GHENT UNIVERSITY

FACULTY OF PHARMACEUTICAL SCIENCES

FACULTY OF MEDICINE AND HEALTH SCIENCES

FACULTY OF VETERINARY MEDICINE
accredited by EAEVE

HONOURS PROGRAMME IN LIFE SCIENCES 2016-2017

Computational protein discovery: can the elimination of ORF length in coding potential prediction improve the detection of micropeptides?

Weyers Basiel

Faculty of Medicine and Health Sciences

Promotor: Dr. Pieter-Jan Volders

Copromotor: Prof. Dr. Lennart Martens

INDEX

ABSTRACT

Open Reading Frame (ORF) size is one of the most powerful parameters to predict the coding potential of a gene, its ability to encode a protein sequence (1). Our hypothesis was that this is caused by a bias in the existing algorithms, since the training data used in these programs are based on current annotations for protein coding genes, which show a strong bias towards proteins larger than 100 amino acids (2).

It was our goal to evaluate the effect of the omission of ORF size in protein coding potential prediction. We modified the Coding-Potential Assessment Tool or CPAT to exclude this parameter and built a new underlying logistic regression model. Afterwards, we evaluated the performance of both the original and our adapted algorithm by comparing them with the results of other algorithms and methods that are less affected by ORF size. All results were collected in a MySQL database. By performing analysis on these results, we were able to assess the real importance of ORF size as a predictor.

Our research shows that by eliminating ORF size as a parameter to predict coding probability, we can make a more accurate analysis of the coding probability of small ORFs. On longer ORFs however, the performance of the adapted algorithm was lower than that of the existing one. As we expected, ORF size is not a good parameter to use when predicting coding probability, especially because the slightly lower performance on longer ORFs is likely caused by a bias in current datasets. It is our recommendation to limit the use of ORF size as a predictor for coding probability, as is the case in most of the algorithms currently available.

1 INTRODUCTION

The central dogma of molecular biology states that DNA is transcribed into RNA, which is translated into protein. It later became clear that the dogma, first stated by Francis Crick in 1958 (3), only takes into account the about 22,000 protein coding genes in the human genome (4). These protein coding genes are transcribed into messenger RNAs (mRNA), whose sequences encode the amino acid sequences for a protein. However, the majority of the human genome is composed of non-coding regions, some of which are transcribed into what we call non-coding RNA. This means that these RNAs are not translated into protein. Instead, the RNA transcript itself or a derivate RNA molecule forms the functional product of the corresponding gene (5).

There are many types of non-coding RNA, including the highly abundant transfer RNAs (tRNA) and ribosomal RNAs (rRNA), but also microRNAs, snoRNAs, snRNAs and many more (5). One group of non-coding RNAs, called long non-coding RNAs or lncRNA, are particularly interesting as they are transcribed from regions that were previously regarded as genomic wasteland. Now we know that long non-coding RNAs are most likely the largest group of genes in the genome. These novel transcripts are long (> 200 nucleotides), multi-exonic and without conserved open reading frames (ORFs) (6). The majority of these lncRNAs are not yet functionally characterized.

The first step in understanding the function of a gene is to assess its coding potential, its ability to encode a protein sequence. While both non-coding and protein coding genes are interesting targets to study, the research strategy involved and the development as therapeutic target will differ. In order to predict coding potential, numerous prediction programs have been developed. In general, they have an excellent performance on the used test data (1). The problem is that the training data used in these programs are based on our current knowledge on protein coding genes, which shows a strong bias towards proteins larger than 100 amino acids (or ORFs larger than 300 nucleotides) with high evolutionary conservation (2). Indeed, the (relative) ORF size is often found to be the most powerful discriminator on typically used benchmarking sets (7). It is plausible that this is reflecting a bias in current annotation, as research and annotation groups have been primarily focusing on transcripts containing ORFs larger than 300 nucleotides (100 amino acids) (8). For instance, the FANTOM consortium originally used a cut-off of 300 nucleotides to help identify putative mRNAs (9).

There are many algorithms available to predict coding probability. Notable examples are CPAT (1), PhyloCSF (10), CPC (11), CONC (7), PORTRAIT (12), PLEK (13) and iSeeRNA (14). The first algorithm that we used is CPAT, or Coding-Potential Assessment Tool (rna-cpat.sourceforge.net/). It is an alignment-free method (thus based on the sequence alone and not its conservation) for distinguishing between coding and noncoding RNA, using a logistic regression approach. It uses four sequence features to achieve this: ORF size, ORF coverage, Fickett score and Hexamer score. CPAT comes with a logistic regression model for humans, which is built by comparing a dataset of human coding DNA to a dataset of human noncoding DNA. The coding sequences were obtained by selecting 10,000 protein-coding transcripts from the RefSeq database (1). The second algorithm used is PhyloCSF (github.com/mlin/PhyloCSF/wiki) which, in contrast to CPAT, is an alignment-based method for distinguishing between coding and noncoding RNA. It examines evolutionary signatures characteristic to alignments of conserved coding regions, such as the high frequencies of synonymous codon substitutions and conservative amino acid substitutions, and the low frequencies of other missense and non-sense substitutions (10).

To study the coding potential of atypical protein, several interesting datasets have recently become available. It was our goal to re-train CPAT using these datasets. Afterwards, we could compare the performance of the adapted algorithm to the original one, using datasets of both coding and non-coding small ORFs.

2 MATERIAL & METHODS

2.1 Coding potential calculation

All of the mentioned scripts can be found at github.com/basielw/HP_paper. Every script was written using Python 2.7.

2.1.1 CPAT

We adapted CPAT so that it evaluates every ORF within a transcript above a user-defined minimum size, instead of only the longest one. We did so by modifying the module that scans for ORFs within the transcripts, called ORF.py. By introducing a 'for loop', it will loop through every ORF within a gene and it will append all the ORFs that are longer than the user-defined minimum length to a list. Afterwards we had to do some minor changes to the main script, CPAT.py, in a separate copy of the original, which we called CPAT_modified.py. Doing so, we could still use the original one to compare the two. We also modified the script used to build a logistic regression model for CPAT, called make_logitModel.py into a copy, make_logitModel_modified.py.

