Supplementary File 5

Exon-Intron Boundary Detection Made Easy by Physicochemical Properties of DNA

Dinesh Sharma[a], Danish Aslam[a], Kopal Sharma[a], Aditya Mittal[a], B Jayaram*[a,b]

a: Supercomputing Facility for Bioinformatics & Computational Biology (SCFBio), Kusuma School of Biological Sciences, Indian Institute of Technology Delhi (IIT Delhi), Hauz Khas, New Delhi, 110016, India.
b: Department of Chemistry, IIT Delhi, Hauz Khas, New Delhi, 110016, India.
*: bjayaram@chemistry.iitd.ac.in

Table of Contents:

General Information:

ChemEXIN is an open-source, deep learning integrated, physicochemical parameter-based exon-intron boundary prediction method. It is based on a three-dimensional convolutional neural network (3D-CNN) architecture. Three organism-specific (*Homo sapiens, Mus musculus,* and *Caenorhabditis elegans*) models have been built and implemented in the final prediction pipeline. The universality of ChemEXIN lies in its ability to predict exon-intron boundaries across varying lengths of known genes (180 to ~2,500,000 nucleotides) in the three organisms under study.

Prerequisites:

Conda, to create an environment for ChemEXIN across Windows, macOS, and Linux operating systems. This environment will include all the required dependencies.
Conda is available at https://conda.io/projects/conda/en/latest/user-guide/install/index.html.

Setup (one-time setup):

Open a command prompt and follow the steps listed below:

Clone the ChemEXIN project repository:

```
$ git clone https://github.com/rnsharma478/ChemEXIN.git
```

Change the working directory to the successfully cloned project directory:

```
$ cd ChemEXIN
```

Install the virtual environment:

```
$ conda env create -f ChemEXIN.yml
```

Working Example (on Ubuntu 22.04):

The following example includes prediction of exon-intron boundary junctions using ChemEXIN in Neuraminidase 1 (NEU1) gene in *Homo sapiens.* The sequence is available as "example.fasta" in the sequence directory within the ChemEXIN directory.

Note: The commands will vary based upon the user's operating system, e.g, "dir" is used instead of "ls" on windows.

STEP1:

Considering that the user has successfully followed the Setup steps, and the ChemEXIN directory is now available at the desired path; set the working directory to ChemEXIN (if not already there).

```
$ cd ChemEXIN
```

STEP2:

Activate the **ChemEXIN** environment and move to the **ChemEXIN** directory.

```
(base) [redacted]:~/Desktop/ChemEXIN$ conda activate ChemEXIN
```

STEP3:

(A) Check the contents of the directory.

```
(ChemEXIN) [redacted]:~/Desktop/ChemEXIN$ ls
build  ChemEXIN_F.egg-info  main.py  models  param_files  README.md  results  sequence  setup.py  src
```

(B) Run the prediction tool.

```
(ChemEXIN) [redacted]:~/Desktop/ChemEXIN$ python main.py
2024-04-22 11:43:46.271735: I tensorflow/core/util/port.cc:110] oneDNN custom operations are on. You may see slightly differ
 orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2024-04-22 11:43:46.314542: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will n
2024-04-22 11:43:46.672464: I tensorflow/tsl/cuda/cudart_stub.cc:28] Could not find cuda drivers on your machine, GPU will n
2024-04-22 11:43:46.674027: I tensorflow/core/platform/cpu_feature_guard.cc:182] This TensorFlow binary is optimized to use
To enable the following instructions: AVX2 AVX512F AVX512_VNNI FMA, in other operations, rebuild TensorFlow with the appropr
2024-04-22 11:43:48.444666: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
```

STEP4:

Provide the necessary inputs* when prompted

```
                    ChemEXIN: A Physicochemical Parameter-Based Exon-Intron Boundary Prediction Method developed by SCFBio, IIT Delhi.
Input your file name present in the sequence directory and hit ENTER:
Waiting for user input: example.fasta

Select the organism and hit ENTER:
> h or H for H. sapiens
> m or M for M. musculus
> c or C for C. elegans
Waiting for user input: h

Carrying forward the analysis with the selected Organism.

Sequence pre-processing started

The input sequence file contains multiple lines.
Do you want to retrieve the single line sequence file?(Y or y/N or n)
Waiting for user input: y

File saved to results/example_single
```

*The **text in white** is the user input, in this case; **"example.fasta"** is the input file name containing the sequence in fasta format; **"h"** is the character input for the *Homo sapiens* model; **"y"** prompts the tool to save a single line file with the sequence named **"example_single"** with no header.

