quad (6)$$

X_m is the value of the n^{th} ($b=1,2,\dots,21$) RNA dinucleotide physicochemical properties (RNA_PCP). Ultimately, DPCP provides a 336D vector.

PseKNC

K-tuple composition is incorporated by PseKNC, which is defined as follows:

$$V = (r_1, r_2, \dots, r_{16}, r_{16+1}, \dots, r_{16+\lambda\tau})^T, \quad (7)$$

where

$$r_u = \begin{cases} \frac{f_m}{\sum_{a=1}^{4^k} f_{a+\sigma} \sum_{b=1}^{\lambda} \theta_b}, & (1 \leq m \leq 4) \\ \frac{\sigma^{\theta} m^{-4^k}}{\sum_{a=1}^{4^k} f_{a+\sigma} \sum_{b=1}^{\lambda} \theta_b}, & (4^k \leq m \leq 4^k + \lambda) \end{cases} \quad (8)$$

where f_m ($m=1, 2, 3, \dots, 4^k$) represents the normalized dinucleotide frequency of the a^{th} nucleotide in the sequence. σ and τ represent the weight factor and number of physiochemical indices, respectively. We set the default values of the six indices for the RNA sequences. θ_b ($b=1, 2, \dots, \lambda$) is the b -tier correlation factor. A detailed description of PseKNC was provided in a previous study [34], and the same procedure was employed to generate a 261D feature vector with the following parameters: $\sigma=0.8$, $k=4$ and $\lambda=5$.

X-mer K-spaced Y-mer composition frequency

This method was used to determine the composition of a sequence of nucleotides composed of X and Y consecutive nucleotides with intervals k . Using $k=2$, we calculated the Mono-Mono, Mono-Di, Mono-Tri, Di-Mono, Di-Tri and Di-Di compositions, which encode 32, 128, 128, 256, 256 and 256D feature vectors, respectively. These six descriptors are referred to as F1, F2, F3, F4, F5 and F6, respectively. Notably, RNA sequences were converted into vectors using the PyFeat tool [35].

Tree-based ML algorithms

The present study focused on predicting subcellular localization of lncRNAs, which is a binary classification problem. The purpose of this study was to determine whether lncRNAs are located in the cytoplasm or the nucleus. To identify the optimal ML algorithms, we investigated six tree-based approaches: RF [36], ERT [37], XGB, AB, LGB and GB. All of these classifiers have been widely applied to diverse bioinformatics sequence-based function prediction tasks [38–40]. Importantly, these classifiers are capable of handling unnormalized features more efficiently than SVMs and deep learning algorithms that use normalized features. Grid search and 10-fold cross-validation were used to optimize the hyperparameters for each classifier [41, 42], the parameter search ranges of which are provided in [Supplementary Table S1](#). In fact, we repeated this procedure 10 times reported the average performance and selected the median parameter for constructing the final model. This study used the scikit-learn package in Python [43] to implement RF, ERT, GB and AB. LGB was implemented in python using the lightgbm package (<https://github.com/Microsoft/LightGBM>) and XGB in python using the xgboost package (https://xgboost.readthedocs.io/en/stable/python/python_api.html).

Performance evaluation

There are several widely used performance metrics [44–46] that can be used to measure each model's performance, including accuracy (ACC), sensitivity (Sn), specificity (Sp), Matthew's correlation coefficient (MCC) and area under the receiver operating curves (AUC). The mathematical equations for ACC, Sn, Sp and MCC are given below:

$$ACC = \frac{TP + TN}{TP + FN + FP + FN} \quad (9)$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \quad (10)$$

$$Sn = \frac{TP}{TP + FN} \quad (11)$$

$$Sp = \frac{TN}{TN + FP} \quad (12)$$

The number of true positives, true negatives, false positives and false negatives is represented by TP, TN, FP and FN, respectively.

Results and discussion

Evaluation of trees-based algorithms for 10 different cell types using training datasets

We carried out a comprehensive analysis of the prediction performance of six classifiers (RF, ERT, LGB, GB, XGB and AB) using 10 different feature descriptors and 10-fold cross-validation. Furthermore, the 10-fold cross-validation was repeated 10 times and

the average metrics for 10 cell types was reported (Supplementary Figures S1–S10). According to Supplementary Figures S1–S9, five different encodings F4, F5, F6, KC and PseKNC in nine different cell types (except HESC) have similar performances regardless of the classifier, which is superior to the remaining encodings. Nevertheless, eight different encodings for HESC (Supplementary Figure S10) showed comparable performance (except F1 and DPCP) regardless of the classifier. In order to obtain an overview of the performances and to understand the level of dataset difficulty, we averaged 10 feature-descriptors-based models for each classifier, and the results are shown in Figure 2. Considering that we deal with a balanced training datasets and imbalanced independent datasets, MCC is an appropriate metric that has been recommended in previous studies [47, 48]. The average MCC of the different classifiers for the three cell types (A549, HESC and SKNS) was above 0.48, which indicates that it is relatively easier to distinguish the cytoplasm from the nucleus when compared with the remaining seven cell types. However, the average MCC of the two cell types (HUVEC and SKMEL cells) was below 0.35, indicating that they are the most challenging cell types for classifying cytoplasm/nuclei. Furthermore, on average, the AB classifier consistently performed well on the training datasets for 10 different cell types.

To obtain a clearer picture of the best-performing model for each cell type, we selected the best single-feature-based model or baseline model for each of the six classifiers. Supplementary Table S2 shows the performances of the best baseline models for each classifier in each cell type. Evidently, AB-PseKNC model consistently produces the best overall metrics, including MCC, ACC and AUC values. Specifically, it achieved MCC scores of 0.629, 0.455, 0.534, 0.485, 0.619, 0.484, 0.411, 0.492, 0.394 and 0.561, respectively, for A549, GM128, HELA, HEPG, HESC, HT1080, HUVEC, NHEK, SKMEL and SKNS. It is noteworthy that the majority of the remaining classifiers also achieved their best performance with PseKNC encoding, yet the metrics were lower than the AB, indicating that the level of discriminative pattern between cytoplasm and nucleus exhibited in PseKNC is higher than that of the other encodings. Among the tree-based algorithms used, the AB classifier effectively identified hidden patterns from PseKNC, resulting in superior performance for all cell types.

Evaluation of baseline models for each cell type using independent dataset

We evaluated all 60 models (6 classifiers \times 10 encodings) on independent datasets for each cell type. Figure 3 shows the average performances of each classifier with respect to different cell types. Generally speaking, the average performance of each classifier declined compared with the cross-validation performance regardless of the cell type. Interestingly, the average MCC of the AB classifier consistently achieved the best performance compared with other classifiers in all cell types, which

is consistent with the AB superiority observed during cross-validation. It is noteworthy that not only the average MCC, but also the best AB-based single model outperformed their counterparts across all cell types (Supplementary Table S2). Specifically, single models AB-F4, AB-PseKNC, AB-PseKNC, AB-PseKNC, AB-KC, AB-PseKNC, AB-F6, AB-KC and AB-PseKNC each achieved the MCC values of 0.472, 0.277, 0.384, 0.320, 0.464, 0.316, 0.199, 0.271, 0.258 and 0.399 for A549, GM128, HELA, HEPG, HESC, HT1080, HUVEC, NHEK, SKMEL and SKNS.