2.1.2 PhyloCSF

To make PhyloCSF work, we had to provide it with the sequences we want to analyse (in a FASTA-file), aligned with the same sequence from 28 other mammals. To do so, we downloaded MAF-files (Multiple Alignment Format) containing these sequences from hgdownload.cse.ucsc.edu/goldenPath/hg19/multiz46way/. Afterwards, we had to retrieve the desired sequences from these files using a tool called MAF_parse, part of a package called Phast, which we downloaded from compgen.cshl.edu/phast/. Finally, the obtained MAF-files (one per gene) had to be converted into FASTA-files that can be used by PhyloCSF, using a tool called MSA_view, also part of Phast. We combined both tools into one Python script using the 'subprocess' module, called phyloCSF_maf.py. In addition, we provided a Docker image to easily deploy our pipeline on different servers.

2.2 Datasets used

2.2.1 Training dataset

We composed a dataset of micropeptides (proteins smaller than 100 amino acids or 300 nucleotides) by using the UniProt database (www.uniprot.org/). We filtered on 'Human' and 'Reviewed' proteins, as well as on protein length: up to 100 amino acids. We downloaded the list of UniProt IDs of all the 695 remaining proteins. Next, we acquired the corresponding gene sequences using Ensembl GRCh37 Biomart (grch37.ensembl.org/Homo_sapiens). We supplied it with the list of UniProt IDs by using the 'Input external references ID list: UniProt/SwissProt ID' function.

Additionally, we composed a dataset of non-coding small ORFs that have the same length of the longest ORF of every micropeptide in the micropeptides dataset. Using a script called make_fasta_from_orfs.py, we cut the non-coding sequences supplied with CPAT to the length of the micropeptide ORFs.

2.2.2 Validation dataset

The sORFs database (www.sORFs.org) (15) was used to validate our method. This database is a public repository of 3,551,506 small ORFs, including ribosome profiling data for many of them. We used the provided Biomart interface to filter on all biotypes except 'protein_coding', which resulted in 756,359 small ORFs, which we downloaded. We excluded the protein coding genes because they may already occur in our training dataset and additionally, we are looking for new ORFs outside of protein coding regions in particular.

2.2.3 Use case datasets

To show the usefulness of our methodology we applied it to two different datasets. As a first use case, we applied our approach to long non-coding RNA (lncRNA) transcripts retrieved from lncipedia.org, a database of 146,742 human annotated lncRNAs. We downloaded the GRCh37/hg19 transcripts in a BED-file.

As a second use case, we used the PANcancer dataset. The PANcancer transcriptome dataset is built from large-scale RNA sequencing efforts in the research group of prof. Jo Vandesompele and prof. Pieter Mestdagh at the Center for Medical Genetics Ghent. When doing descriptive analysis on the mRNA size of the PANcancer transcripts, we noticed that there where many outliers. Therefore, we removed all transcripts with an mRNA size over 22,492, determined using the 'Tukey's fences' method (16).

2.3 MySQL database with results

To store the transcripts, the different ORFs within the transcripts and the corresponding CPAT and PhyloCSF scores of every ORF, we created two MySQL (version 14.14) databases: one for the LNCipedia and sORFs.org data and one for PANcancer transcriptome data. Both databases use more or less the same schema, which you can find below. The only major difference is that the PANcancer database does not contain the table with ribosome profiling data (Rib_prof), because that data is not available for the PANcancer dataset. We used MySQL Workbench version 6.3.7 and Python 2.7 with the MySQLdb module to perform queries.

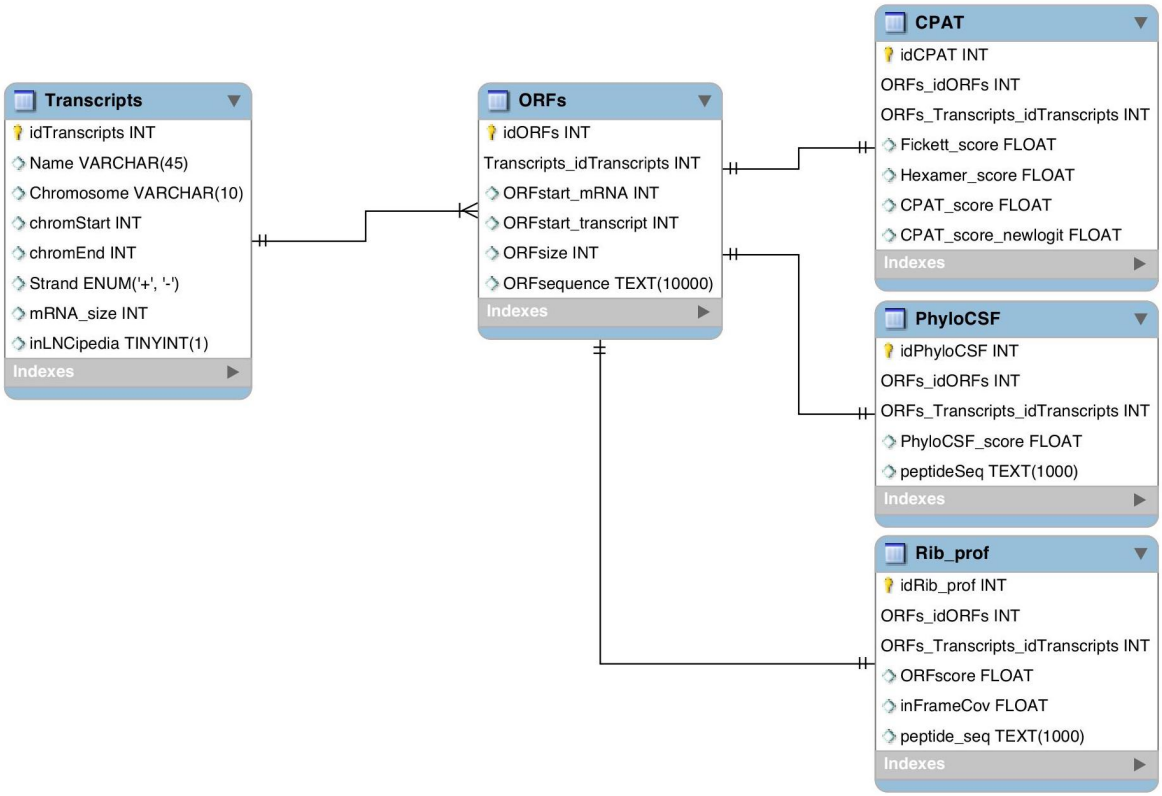

Figure 1. MySQL schema for the PANcancer and small ORFs database.