Note: Single line file generation option is available only if the file contains a sequence spanning multiple lines. This option is an add-on utility that can be used for generating single lined sequences from multi-lined fasta files.

STEP6:

Users can track real-time progress following the processing statements displayed in the command prompt.

The displayed statements are as follows,

STEP1/8.

(A) Checks the sequence length based on the length filter incorporated in the prediction pipeline.
> For the input sequence “example.fasta”, the length = 10,881 nucleotides.

(B) Sequence validity filter checks if all the characters in the sequence are valid DNA bases (A/T/G/C) or not.
> The input sequence “example.fasta” has all the valid characters.

```
STEP 1/8: Checking for the sequence length and sequence characters.
        Check 1: PASSED ---> The input sequence length >= 180 and <= 2500000 (input length = 10881).
        Check 2: PASSED ---> All the input sequence characters are valid.
STEP 1/8: SUCCESSFULLY COMPLETED :)
```

STEP2/8.

(A) Converts the sequence into trinucleotide parameter profiles followed by its normalization.
> Two "Checks" appear, one when the conversion ends and the other when normalization ends.

(B) "SUCCESSFULLY COMPLETED :)″ is prompted in case there are no errors.

STEP3/8.

(A) Combines the individual trinucleotide numerical profiles into the major feature categories, i.e., BP-Axis, Backbone organization, Intra-BP organization, and the three energetics (Hydrogen bond energy, Solvation energy, and Stacking energy).

(B) "SUCCESSFULLY COMPLETED :)″ is prompted in case there are no errors.

```
STEP 2/8: Converting sequence into normalised tri-nucleotide parameter profiles.
        Check 1: PASSED ---> Computed moving averages.
        Check 2: PASSED ---> Normalised moving averages.
STEP 2/8: SUCCESSFULLY COMPLETED :)

STEP 3/8: Combining individual tri-nucleotoide numerical profiles into the major feature categories.
STEP 3/8: SUCCESSFULLY COMPLETED :)
```

STEP4/8.

(A) Converts the sequence into tetranucleotide parameter profiles followed by the normalization.
> Two "Checks" appear one when the conversion ends and the other when normalization ends.

```
STEP 4/8: Converting sequence into normalised tetra-nucleotide parameter profiles.
        Check 1: PASSED ---> Computed moving averages.
        Check 2: PASSED ---> Normalised moving averages.
STEP 4/8: SUCCESSFULLY COMPLETED :)
```

(B) "SUCCESSFULLY COMPLETED :)″ is prompted in case there are no errors.

STEP5/8.

(A) Combines the individual tetranucleotide numerical profiles into the major feature category i.e., Inter-BP organization.

(B) Concatenates the combined tri- and tetra-nucleotide profiles together.

```
STEP 5/8: Combining individual tetra-nucletoide numerical profiles into major feature category and concatenating tri- and tetra-nucleotide profiles together.
STEP 5/8: SUCCESSFULLY COMPLETED :)
```

(C) "SUCCESSFULLY COMPLETED :)″ is prompted in case there are no errors.

STEP6/8.

(A) Creates the Final prediction data frame.

(B) “SUCCESSFULLY COMPLETED :)″ is prompted in case there are no errors.

STEP7/8.

(A) Runs the user selected 3D-CNN model (in this case, we chose, “h”, so the *H. sapiens* prediction model is used).

(B) “SUCCESSFULLY COMPLETED :)″ is prompted in case there are no errors.

(C) The probability threshold value prompt appears, the default is set at 0.75, if pressed enter without choosing a/A, b/B, or c/C, then predictions are automatically made on the 0.75 threshold. In this case, we proceeded with the default value by hitting ENTER.

```
Select the threshold value and hit ENTER else hit ENTER to proceed with default (0.75):
> a or A for PROB: 0.70
> b or B for PROB: 0.80
> c or C for PROB: 0.85
Waiting for user input:
The entered option doesn't correspond to a valid threshold.
Carrying the analysis with the default value (0.75).
```

STEP8/8.

(A) Captures, refines and generates the position output.

> Three “Checks” appear on all positions; capture, refining, and results generation.

