Contents lists available at ScienceDirect

Computational and Structural Biotechnology Journal

journal homepage: www.elsevier.com/locate/csbj

ProFeatX: A parallelized protein feature extraction suite for machine learning

David Guevara-Barrientos [a,b], Rakesh Kaundal [a,b,c,*]

[a] Department of Computer Science, College of Science, Utah State University, Logan, UT, USA
[b] Bioinformatics Facility, Center for Integrated BioSystems, Utah State University, Logan, UT, USA
[c] Department of Plants, Soils, and Climate, College of Agriculture and Applied Sciences, Utah State University, Logan, UT, USA

ARTICLE INFO

Article history:
Received 7 July 2022
Received in revised form 26 December 2022
Accepted 27 December 2022
Available online 29 December 2022

Keywords:
Feature extraction
Amino-acid sequence
Descriptors
Protein-protein interactions
Machine learning

ABSTRACT

Summary: Machine learning algorithms have been successfully applied in proteomics, genomics and transcriptomics. and have helped the biological community to answer complex questions. However, most machine learning methods require lots of data, with every data point having the same vector size. The biological sequence data, such as proteins, are amino acid sequences of variable length, which makes it essential to extract a definite number of features from all the proteins for them to be used as input into machine learning models. There are numerous methods to achieve this, but only several tools let researchers encode their proteins using multiple schemes without having to use different programs or, in many cases, code these algorithms themselves, or even come up with new algorithms. In this work, we created ProFeatX, a tool that contains 50 encodings to extract protein features in an efficient and fast way supporting desktop as well as high-performance computing environment. It can also encode concatenated features for protein-protein interactions. The tool has an easy-to-use web interface, allowing non-experts to use feature extraction techniques, as well as a stand-alone version for advanced users. ProFeatX is implemented in C++ and available on GitHub at https://github.com/usubioinfo/profeatx. The web server is available at http://bioinfo.usu.edu/profeatx/.

1. Introduction

Proteins are chains of amino acids with variable length, ranging from tens to thousands long. This entails a problem for machine learning (ML) algorithms, most of which require that all the input data have the same vector size. There have been ML algorithms that surpass this barrier, most notably those dealing with natural language processing (NLP) [20], that treat each amino acid like a part of a language, but it has its limitations, since words in human speech have a grammatical structure and a relation to each other in relatively short distances. This means that the number of words that separates two words is small, whereas in proteins a group of amino acids, due to the way it folds, can exchange forces with amino acids that are far apart, altering the structure and, hence, other properties like its function or expression.

Over the last decades, there has been an effort to encode the proteins amino acid string into arrays of numbers with consistent length. These methods have allowed computers to solve multiple problems involving proteins, like predicting their structure [1] and functions [3], predicting side chain conformations [18], or helping with enzyme engineering [17]. Multiple tools have been developed to achieve this, such as iFeature [7], iFeature Omega [9], iLearnPlus [8], protr [23], Rcpi [6], Pfeature [21] and MathFeature [4]. Moreover, these tools do not include protein-protein interaction (PPI) encoding, which is a utility that would be very useful for machine learning studies that work with PPIs, since the authors usually have to do it on their own [22,15,16]. This represents an additional data preprocessing task.

In this work, we present ProFeatX, a tool for encoding amino acid sequences into 50 different representations. It has a stand-alone version and a web server version freely accesible at https://bioinfo.usu.edu/profeatx. This web version can take DNA sequences as input as it can perform extractions from open reading frames (ORFs) found in DNA/RNA sequences, and a major novelty in the capability to encode features for protein-protein interactions from two protein sets by concatenation, crossing them in an all-vs-all fashion.

* Corresponding author at: Department of Computer Science, College of Science, Utah State University, Logan, UT, USA.
E-mail address: rkaundal@usu.edu (R. Kaundal).

https://doi.org/10.1016/j.csbj.2022.12.044

Table 1
List of available encodings. Those listed as “Same length” mean that all input sequences must be the same length for the algorithm to run. Those that need additional files besides the FASTA input file with the amino acid sequences are not available in the web server version. All additional input files are per sequence.