For both databases, we started by importing the transcripts from the BED-files to the table 'Transcripts', using a script called general_import_transcripts.py. We used the CPAT output to import all ORFs into the 'ORFs' table, along with the computed coding probability from both the old and the new logistic regression model into the 'CPAT' table, using a script called general_import_CPAT.py. Finally, we imported the PhyloCSF output into the table 'PhyloCSF', using a script called general_import_PhyloCSF.py. For this last step, we used the ORF location that PhyloCSF provides to link every PhyloCSF score to the correct ORF.

For the small ORFs database, we imported the ribosome profiling data from sORFs.org. We started by converting the starting points of each ORF within the chromosome from GRCh38/hg38, used by sORFs.org to GRCh37/hg19, used in our database. We achieved this by retrieving the starting location of the transcripts from both GRCh37 and GRCh38 from the Ensembl database in a script called sorfs_convert_locations.py.

Because many of the positions were missing for the sORFs.org transcripts, we converted the location of the original LNCipedia ORFs within the mRNA (in the column "ORFstart_mRNA") to the location within the transcript (in the column "ORFstart_transcript"), using a script called sorfs_convert_ORFstart.py. Doing this, we could link up every ribosome profiling score with the correct ORF from LNCipedia. Afterwards, we imported the transcripts from sORFs.org. We linked them to the existing LNCipedia transcripts by comparing the start- and endpoints of the sORFs.org transcripts with the ones from LNCipedia. However, 553,336 from the 756,359 ORFs from sORFs.org (73%) were from a transcript that is not in the LNCipedia database. For these ORFs, we kept the original Ensembl IDs and for the corresponding transcripts, we gave them a '0' in the column "inLNCipedia", meaning they are not present in the LNCipedia database. The other transcripts got a '1' in that column. We achieved all of this using a script called sorfs_import_rib_prof.py. This script also imports all the ORFs that are not yet in the database, as well as the ribosome profiling ORFscore from every ORF.

Finally, we imported the CPAT and PhyloCSF output from the ORFs from sORFs.org that were not yet in LNCipedia using sorfs_import_CPAT_update.py and sorfs_import_PhyloCSF_update.py.

2.4 Analysis of the data

To evaluate the performance of the newly built logistic regression model, we performed a k-fold cross validation, using the same dataset that was used to train the model. When using the same training data as testing data, cross validation is important because it prevents overfitting. The script we created for this purpose, called cross_validation_CPAT.py, divides both the coding and non-coding transcripts in k parts. Afterwards, it uses k-1 of the parts to train CPAT, or build a new logistic regression model and uses the remaining part of both datasets to test the newly built model. This is repeated k times so that every sequence in either dataset is used exactly once to test the model, but never in the same round where it is used to train the model. In our case, we performed a 10-fold cross validation, only looking at the longest ORFs of each transcript.

After storing the results into the database, we analysed the data using R version 3.3.2. The receiver operating characteristic (ROC) curves, the precision-recall (PR) curves and the accuracy versus cut-off curves were plotted using the R package 'ROCR' (version 1.0.7). The distribution curves where plotted using the function 'sm.density.compare' from the package 'sm' (version 2.2.5.4). Finally, the Venn-Euler diagram was plotted using the function 'Vcombo' from the package 'Vennerable' (version 3.1.0.9000).

3 RESULTS AND DISCUSSION

3.1 Original CPAT algorithm underperforms for small ORFs

First, we wanted to test the performance of the original CPAT on small ORFs, using a dataset containing curated small (< 100 AA) proteins from UniProt, along with a dataset of size matched non-coding sequences. Doing this, the algorithm cannot rely on ORF size as a predictor. We ran CPAT using the original logistic regression model and only looking at the longest ORF within each transcript.

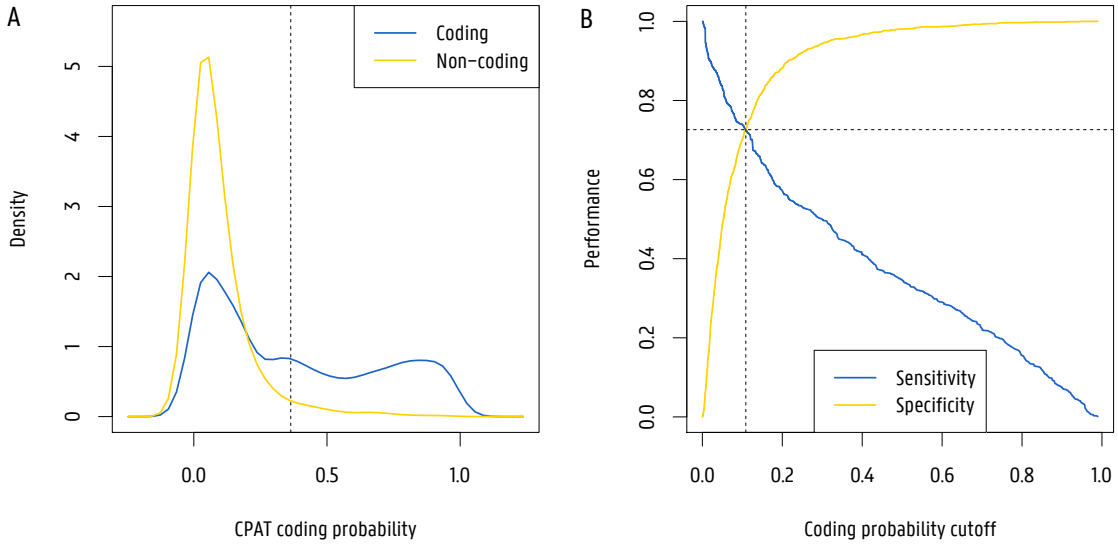

*Figure 2. Performance evaluation of the original CPAT algorithm on small ORFs. (**A**) Distribution of CPAT coding probability scores using the original algorithm shows good specificity but poor sensitivity. Dotted line represents the original cut-off value. (**B**) Two-graph ROC curve reveals new optimum cut-off value of 0.109 at a sensitivity and specificity of 0.726, pictured by the dotted lines.*