It is interesting to note that three encodings (F4, F6 and KC) were able to achieve their best performance on four different cell types during independent evaluation, but their performance was relatively average during cross-validation. However, the AB-based best baseline model (AB-PseKNC) for six species (GM128, HELA, HEPG, HESC, HUVEC and SKNS), obtained based on cross-validation performance or training, maintained a similar level of performance (in terms of ACC) during the independent evaluation. The absolute difference in ACC for these six species models was $<6\%$, indicating their robustness compared with the other models. In general, it is straightforward to select the most consistent model for each cell type. By contrast, we employed different strategies to enhance the robustness of the model.

Exploration of three different strategies to improve the model performances

In this section, we summarize the three strategies followed by their performances as follows:

(i) Strategy (S1): We combined all the feature descriptors linearly and created a hybrid feature, which was fed into a tree-based algorithm for the development of respective prediction models using 10-fold cross-validation. Subsequently, we compared the six models and selected the one with the highest MCC. The feature selection technique described in a previous study [49] was applied to the hybrid features, but it did not improve the prediction performance as much as anticipated (data not shown). Therefore, the control (hybrid feature based) was considered as the final model.

(ii) Strategy 2 (S2): The AB-based classifier achieved the best performance on both cross-validation and independent datasets; therefore, we only considered AB-based baseline models based on 10 feature descriptors for each cell type. The predicted probability values of the cytoplasmic location derived from these 10 models were integrated, and stacked model was developed using an appropriate classifier derived from tree-based classifiers.

(iii) Strategy 3 (S3): Unlike in S2, all six classifiers-based baseline models were considered. In total, we obtained 60 baseline models (6 classifiers \times 10 encodings), whose predicted probability of cytoplasmic location was concatenated and considered as the novel feature vector for training six different classifiers, and the appropriate meta-model for each cell type was identified.

The performance of different strategies (S1–S3) for each cell type, based on both cross-validation and

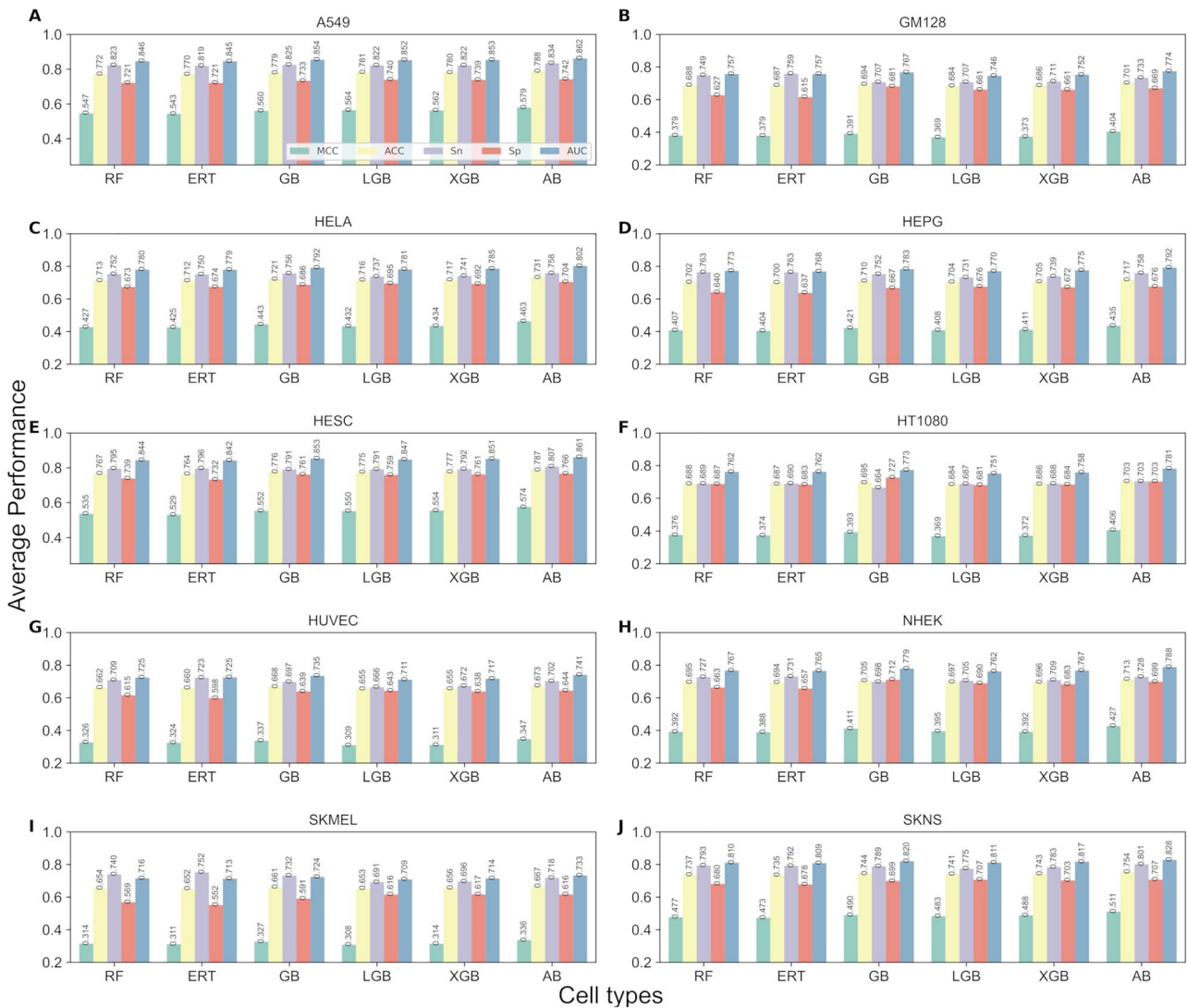


Figure 2. Average performance of tree-based algorithms with 10 different feature descriptors for 10 different cell types, based on the training datasets. Each of the A–J columns represents a different cell line, namely: A549, GM128, HELA, HEPG, HESC, HT1080, HUVEC, NHEK, SKMEL and SKNS.

independent assessments is shown in Table 2. We observed that in addition to AB, other classifiers (GB and XGB) achieved a superior performance for S1 in a few cell types. In the case of S2, most tree-based algorithms (except RF) were used at least once in the stacking approach. However, for S3, only three classifiers (RF, GB and ERT) were used for each cell type. These results suggest that it is important to conduct experiments using different tree-based classifiers while employing a variety of strategies. In terms of MCC, S3 consistently outperforms S2 and S1 approaches on the training datasets of the 10 cell types (Figure 4A). In contrast, S2 consistently performed better than both S1 and S3 approaches on independent datasets (Figure 4B). The S2 approach demonstrates greater consistency than the other two approaches. Therefore, we selected S2-based models for 10 cell types and named them TACOS.