(B) “SUCCESSFULLY COMPLETED :)″ is prompted in case there are no errors.

```
STEP 8/8: Exon-intron boundary capture, position refining and Output generation.
        Check 1: PASSED ---> Positions captured successfully.
        Check 2: PASSED ---> Positions refined successfully.
        Check 3: PASSED ---> Output saved successfully to results/example_results.csv

STEP 8/8: SUCCESSFULLY COMPLETED :)
Total Execution time: 9.47407579421997 secs
```

(C) Total Execution time is displayed for the entire run.

(D) The prediction results are made available in the results/ directory of the ChemEXIN.

Output description:

In the output file, the user-specific metadata is available in the multi-line header; each line starts with a "#" character, followed by the predicted boundary windows information.

Metadata includes,

- the user input parameters;
  Employed Model---> *H. sapiens*
  Input File Name---> example
- the field description;
  S.No.---> Predicted boundary serial number.
  Primary_ start---> Target boundary window start site.
  Primary_end---> Target boundary window end site.
  Secondary_start---> Extended boundary window start site.

#ChemEXIN OUTPUT FILE

#ChemEXIN 1.0 Output generated on: 2024-04-22 11:43:59.777340

#Input Parameters:
#Employed Model---> H. sapiens
#Input File Name---> example

#Field description:
#S.No.---> Predicted boundaty serial number.
#Primary_start---> Target boundary window start site.
#Primary_end---> Target boundary window end site.
#Secondary_start---> Extended boundary window start site.
#Secondary_end---> Extended boundary window end site.

#Predicted Exon-intron boundaries at 0.75 reliability threshold value.
#Threshold value corresponds to the prediction probability of the boundary windows.

S.No.	Primary_start	Primary_end	Secondary_start	Secondary_end
1	50	99	20	179
2	400	449	370	479
3	650	699	620	729
4	850	899	820	929
5	1000	1049	970	1129
6	1350	1399	1270	1429
7	1600	1649	1570	1679

  Secondary_end---> Extended boundary window end site.
- The **reliability threshold value** description.

Description:

The probabilities of exon-start (Class 1) and exon-end (Class 2) have been merged due to similar area under the receiver operating characteristic (AUROC) curves, as illustrated in Figure 4 of the main text. This similarity reflects the nearly identical biophysical profiles observed for acceptor and donor sites. Therefore, the predictions generated by our model undergo uniform processing procedures and incorporate multiple filters in the backend to enhance accuracy and reliability.

The main filters include,

- *the sequence length filter;*

Sequences must meet specific length criteria: they should be 180 nucleotides long or more but not exceed 2,500,000 nucleotides. This restriction is imposed because predictions are made on independent, non-overlapping windows of 50 nucleotides.

- *the terminal exon filter;*

Specific precautions have been taken since exon-end sites could erroneously appear at the beginning of a sequence, or exon-start sites might be predicted towards the sequence's end due to our combined probabilities approach. Predictions within the sequence's initial window (0-50) are disregarded, as are any nucleotides beyond the last possible 50-nucleotide window from the input sequence.

- *the probability threshold filter;*

(A) The model searches for positions with the user-specified seeding threshold value (0.75 by default or 0.70/0.80/0.85). After creating a 50-length window using the predicted position, it designates this window as the target window and demarcates the start-/end- sites as the primary start and primary end. The model reports an extended window with the secondary start-/end- sites to refine the search scope and enhance the prediction reliability. This window spans –30 and +30, respectively, around the primary start and end sites.

(B) If a subsequent position after a predicted seed position (predicted with 0.75 (default) or 0.70/0.80/0.85 threshold value) has a probability of boundary occurrence >=0.70, then the primary start-/end-sites are given for a window with respect to the position with a higher probability score, while the secondary start-/end-sites are given such that –30 upstream is taken for the first window's start site, and +30 downstream is defined around the second window's end site.

(C) If a longer stretch comprising of three or more than three positions (each having probability >= 0.70) occurs (after the first seed threshold position), we can have two cases:

> (I) If the stretch has an even number of positions (including the first seed position), then for this stretch, n/2 and n+½ positions are considered (n=total number of positions in the stretch). Final processing is then applied as per step B (the primary start-/end-sites are reported for a window with respect to a position with higher probability, and the extended secondary window is created by moving -30 upstream from the first window's start site and +30 nucleotides downstream from the second window's end site).
>
> (II) If the stretch has an odd number of positions, the median position is selected from the total number of positions (including the seed position). This gives a single position that undergoes further processing (to generate a window), the same as the single position processing explained in step A.

Note: While the reliability threshold value is flexible, ranging from 0.70 to 0.85, it is advisable to consider factors such as over-representation and loss of information.