In the original CPAT paper, an optimal cut-off value of 0.364 is put forward (pictured by the dotted line on Figure 2). This means that all the ORFs with a coding probability score of less than 0.364 are considered non-coding, while the ORFs with a score above 0.364 are considered coding (1). Thus, if the algorithm would perform perfectly, all of the non-coding transcripts should have a coding probability less than 0.364, while all of the coding transcripts should have a coding probability above 0.364. From figure 2A, it becomes obvious that, with the original cut-off of 0.364, most of the non-coding transcripts are interpreted correctly (0.96, specificity), while less than half of the coding transcripts (the micropeptides) are interpreted as being coding (0.44, sensitivity). The reported sensitivity and specificity are however much higher (both 0.966) at the original cut-off (1). This is as we expected: because ORF length is such an important discriminator in the CPAT model and the micropeptides are much shorter than the protein coding genes that were used to build the original logistic regression model, the sensitivity of model is much worse (1).

This is also displayed in figure 2B. The new optimal cut-off, when allocating the same importance to sensitivity and specificity, would be 0.109 with sensitivity and specificity both being 0.726, displayed by the dotted lines on the graph. Hence, even when using a different cut-off, the sensitivity and specificity are still lower than the original research (0.966). The Area Under the Curve (AUC) of the current test was 0.780, compared to 0.993 in the original research (1). As a result, the original CPAT algorithm is not suitable for computational discovery of novel micropeptides.

3.2 Re-training the CPAT model for small ORFs

Because CPAT performs so poorly on small ORFs, it was our goal to adapt the algorithm and re-train the underlying logistic regression model to counter this problem. First, we adapted CPAT so that it analyses all ORFs within a transcript instead of only the longest one. Indeed, for most protein coding transcripts, it is the longest ORF that is coding. However, ORFs can occur by chance and there are cases where the coding ORF is not the longest within a gene (17), especially when working with small ORFs. Second, we built our own logistic regression model using both the dataset of micropeptides (coding small ORFs) and the dataset of non-coding small ORFs of the exact same length. By doing this, we were able to completely disable the parameter 'ORF length'.

3.2.1 Using a 10-fold cross validation

To test the performance of the newly built logistic regression model, we used a 10-fold cross validation, only looking at the longest ORF of each transcript in the dataset.

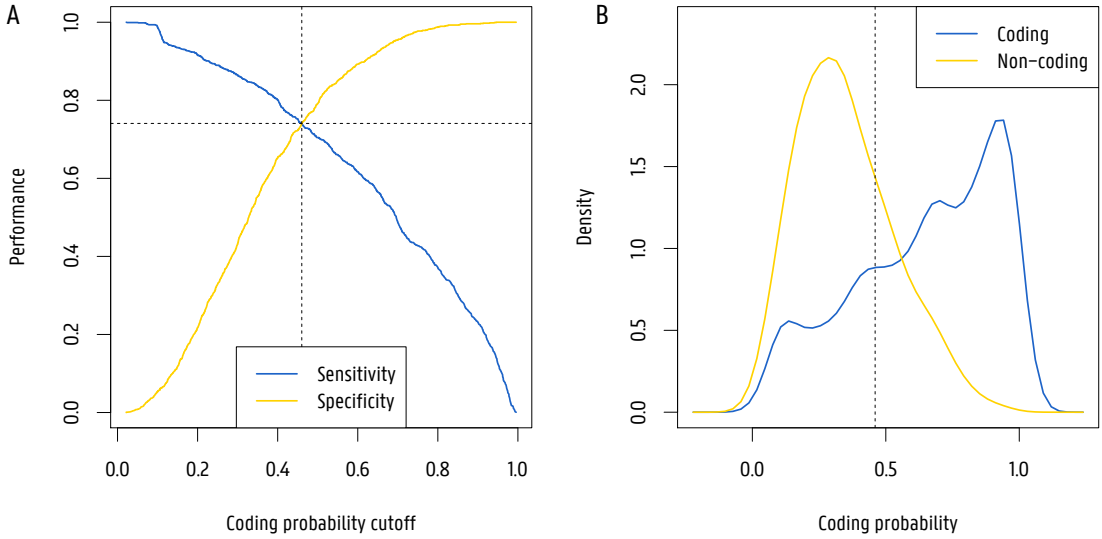

Figure 3. Performance evaluation of the newly built CPAT logistic regression model using a 10-fold cross validation. The two-graph ROC curve shows decreased specificity, but improved sensitivity when compared to the original logistic regression model on the same data (see Figure 2B). Dotted lines represent the new optimal cut-off value of 0.460 at a sensitivity and specificity of 0.741.

When we compare the two-graph ROC curve on figure 3A to the one on figure 2, using the original logistic regression model, it becomes obvious that the sensitivity has improved, while the specificity has decreased. The new optimal cut-off for this logistic regression model is 0.460, at a sensitivity and specificity of 0.741. When we plot this cut-off on the distribution curve on figure 3B, we can indeed observe that less of the non-coding transcripts and more of the coding transcripts are being interpreted correctly. The Area Under the Curve (AUC) is now 0.806, which is slightly better than the performance of the original logistic regression model on the small ORFs.

Even though the performance of the newly built logistic regression model on small ORFs is only slightly better when comparing it to the original logistic regression model, this provides valuable information. In fact, this means that, for small ORFs, the performance of CPAT is about the same with or without ORF size as a predictor. This proves that ORF size is not a valuable discriminator to use on small ORFs, as we expected. Additionally, we believe that the performance of the adapted algorithm can be improved by increasing the training set size. This time, we used 1,242 ORFs for the coding dataset, along with the same number of ORFs for the non-coding dataset.