Our objective was to improve the performance of the best baseline model. Therefore, we compared their performances with that of TACOS. To get an overview,

we computed the average metrics from 10 different cell types for both the baseline and TACOS models. Figure 5 shows that TACOS improved MCC by 1.48% during cross-validation and 2.61% during independent dataset validation, suggesting that the stacking strategy improved the overall predictive performance on both datasets.

Comparison of TACOS with the existing method

We compared the performance of TACOS with that of IncLocator 2.0, which is the only available cell-specific predictor. It is important to note that IncLocator 2.0 used different training datasets for model development, assessed using very small independent datasets and reported only AUC values [21]. Therefore, we compared the reported values with those of TACOS. Figure 6 shows that TACOS consistently outperformed IncLocator 2.0 across all 10 cell types. Specific improvements of AUC values for A549, GM128, HELA, HT108, HUVEC, HEPG, NHEK, SKMEL, SKNS and HESC are 1.1, 13.5, 7.6, 10.7, 10.6, 10.2, 6.4, 15.5, 9.0 and 3.4%, respectively. According

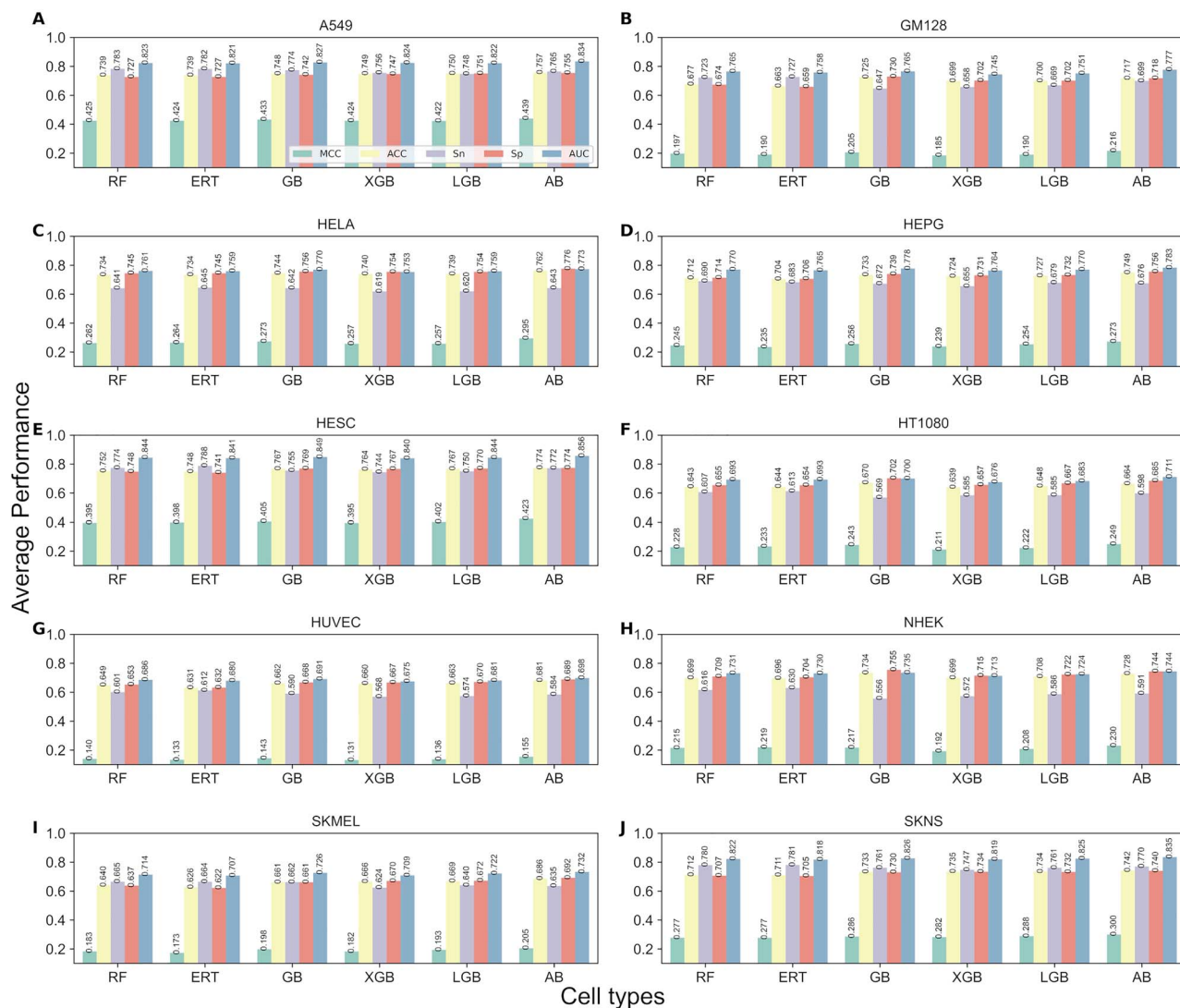


Figure 3. Average performance of tree-based algorithms with 10 different feature descriptors for 10 different cell types, based on the independent datasets. Each of the A-J) columns represents a different cell line, namely: A549, GM128, HELA, HEPG, HESC, HT1080, HUVEC, NHEK, SKMEL and SKNS.

to this analysis, TACOS's improvement is remarkable compared with the existing method.

Feature relevant analysis between the 10 probabilistic features

The TACOS model has 10 different feature descriptors, each of which was used to generate a probabilistic feature using an AB classifier. Next, we evaluated whether the 10 feature models were equally relevant to different species. Figure 7 illustrates a cluster heat map of the correlations between the 10 probabilistic features, which provides answers to the question raised above. The results showed that the probabilistic features based on F2, F3, F4, F5, F6 and KC descriptors were highly correlated with each other. Nevertheless, the two probabilistic features based on PseKNC and KC SNAP were moderately correlated with the other features. Accordingly, these features complement each other to improve the prediction accuracy. Meanwhile, it is evident that the pattern of feature correlation is similar among all the cell types.

Further analysis based on different cell types is required to confirm the levels of different features contributions in the final prediction.