3.2.2 The re-trained CPAT model outperforms the original model on ribosome profiling data

After re-training CPAT we put it to the test by using a much bigger dataset of putative small ORFs (www.sORFs.org) predicted by ribosome profiling data. Ribosome profiling is based on deep sequencing of ribosome-protected mRNA fragments. Although many non-coding RNAs show ribosome occupancy, by using initiation-specific translation inhibitors in combination with ribosome profiling, researchers were able to map translation initiation sites (TIS) with base pair resolution and improve the detection of true ORFs (18, 19). The coding probability of each ORF derived from these ribosome profiling experiments is expressed with an ORFscore. A score of 6.044 is used as a cut-off to discriminate coding from non-coding ORFs (20).

We analysed all the transcripts from LNCipedia and sORFs.org with CPAT using both the original logistic regression model and the newly built logistic regression model from the micropeptides. Both times, we analysed all ORFs longer than 10 nucleotides within each transcript.

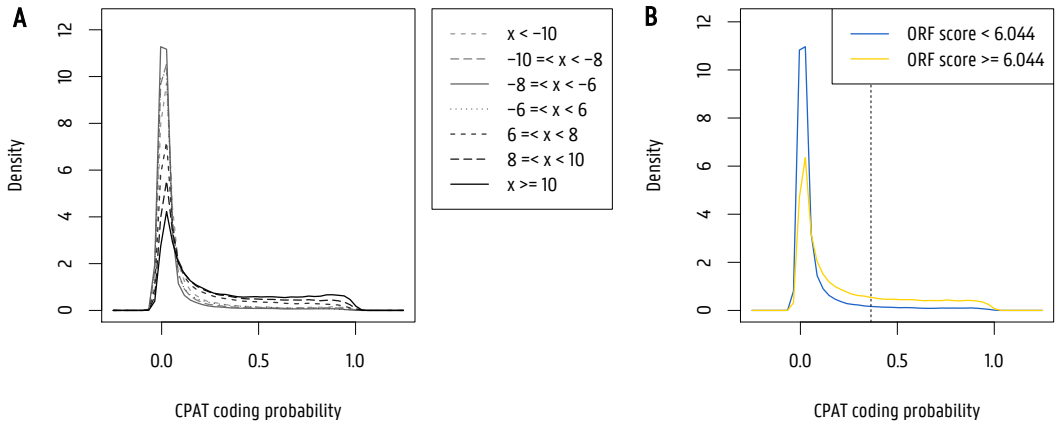

*Figure 4. Performance evaluation of CPAT using **the original logistic regression model** on ribosome profiling data. (A) Higher ORFscores are not well reflected in the CPAT coding probability. X equals the ORFscore. (B) When splitting up the ORFscores by their cut-off value of 6.044, the original CPAT scores show a good specificity, but a poor sensitivity. Dotted line represents the original CPAT cut-off value of 0.364.*

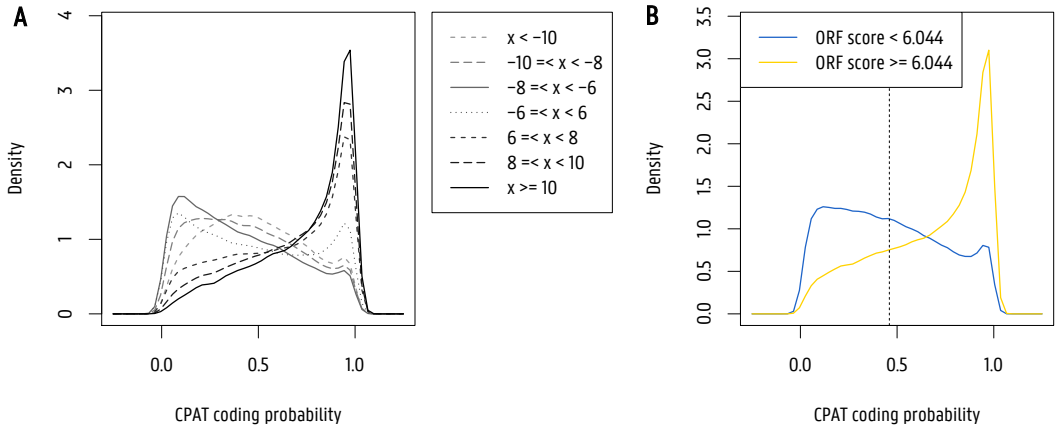

Figure 5. Performance evaluation of CPAT using the new logistic regression model on ribosome profiling data. (A) ORFs with higher ORFscores also have a higher CPAT coding probability and vice versa. X equals the ORFscore. (B) The new CPAT scores show a major decrease in specificity, but a raise in sensitivity. Dotted line represents the new CPAT cut-off value of 0.460.

When we compare the ribosome profiling scores with the CPAT scores, using the original logistic regression model, we obtain a specificity of 0.936 and a sensitivity of 0.263. The area under the curve was 0.731. However, using the newly built one, we observed a reduced specificity of 0.535, but a greatly improved sensitivity of 0.751. The area under the curve was 0.712. Even though the nature of the ribosome profiling data and ORFscore is very different compared to the used training data, our new model is an excellent predictor for ORFs with a high ORFscore.

3.3 Applying the re-trained CPAT model to large transcriptomes reveals potential to use it for discovery of novel micropeptides

We used two large transcriptomes to show the application of our new model in a real-world scenario. Additionally, this allows us to compare its performance with another coding potential prediction algorithm, PhyloCSF.

3.3.1 LNCipedia

First of all, we analysed all lncRNAs from LNCipedia.org. The lncRNA transcripts in this database have been filtered to contain only transcripts lacking large ORFs. As such, it is an excellent dataset to evaluate the ability of our model to predict putative small ORFs. All transcript in the database are analysed with CPAT, using both the original and newly built logistic regression model, as well as using PhyloCSF. As mentioned before, PhyloCSF is an alignment-based method based on conservation of codons to calculate the coding probability of a given transcript. Because of this, the score is less influenced by the ORF length. A PhyloCSF score above 60.7876 means that the ORF is likely to be coding (10).

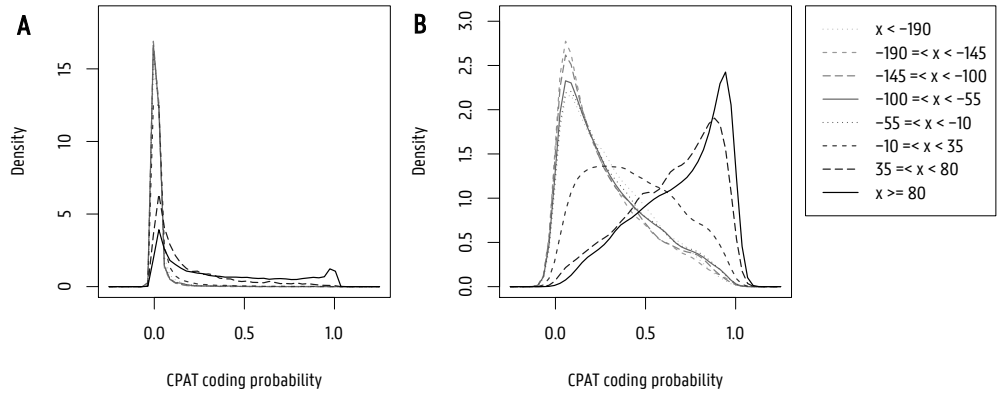

*Figure 6. Performance comparison between PhyloCSF and CPAT using **the original logistic regression model** (A), where higher PhyloCSF scores are not well reflected in the CPAT coding probability, and **the new logistic regression model** (B), where higher PhyloCSF scores correspond to higher CPAT coding probability. X equals the PhyloCSF score.*

When we compare the PhyloCSF scores to the CPAT scores, we observe that the analysis of CPAT using the new logistic regression model is more in line with the analysis of PhyloCSF, when comparing it with the analysis of CPAT using the original regression model. Indeed, with the new model, higher PhyloCSF scores also correspond to a higher CPAT coding probability, which wasn't the case when using the original logistic regression model. This is not surprising as LNCipedia transcripts are filtered for transcripts containing large ORFs and again shows that our new model outperforms the original model in predicting coding potential in small ORFs.

3.3.2 PANcancer

As a second application, we used the PANcancer transcriptome dataset to compare the analysis of CPAT and PhyloCSF. This large RNA sequencing dataset provides an unbiased view of the transcriptome as it contains both coding and non-coding transcripts. In addition, it contains a large number of uncharted transcripts and can thus be used to evaluate the usability of the new CPAT model to discover novel protein coding genes. Identifying new oncogenes in this dataset might be interesting for therapeutic purposes.

We executed PhyloCSF on the PANcancer transcripts, which left us with 906,153 ORFs with a computed PhyloCSF score. We compared these scores with the corresponding CPAT scores, both with the original and newly built logistic regression model.

Because the PANcancer dataset also contains protein coding genes, the average ORF length is much higher in comparison to the LNCipedia.org dataset which only contains lncRNAs. In fact, 417,412 of the 906,153 (46%) remaining ORFs were larger than 300 nucleotides. For the purpose of testing our model, we did a separate analysis on ORFs limited to a length of 300 nucleotides, thus leaving 488,741 ORFs (54%).