Feature visualization

We used t-distributed stochastic neighbor embedding (t-SNE) to transform high-dimensional data into two-dimensional maps. On the training datasets, we applied t-SNE to hybrid features as well as to probabilistic features for all 10 cell types. Supplementary Figure S11 illustrates that the hybrid features of the positive and negative samples are highly overlap, regardless of the cell type. In contrast, the probabilistic features (predicted output of AB-based baseline models) depicted two distinct clusters, with little overlap between the two samples for A549 and HESC cells (Supplementary Figure S12). The feature distribution was directly correlated with the performance, where A549 and HESC produced ACC values of 0.823 and 0.813, respectively. The remaining cell types achieved an ACC in the range of 70–80.0%, but

Table 2. Comparison of the performance of different strategies for different cell types using the training and independent datasets

Cell types	Strategy	Classifier	Training					Independent				
			MCC	ACC	Sn	Sp	AUC	MCC	ACC	Sn	Sp	AUC
A549	S1	GB	0.616 (0.010)	0.806 (0.005)	0.854 (0.021)	0.759 (0.019)	0.885 (0.005)	0.452	0.752	0.803	0.739	0.836
	S2	AB	0.649 (0.010)	0.823 (0.004)	0.855 (0.019)	0.792 (0.013)	0.902 (0.002)	0.476	0.786	0.757	0.794	0.861
	S3	XGB	0.656 (0.003)	0.827 (0.001)	0.857 (0.014)	0.797 (0.011)	0.906 (0.001)	0.442	0.784	0.692	0.807	0.848
GM12878	S1	GB	0.416 (0.023)	0.706 (0.012)	0.774 (0.020)	0.638 (0.037)	0.784 (0.011)	0.227	0.701	0.756	0.697	0.787
	S2	XGB	0.466 (0.015)	0.732 (0.008)	0.746 (0.017)	0.719 (0.032)	0.816 (0.010)	0.312	0.809	0.734	0.814	0.832
	S3	GB	0.484 (0.013)	0.742 (0.006)	0.741 (0.007)	0.742 (0.015)	0.830 (0.008)	0.264	0.810	0.634	0.823	0.816
HELA	S1	AB	0.519 (0.017)	0.759 (0.009)	0.788 (0.001)	0.729 (0.011)	0.836 (0.010)	0.327	0.782	0.666	0.796	0.802
	S2	XGB	0.550 (0.018)	0.775 (0.009)	0.792 (0.015)	0.757 (0.009)	0.855 (0.009)	0.369	0.811	0.672	0.828	0.805
	S3	RF	0.556 (0.019)	0.778 (0.009)	0.792 (0.014)	0.764 (0.009)	0.862 (0.011)	0.358	0.814	0.643	0.834	0.818
HT1080	S1	AB	0.477 (0.021)	0.738 (0.010)	0.742 (0.038)	0.734 (0.023)	0.825 (0.007)	0.311	0.697	0.634	0.716	0.746
	S2	GB	0.493 (0.027)	0.746 (0.014)	0.735 (0.017)	0.758 (0.016)	0.835 (0.008)	0.327	0.718	0.608	0.753	0.762
	S3	RF	0.512 (0.010)	0.756 (0.005)	0.745 (0.016)	0.767 (0.014)	0.847 (0.006)	0.303	0.701	0.608	0.731	0.746
HUVEC	S1	AB	0.400 (0.023)	0.700 (0.011)	0.737 (0.046)	0.662 (0.034)	0.782 (0.012)	0.173	0.698	0.598	0.707	0.727
	S2	ERT	0.435 (0.035)	0.717 (0.017)	0.701 (0.029)	0.734 (0.022)	0.800 (0.013)	0.200	0.772	0.526	0.792	0.741
	S3	RF	0.438 (0.030)	0.719 (0.015)	0.704 (0.023)	0.733 (0.026)	0.808 (0.010)	0.182	0.755	0.524	0.774	0.738
HEPG	S1	AB	0.467 (0.010)	0.732 (0.005)	0.785 (0.016)	0.679 (0.016)	0.812 (0.007)	0.298	0.761	0.705	0.766	0.807
	S2	LGB	0.490 (0.007)	0.741 (0.004)	0.829 (0.019)	0.653 (0.022)	0.829 (0.006)	0.314	0.775	0.705	0.782	0.806
	S3	GB	0.517 (0.016)	0.758 (0.008)	0.785 (0.007)	0.731 (0.014)	0.841 (0.009)	0.307	0.787	0.663	0.800	0.810
NHEK	S1	AB	0.479 (0.021)	0.739 (0.011)	0.759 (0.018)	0.719 (0.030)	0.830 (0.010)	0.233	0.742	0.573	0.763	0.767
	S2	XGB	0.508 (0.022)	0.754 (0.010)	0.755 (0.019)	0.752 (0.014)	0.845 (0.010)	0.289	0.774	0.611	0.794	0.764
	S3	GB	0.522 (0.028)	0.761 (0.014)	0.758 (0.023)	0.764 (0.013)	0.850 (0.013)	0.207	0.749	0.516	0.777	0.733
SKMEL	S1	XGB	0.341 (0.013)	0.670 (0.006)	0.697 (0.027)	0.644 (0.030)	0.735 (0.015)	0.229	0.688	0.681	0.688	0.742
	S2	ERT	0.411 (0.025)	0.705 (0.012)	0.738 (0.020)	0.672 (0.006)	0.792 (0.009)	0.288	0.765	0.654	0.777	0.782
	S3	GB	0.426 (0.014)	0.712 (0.006)	0.752 (0.021)	0.673 (0.009)	0.796 (0.008)	0.237	0.736	0.615	0.749	0.770
SKNS	S1	AB	0.547 (0.010)	0.772 (0.005)	0.822 (0.010)	0.723 (0.010)	0.857 (0.004)	0.274	0.712	0.775	0.707	0.825
	S2	ERT	0.580 (0.009)	0.789 (0.004)	0.815 (0.004)	0.764 (0.008)	0.873 (0.002)	0.355	0.806	0.749	0.811	0.862
	S3	RF	0.583 (0.010)	0.791 (0.005)	0.819 (0.006)	0.763 (0.008)	0.874 (0.003)	0.333	0.798	0.725	0.804	0.853
HESC	S1	AB	0.613 (0.016)	0.806 (0.008)	0.821 (0.006)	0.791 (0.012)	0.885 (0.006)	0.452	0.795	0.775	0.799	0.873
	S2	ERT	0.626 (0.015)	0.813 (0.008)	0.825 (0.006)	0.801 (0.014)	0.894 (0.005)	0.474	0.808	0.783	0.813	0.881
	S3	ERT	0.642 (0.013)	0.821 (0.007)	0.832 (0.004)	0.810 (0.014)	0.898 (0.006)	0.435	0.797	0.737	0.807	0.841

The first column indicates the cell type. Each cell type consists of three different strategies whose performance on training and independent datasets is described. In the training metrics, two values indicate average values from cross-validation, and a standard error is included in brackets.

feature overlaps between positive and negative samples still exist, suggesting that adopting novel feature encoding will improve prediction performance and distinguish positively and negatively skewed samples.

Cross-model validation

It is common for a cell-specific model to perform exceptionally well with its own cell type. Specifically, we investigated whether a cell-specific model could be applied to other cell types. As shown in Figure 8, some cell-specific models had excellent transferability to other cell types, with MCC ≥ 0.50 . The GM12878 model demonstrated good performance in six other cell types, including A549, HELA, HT1080, HUVEC, HEPG and SKMEL. Similarly, the HELA model was transferrable to three other types: HEPG, HT1080 and GM12878. Additionally, A549, HT1080 and HUVEC models were transferrable to other two cell types, indicating that cell-specific models perform reasonably well on other types as well. However, the four cell-specific models (NHEK, SKMEL, SKNS and HESC) were not transferable to other types of cells, suggesting that cell specificity is the dominant characteristic of these lncRNA sequences. Based on our analysis, we found that highly accurate predictions required cell-specific models. Additionally, this provides a clue for developing a generic model by combining six cell types

(A549, GM12878, HELA, HT108, HEPG and NHEK). Generic models can also be applied to other cell types by compromising a slightly lower performance than cell-specific models.

Limitations of the current study

Although TACOS can predict cell-specific lncRNA localization, it has the following limitations:

(i) TACOS is an ML-based approach based on multiple manually derived features derived from sequences. It is widely recognized that the effectiveness of ML models is highly dependent on the feature representations used during training [29–31]. Accordingly, this study only considered composition and physicochemical properties. Further improvements in the prediction performance may be achieved by incorporating features based on other perspectives, including evolutionary information and novel features, based on extensive sequence analysis. Therefore, in future, we plan to explore and incorporate this information to improve the performance.

(ii) TACOS predicts subcellular localization based on the assumption that a given sequence belongs to a lncRNA. It is necessary to use another prediction model to identify lncRNAs from given RNA sequences before TACOS can be used. Currently, a few methods are available for identifying lncRNAs from RNA sequences

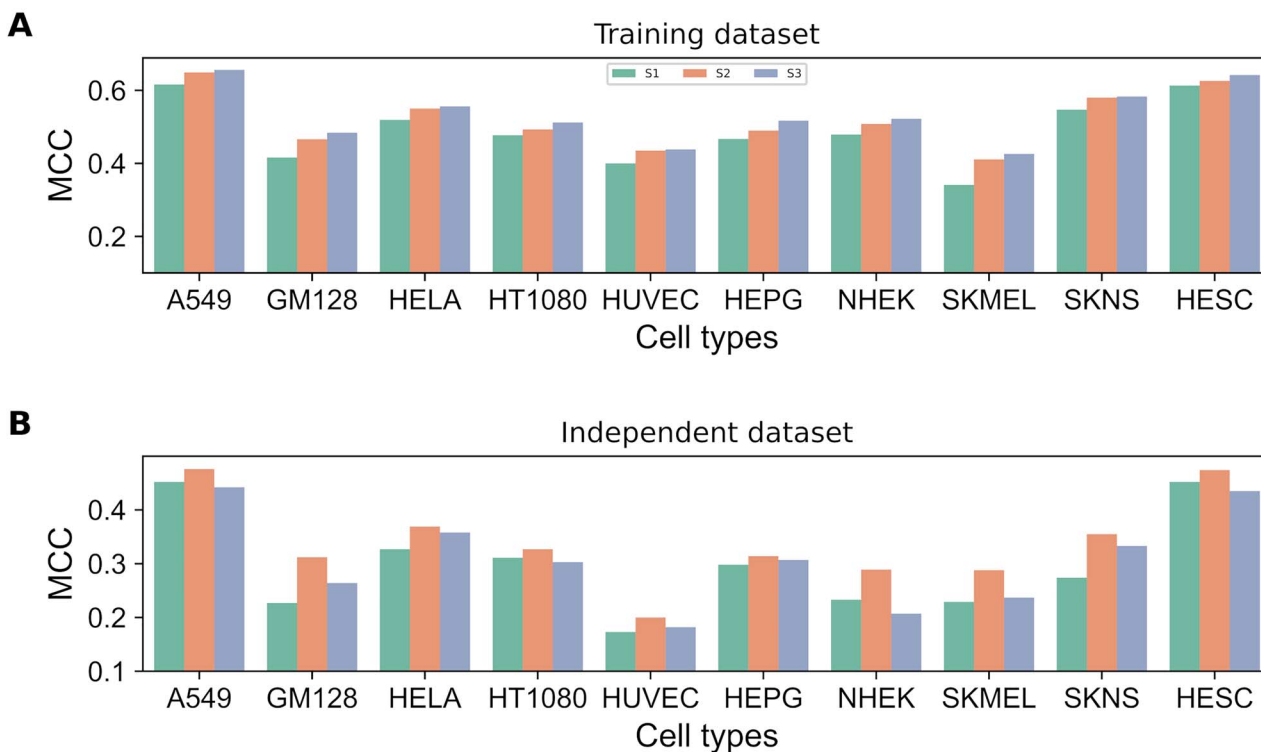


Figure 4. Comparison of the performance of optimal models derived from various strategies. Three different models [strategy 1 (S1), strategy 2 (S2) and strategy 3 (S3)] are compared based on cross-validation for each type of cell in (A) and independent assessments are shown in (B).

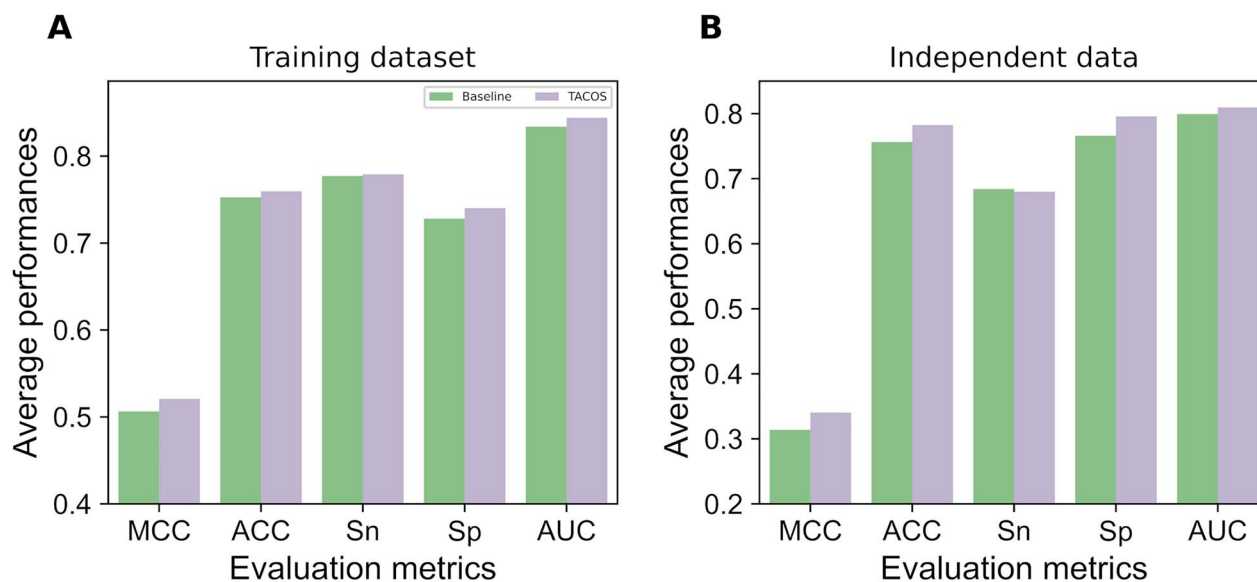


Figure 5. Performance comparison between baseline and TACOS models. (A) and (B) represent the average performance on 10 cell types over training and independent datasets, respectively.

[50, 51]. It is recommended to use these methods to identify lncRNAs before using TACOS. An additional prediction model will be integrated with TACOS in the future, able to identify lncRNAs and subcellular localization based upon the given mRNA sequence.