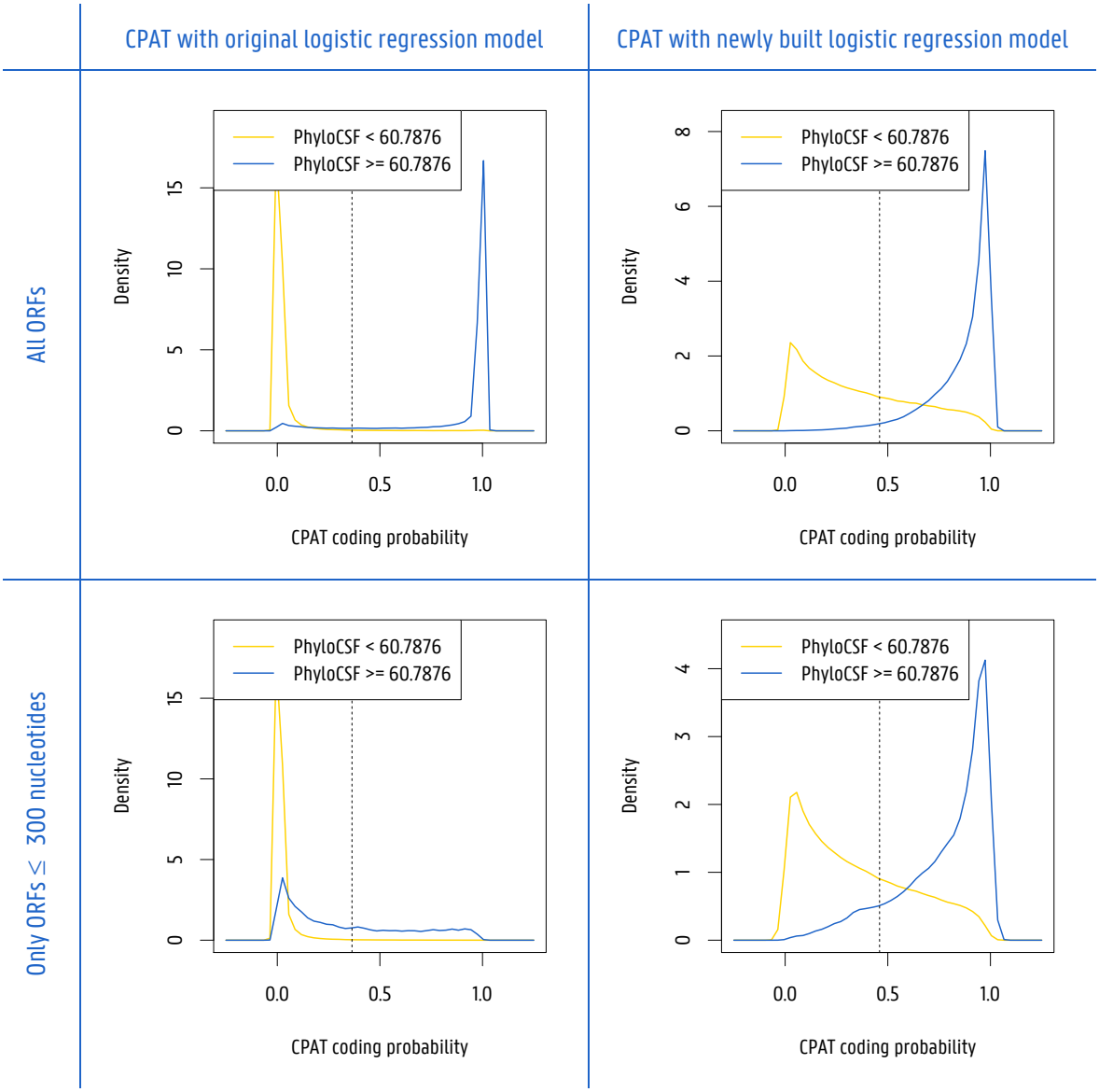

Figure 7. Comparison of PhyloCSF and CPAT scores of the PANcancer dataset. When we look at all ORFs, regardless of their length, the scores of CPAT using original logistic regression model are better in line with the PhyloCSF scores. However, when we limit the ORF size to 300 nucleotides, the newly built logistic regression model performs better. PhyloCSF scores are divided by their cut-off of 60.7876. Dotted lines represent the CPAT cut-off, being 0.364 for the original logistic regression model and 0.460 for the newly built one.

When including all ORFs, regardless of their length, it becomes clear that the scores of CPAT using the original logistic regression model are slightly better in line with the PhyloCSF scores if we compare them to the scores of CPAT using the newly built logistic regression model. However, when we limit the ORF length to 300 nucleotides, the CPAT scores from the original logistic regression model are not as much in line with the ones from PhyloCSF. Especially the ORFs that PhyloCSF considers coding (having a score above 60.7876) are often regarded by CPAT as being non-coding. This shows again that the original logistic regression model is heavily influenced by ORF length to determine coding probability. The performance of the newly built logistic regression model however is not influenced by the limit in ORF length, with the graph being almost identical to the one without the limit. This again underscores that it works independent of the ORF length to predict coding probability.

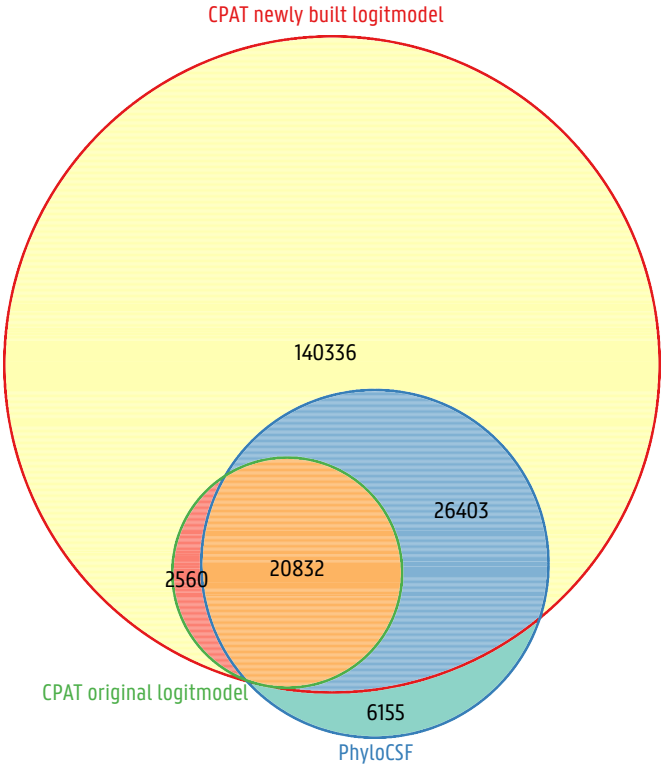

Figure 8. Venn diagram showing the amount of ORFs that are considered coding, thus having a PhyloCSF score above 60.7876, a CPAT score above 0.364 for the original logistic regression model or a CPAT score above 0.460 for the newly built logistic regression model. Only ORFs with a length smaller than or equal to 300 nucleotides are included. The original logistic regression model misses 32,558 of the 53,390 ORFs that are considered coding by PhyloCSF (61%). The newly built logistic regression model only misses 6,155 (11%). Additionally, it considers 142,896 additional ORFs as being coding that PhyloCSF considers non-coding.

When we examine the number of ORFs that are considered coding by each algorithm, it becomes clear how many more ORFs can be found by CPAT when eliminating ORF length as a parameter to predict coding probability. Compared to the original CPAT model and PhyloCSF, many new ORFs are detected that would have been overlooked otherwise. Given the excellent specificity and sensitivity of the model, it is unlikely that this large number can be attributed to false positives alone.

4 CONCLUSION

Our goal was to evaluate the effect of the omission of ORF size as a parameter in coding potential prediction, because of the possible bias which might explain the great power of this parameter on typically used benchmarking sets (8). Therefore, we re-trained the Coding-Potential Assessment Tool or CPAT to not include ORF size as a predictor for coding probability. Afterwards, we compared the performance of the adapted algorithm to the original one, using datasets of both coding and non-coding ORFs.