Conclusion

In this study, we present a method called TACOS that allows accurate identification of the subcellular local-

ization of human cell-specific lncRNAs using multiple feature encodings and tree-based algorithms. In order to identify the most suitable ML algorithm, we conducted a comprehensive performance evaluation of six different tree-based algorithms using 10 different feature descriptors. On average, AB-based baseline models performed well for all cell types in both cross-validation and independent assessment.

Subsequently, an optimal tree-based algorithm was utilized to construct the stacking model, using the

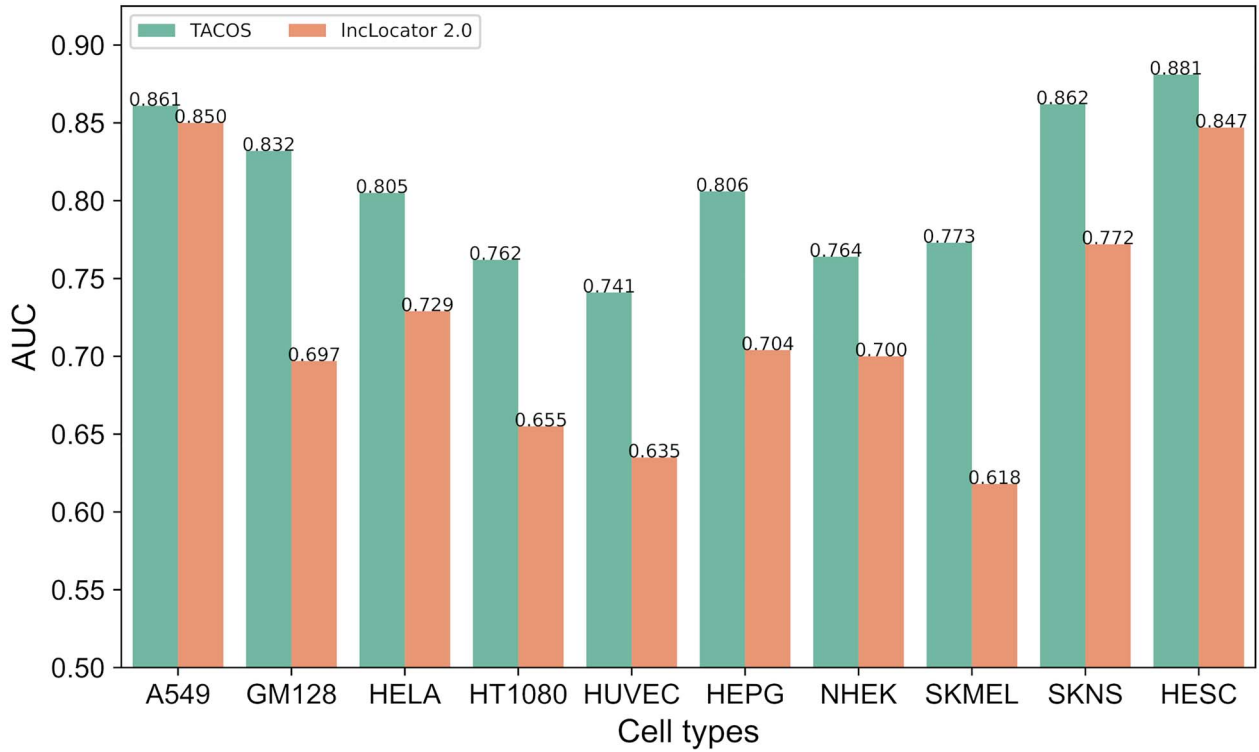


Figure 6. Performance comparison between TACOS and IncLocator 2.0 for different cell types.

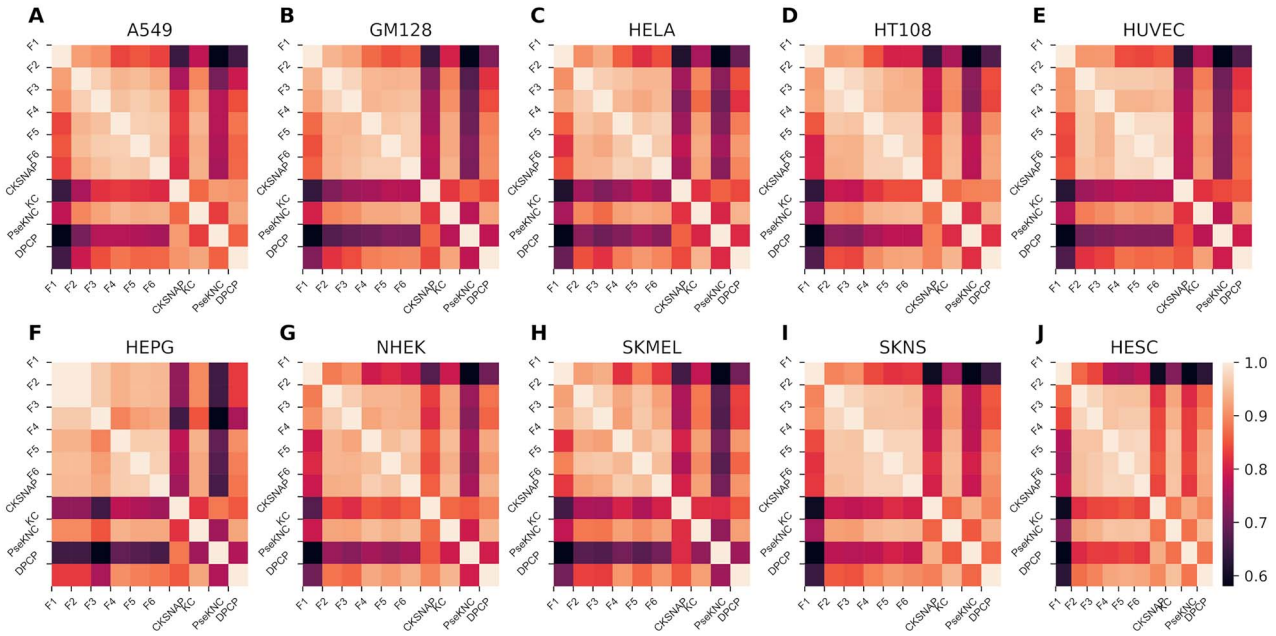


Figure 7. Cluster heatmap of the correlation between the 10 feature types in different cell types. (A-J) represents the correlation heatmap in A549, GM128, HELA, HT108, HUVEC, HEPG, NHEK, SKMEL, SKNS and HESC, respectively.

predicted probability values from 10 AB-based baseline models for each cell type to improve the prediction performance. Our approach differs from the previous methods because we developed TACOS using a balanced training dataset and evaluated it using a large imbalanced independent dataset. TACOS performed consistently well on both datasets (training and independent) compared with the two other strategies employed in this study.