Our research shows that by eliminating ORF size as a parameter to predict coding probability, we can make a more accurate analysis of the coding probability of small ORFs. Thus, ORF size is not a good parameter to predict coding probability, confirming our hypothesis. There are various possible explanations for this. For one, bona fide long non-coding RNAs will by chance contain putative ORFs that are quite long (8). Secondly, there are also many proteins of <100 amino acids in size that may be incorrectly classified as non-coding RNAs (8). The potential scale of such errors is significant, given recent estimates that the mammalian proteome contains 3,700 proteins below this size (2). By using ORF size as such an important parameter, very small proteins or micropeptides might be missed. Nonetheless, these micropeptides can be stable and functional. A good example of this is the tarsal-less (tal) gene, that controls tissue folding in Drosophila and encodes a ~1.5 Kb transcript (21), whose putative ORFs are all extremely short. Tal was therefore initially classified as a non-coding RNA (22), but it has subsequently been shown that it is actually translated into multiple 11 amino acids peptides that fulfil the function of the gene (21). It can be expected that many more micropeptides exist and our algorithm has the potential to be an important contributor to their discovery. Of course, it is unclear how many of the over 140,000 additional ORFs that are found by our adapted algorithm over the original one and PhyloCSF are indeed functional. It is likely that this is an overestimate, but naturally, more research is necessary on this topic.

Unfortunately, with the current algorithm, we lost some of the performance on bigger ORFs. Nonetheless, with some optimizations, we believe that it is possible to create an algorithm that is even more accurate, also on the bigger ORFs. For example, these optimizations could include an increase in training set size. The small loss in performance on longer ORFs should not be a reason to continue using ORF size as a parameter, especially because it is likely based on a bias in current annotations. More research is necessary on the use of ORF size as a predictor for coding probability, but this could be a first step in the elimination of this parameter in future coding probability prediction programs.

Compared to PhyloCSF, our new algorithm is even less influenced by ORF size. Additionally, it is much faster and easier to execute, because there is no need to align the transcripts. Finally, our database can be an interesting resource to discover new proteins and oncogenes with therapeutic potential.

REFERENCES

1. Wang L, Park HJ, Dasari S, Wang S, Kocher JP, Li W. CPAT: Coding-Potential Assessment Tool using an alignment-free logistic regression model. Nucleic Acids Res. 2013;41(6):e74.
2. Frith MC, Forrest AR, Nourbakhsh E, Pang KC, Kai C, Kawai J, et al. The abundance of short proteins in the mammalian proteome. PLoS Genet. 2006;2(4):e52.
3. Crick F. Central dogma of molecular biology. Nature. 1970;227(5258):561-3.
4. UniProt: the universal protein knowledgebase. Nucleic Acids Res. 2017;45(D1):D158-D69.
5. Eddy SR. Non-coding RNA genes and the modern RNA world. Nat Rev Genet. 2001;2(12):919-29.
6. Derrien T, Johnson R, Bussotti G, Tanzer A, Djebali S, Tilgner H, et al. The GENCODE v7 catalog of human long noncoding RNAs: analysis of their gene structure, evolution, and expression. Genome Res. 2012;22(9):1775-89.
7. Liu J, Gough J, Rost B. Distinguishing protein-coding from non-coding RNAs through support vector machines. PLoS Genet. 2006;2(4):e29.
8. Dinger ME, Pang KC, Mercer TR, Mattick JS. Differentiating protein-coding and noncoding RNA: challenges and ambiguities. PLoS Comput Biol. 2008;4(11):e1000176.
9. Okazaki Y, Furuno M, Kasukawa T, Adachi J, Bono H, Kondo S, et al. Analysis of the mouse transcriptome based on functional annotation of 60,770 full-length cDNAs. Nature. 2002;420(6915):563-73.
10. Lin MF, Jungreis I, Kellis M. PhyloCSF: a comparative genomics method to distinguish protein coding and non-coding regions. Bioinformatics. 2011;27(13):i275-82.
11. Kong L, Zhang Y, Ye ZQ, Liu XQ, Zhao SQ, Wei L, et al. CPC: assess the protein-coding potential of transcripts using sequence features and support vector machine. Nucleic Acids Res. 2007;35(Web Server issue):W345-9.
12. Arrial RT, Togawa RC, Brigido Mde M. Screening non-coding RNAs in transcriptomes from neglected species using PORTRAIT: case study of the pathogenic fungus Paracoccidioides brasiliensis. BMC Bioinformatics. 2009;10:239.
13. Li A, Zhang J, Zhou Z. PLEK: a tool for predicting long non-coding RNAs and messenger RNAs based on an improved k-mer scheme. BMC Bioinformatics. 2014;15:311.
14. Sun K, Chen X, Jiang P, Song X, Wang H, Sun H. iSeeRNA: identification of long intergenic non-coding RNA transcripts from transcriptome sequencing data. BMC Genomics. 2013;14 Suppl 2:S7.
15. Olexiouk V, Crappe J, Verbruggen S, Verhegen K, Martens L, Menschaert G. sORFs.org: a repository of small ORFs identified by ribosome profiling. Nucleic Acids Res. 2016;44(D1):D324-9.
16. Tukey JW. Exploratory data analysis. 1977.
17. Anderson DM, Anderson KM, Chang CL, Makarewich CA, Nelson BR, McAnally JR, et al. A micropeptide encoded by a putative long noncoding RNA regulates muscle performance. Cell. 2015;160(4):595-606.
18. Ingolia NT, Lareau LF, Weissman JS. Ribosome profiling of mouse embryonic stem cells reveals the complexity and dynamics of mammalian proteomes. Cell. 2011;147(4):789-802.
19. Lee S, Liu B, Lee S, Huang SX, Shen B, Qian SB. Global mapping of translation initiation sites in mammalian cells at single-nucleotide resolution. Proc Natl Acad Sci U S A. 2012;109(37):E2424-32.
20. Bazzini AA, Johnstone TG, Christiano R, Mackowiak SD, Obermayer B, Fleming ES, et al. Identification of small ORFs in vertebrates using ribosome footprinting and evolutionary conservation. EMBO J. 2014;33(9):981-93.
21. Galindo MI, Pueyo JI, Fouix S, Bishop SA, Couso JP. Peptides encoded by short ORFs control development and define a new eukaryotic gene family. PLoS Biol. 2007;5(5):e106.
22. Tupy JL, Bailey AM, Dailey G, Evans-Holm M, Siebel CW, Misra S, et al. Identification of putative noncoding polyadenylated transcripts in Drosophila melanogaster. Proc Natl Acad Sci U S A. 2005;102(15):5495-500.