The improved performance of TACOS is attributed to three factors: (i) the exploration of extensive feature descriptors that include different aspects of RNA sequence information, (ii) the selection of AB-specific baseline models and (iii) the identification of an appropriate classifier for building a stacking model that incorporates the strength of baseline models. It should be noted that the tree-based approaches employed

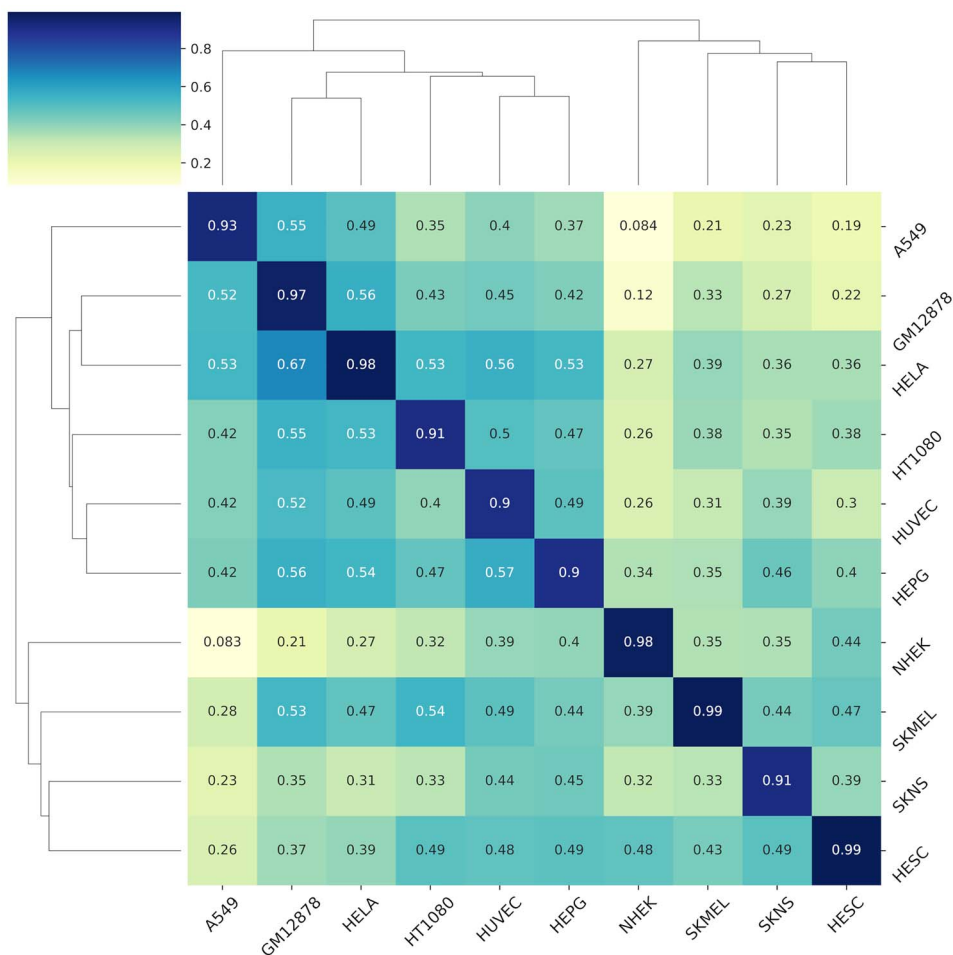


Figure 8. Cluster heat map showing the cell-specific prediction accuracies in terms of MCC. The cell-specific model was established on its own training dataset in columns, and it was validated on its own or other cell training datasets in rows.

in this study are applicable to other sequence-based function prediction problems [52], including enhancer prediction [53] and replication origin site prediction [54]. In future studies, a systematic approach similar to that used in this study will be employed if more than two locations with a reasonable dataset size are publicly available. Moreover, we developed the TACOS webserver and made them freely available at <https://balalab-skku.org/TACOS>. TACOS is expected to be an invaluable tool for experimentalists to identify the subcellular localization of lncRNAs, which will be useful for carrying out subsequent experiments to better understand its function.

Key Points

- A new computational framework known as TACOS was introduced and implemented as a web server for the prediction of cell-specific long noncoding RNA subcellular locations in the human genome.
- TACOS uses tree-based algorithms along with various sequence compositional and physicochemical features.
- Benchmarking experiments demonstrate that TACOS outperforms its constituent baseline models on both

training and independent datasets, thus making it a powerful generalization tool.

- The webserver of TACOS is publicly available at <https://balalab-skku.org/TACOS>.

Supplementary Data

Supplementary data are available online at <https://academic.oup.com/bib>

Funding

National Research Foundation of Korea (NRF) funded by the Korean government (MSIT) (2021R1A2C1014338, 2021R1C1C1007833).

Data Availability

Datasets used in this study, including training and independent datasets, are available at <https://balalab-skku.org/TACOS/download/>.

References

- Sun YM, Chen YQ. Principles and innovative technologies for decrypting noncoding RNAs: from discovery and functional prediction to clinical application. *J Hematol Oncol* 2020;**13**:109.
- Djebali S, Davis CA, Merkel A, et al. Landscape of transcription in human cells. *Nature* 2012;**489**:101–8.
- Derrien T, Johnson R, Bussotti G, et al. The GENCODE v7 catalog of human long noncoding RNAs: analysis of their gene structure, evolution, and expression. *Genome Res* 2012;**22**:1775–89.
- Quinn JJ, Chang HY. Unique features of long non-coding RNA biogenesis and function. *Nat Rev Genet* 2016;**17**:47–62.
- Memczak S, Jens M, Elefsinioti A, et al. Circular RNAs are a large class of animal RNAs with regulatory potency. *Nature* 2013;**495**:333–8.
- Lambert M, Benmoussa A, Provost P. Small non-coding RNAs derived from eukaryotic ribosomal RNA. *Noncoding RNA* 2019;**5**:16.
- Gao N, Li Y, Li J, et al. Long non-coding RNAs: the regulatory mechanisms, research strategies, and future directions in cancers. *Front Oncologia* 2020;**10**:598817.
- Fang Y, Fullwood MJ. Roles, functions, and mechanisms of long non-coding RNAs in cancer. *Genomics Proteomics Bioinformatics* 2016;**14**:42–54.
- Statello L, Guo CJ, Chen LL, et al. Gene regulation by long non-coding RNAs and its biological functions. *Nat Rev Mol Cell Biol* 2021;**22**:96–118.
- Uszczynska-Ratajczak B, Lagarde J, Frankish A, et al. Towards a complete map of the human long non-coding RNA transcriptome. *Nat Rev Genet* 2018;**19**:535–48.
- Fang S, Zhang L, Guo J, et al. NONCODEV5: a comprehensive annotation database for long non-coding RNAs. *Nucleic Acids Res* 2018;**46**:D308–14.
- Zhao Y, Wang CC, Chen X. Microbes and complex diseases: from experimental results to computational models. *Brief Bioinform* 2021;**22**.
- Hougaard DM, Hansen H, Larsson LI. Non-radioactive in situ hybridization for mRNA with emphasis on the use of oligodeoxynucleotide probes. *Histochem Cell Biol* 1997;**108**:335–44.
- Cabili MN, Dunagin MC, McClanahan PD, et al. Localization and abundance analysis of human lncRNAs at single-cell and single-molecule resolution. *Genome Biol* 2015;**16**:20.
- Raj A, van den Bogaard P, Rifkin SA, et al. Imaging individual mRNA molecules using multiple singly labeled probes. *Nat Methods* 2008;**5**:877–9.
- Lee JH, Daugharthy ER, Scheiman J, et al. Fluorescent in situ sequencing (FISSEQ) of RNA for gene expression profiling in intact cells and tissues. *Nat Protoc* 2015;**10**:442–58.
- Wang X, Allen WE, Wright MA, et al. Three-dimensional intact-tissue sequencing of single-cell transcriptional states. *Science* 2018;**361**.
- Ahmad A, Lin H, Shatabda S. Locate-R: subcellular localization of long non-coding RNAs using nucleotide compositions. *Genomics* 2020;**112**:2583–9.
- Cao Z, Pan X, Yang Y, et al. The lncLocator: a subcellular localization predictor for long non-coding RNAs based on a stacked ensemble classifier. *Bioinformatics* 2018;**34**:2185–94.
- Gudenas BL, Wang L. Prediction of lncRNA subcellular localization with deep learning from sequence features. *Sci Rep* 2018;**8**:16385.
- Lin Y, Pan X, Shen HB. lncLocator 2.0: a cell-line-specific subcellular localization predictor for long non-coding RNAs with interpretable deep learning. *Bioinformatics* 2021;**37**:2308–16.
- Su ZD, Huang Y, Zhang ZY, et al. iLoc-lncRNA: predict the subcellular location of lncRNAs by incorporating octamer composition into general PseKNC. *Bioinformatics* 2018;**34**:4196–204.
- Zhang T, Tan P, Wang L, et al. RNALocate: a resource for RNA subcellular localizations. *Nucleic Acids Res* 2017;**45**:D135–8.
- Blagus R, Lusa L. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics* 2013;**14**:106.
- Frankish A, Diekhans M, Ferreira AM, et al. GENCODE reference annotation for the human and mouse genomes. *Nucleic Acids Res* 2019;**47**:D766–73.
- Mas-Ponte D, Carlevaro-Fita J, Palumbo E, et al. LncAtlas database for subcellular localization of long noncoding RNAs. *RNA* 2017;**23**:1080–7.
- Fu L, Niu B, Zhu Z, et al. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* 2012;**28**:3150–2.
- Chen Z, Zhao P, Li F, et al. iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data. *Brief Bioinform* 2020;**21**:1047–57.
- Li HL, Pang YH, Liu B. BioSeq-BLM: a platform for analyzing DNA, RNA and protein sequences based on biological language models. *Nucleic Acids Res* 2021;**49**:e129.
- Liu B. BioSeq-Analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches. *Brief Bioinform* 2019;**20**:1280–94.
- 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.
- Basith S, Manavalan B, Shin TH, et al. SDM6A: a web-based integrative machine-learning framework for predicting 6mA sites in the rice genome. *Mol Ther Nucleic Acids* 2019;**18**:131–41.
- Manavalan B, Basith S, Shin TH, et al. Meta-4mCpred: a sequence-based meta-predictor for accurate DNA 4mC site prediction using effective feature representation. *Mol Ther Nucleic Acids* 2019;**16**:733–44.
- Chen W, Lei TY, Jin DC, et al. PseKNC: a flexible web server for generating pseudo K-tuple nucleotide composition. *Anal Biochem* 2014;**456**:53–60.
- Muhammad R, Ahmed S, Md Farid D, et al. PyFeat: a Python-based effective feature generation tool for DNA, RNA and protein sequences. *Bioinformatics* 2019;**35**:3831–3.
- Breiman L. Random forests. *Mach Learn* 2001;**45**:5–32.
- Geurts P, Ernst D, Wehenkel L. Extremely randomized trees. *Mach Learn* 2006;**63**:3–42.
- Charoenkwan P, Nantasenamat C, Hasan MM, et al. StackDP-PIV: a novel computational approach for accurate prediction of dipeptidyl peptidase IV (DPP-IV) inhibitory peptides. *Methods* 2021;**204**:189–98.
- Hasan MM, Alam MA, Shoombuatong W, et al. NeuroPred-FRL: an interpretable prediction model for identifying neuropeptide using feature representation learning. *Brief Bioinform* 2021;**22**.
- Xie R, Li J, Wang J, et al. DeepVF: a deep learning-based hybrid framework for identifying virulence factors using the stacking strategy. *Brief Bioinform* 2021;**22**.
- Basith S, Lee G, Manavalan B. STALLION: a stacking-based ensemble learning framework for prokaryotic lysine acetylation site prediction. *Brief Bioinform* 2022;**23**.
- Manavalan B, Basith S, Shin TH, et al. Computational prediction of species-specific yeast DNA replication origin via iterative feature representation. *Brief Bioinform* 2021;**22**.

43. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in Python. *J Mach Learn Res* 2011;**12**:2825–30.
44. Malik A, Subramaniyam S, Kim CB, et al. SortPred: the first machine learning based predictor to identify bacterial sortases and their classes using sequence-derived information. *Comput Struct Biotechnol J* 2022;**20**:165–74.
45. Dao FY, Lv H, Su W, et al. iDHS-Deep: an integrated tool for predicting DNase I hypersensitive sites by deep neural network. *Brief Bioinform* 2021;**22**.
46. Jiao Y, Du P. Performance measures in evaluating machine learning based bioinformatics predictors for classifications. *Quant Biol* 2016;**4**:320–30.
47. Basith S, Manavalan B, Hwan Shin T, et al. Machine intelligence in peptide therapeutics: a next-generation tool for rapid disease screening. *Med Res Rev* 2020;**40**:1276–314.
48. Chicco D, Jurman G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* 2020;**21**:6.
49. Li F, Guo X, Xiang D, et al. Computational analysis and prediction of PE_PGRS proteins using machine learning, ational and Structural. *Comput Struct Biotechnol J* 2022;**20**:662–674.
50. Cao L, Wang Y, Bi C, et al. PreLnc: an accurate tool for predicting lncRNAs based on multiple features. *Genes (Basel)* 2020;**11**:981.
51. Han S, Liang Y, Ma Q, et al. LncFinder: an integrated platform for long non-coding RNA identification utilizing sequence intrinsic composition, structural information and physicochemical property. *Brief Bioinform* 2019;**20**:2009–27.
52. Hasan MM, Tsukiyama S, Cho JY, et al. Deepm5C: a deep learning-based hybrid framework for identifying human RNA N5-methylcytosine sites using a stacking strategy. *Mol Ther* 2022.
53. Basith S, Hasan MM, Lee G, et al. Integrative machine learning framework for the identification of cell-specific enhancers from the human genome. *Brief Bioinform* 2021;**22**.
54. Wei L, He W, Malik A, et al. Computational prediction and interpretation of cell-specific replication origin sites from multiple eukaryotes by exploiting stacking framework. *Brief Bioinform* 2021;**22**.