Name	Short name	Vector size (default parameters)	Same length	Additional files
Amino Acid Composition	AAC	20		
Dipeptide Composition	DPC	400		
Tripeptide Composition	TPC	8000		
Composition of k-Spaced Amino Acid Pairs	CKSAAP	2400		
Dipeptide Deviation from Expected Mean	DDE	400		
Enhanced Amino Acid Composition	EAAC	(sequence length – 6) * 20	Yes	
Amino Acid Pair Antigenicity Scale	AAPAS	400		
Composition Moment Vector	CMV	20		
Grouped Amino Acid Composition	GAAC	5		
Enhanced Grouped Amino Acid Composition	EGAAC	(sequence length – 4) * 5	Yes	
Composition of k-Spaced Amino Acid Group Pairs	CKSAAGP	150		
Grouped Dipeptide Composition	GDPC	25		
Grouped Tripeptide Composition	GTPC	125		
Encoding Based on Grouped Weight	EBGW	60		
Quasi-Sequence-Order	QSO	100		
Sequence-Order-Coupling Number	SOCN	60		
Geary Autocorrelation	Geary	240		
Moran Autocorrelation	Moran	240		
Normalized Moreau-Broto Autocorrelation	NMB	240		
Composition / Transition / Distribution - Composition	CTDC	42		
Composition / Transition / Distribution - Transition	CTDT	42		
Composition / Transition / Distribution - Distribution	CTDD	210		
Conjoint Triad	CT	343		
k-Spaced Conjoint Triad	KSCT	646		
Pseudo-Amino Acid Composition	PAAC	50		
Ampiphilic Pseudo-Amino Acid Composition	APAAC	80		
Binary	Binary	sequence length * 20	Yes	
Taylor's Venn Diagram	TVD	sequence length	Yes	
Pseudo k-Tuple Reduced Amino Acid Composition	PseKRAAC	4		
Amino Acid Index	AAI	sequence length * 531	Yes	
BLOSUM62	BLOSUM62	sequence length * 20	Yes	
Z-Scale	ZS	sequence length * 5	Yes	
Secondary Structure Elements Binary	SSEB	sequence length * 3	Yes	.ss2 file generated by PSIPRED OR .spXout file generated by SPINE-X
Secondary Structure Elements Content	SSEC	3		.ss2 file generated by PSIPRED OR .spXout file generated by SPINE-X
Secondary Structure Probabilities Bigram	SSPB	9		.ss2 file generated by PSIPRED OR .spXout file generated by SPINE-X
Seconday Structure Probabilities Auto-Covariance	SSPAC	30		.ss2 file generated by PSIPRED OR .spXout file generated by SPINE-X
Disorder	Disorder	sequence length	Yes	.dis file generated by VSL2
Disorder Content	DisorderC	2		.dis file generated by VSL2
Disorder Binary	DisorderB	sequence length * 2	Yes	.dis file generated by VSL2
Torsion Angles	TA	sequence length * 2	Yes	.spXout file generated by SPINE-X
Torsion Angles Composition	TAC	4		.spXout file generated by SPINE-X
Torsion Angles Bigram	TAB	10		.spXout file generated by SPINE-X
Torsion Angles Autocovariance	TAAC	4		.spXout file generated by SPINE-X
Accessible Surface Area	ASA	sequence length	Yes	.spXout file generated by SPINE-X
k-Nearest Neighbor for Peptides	KNNpeptide	number of labels in training file * 30	Yes	Training FASTA file and labels file with sequence names and classes
k-Nearest Neighbor for Proteins	KNNproteins	number of labels in training file * 30		Training FASTA file and labels file with sequence names and classes
Position-Specific Scoring Matrix	PSSM	sequence length * 20	Yes	.pssm file generated by blastpgp or psiblast
PSSM Amino Acid Composition	PSSMAAC	20		.pssm file generated by blastpgp or psiblast
Bigram PSSM	BiPSSM	400		.pssm file generated by blastpgp or psiblast
PSSM Autocovariance	PSSMAC	600		.pssm file generated by blastpgp or psiblast
Pseudo-PSSM	PPSSM	50		.pssm file generated by blastpgp or psiblast

Therefore, this tool can be used for extracting various encodings of protein sequences for single proteins as well as for protein-protein interaction pairs and use them as input to ML algorithms. ProfeatX can handle large-scale submissions as it has been implemented on a high-performance computing environment and supports parallelization.

2. Materials and methods

ProFeatX was implemented in C++, taking advantage of its performance, plus parallelization with OpenMP, substantially improving the speed at which it encodes the proteins depending on the number of threads that the user wants to use. This is important for large data sets that, without the parallelization, could take several days.

We implemented 50 different methods for encoding the proteins, represented in Table 1. Explanations on how to calculate them can be found in Supplementary Material 1.

In addition to the stand-alone program, for cases where the data sets are not large, we created a web server built with Python's framework, Django.

3. Results

3.1. Stand-alone version

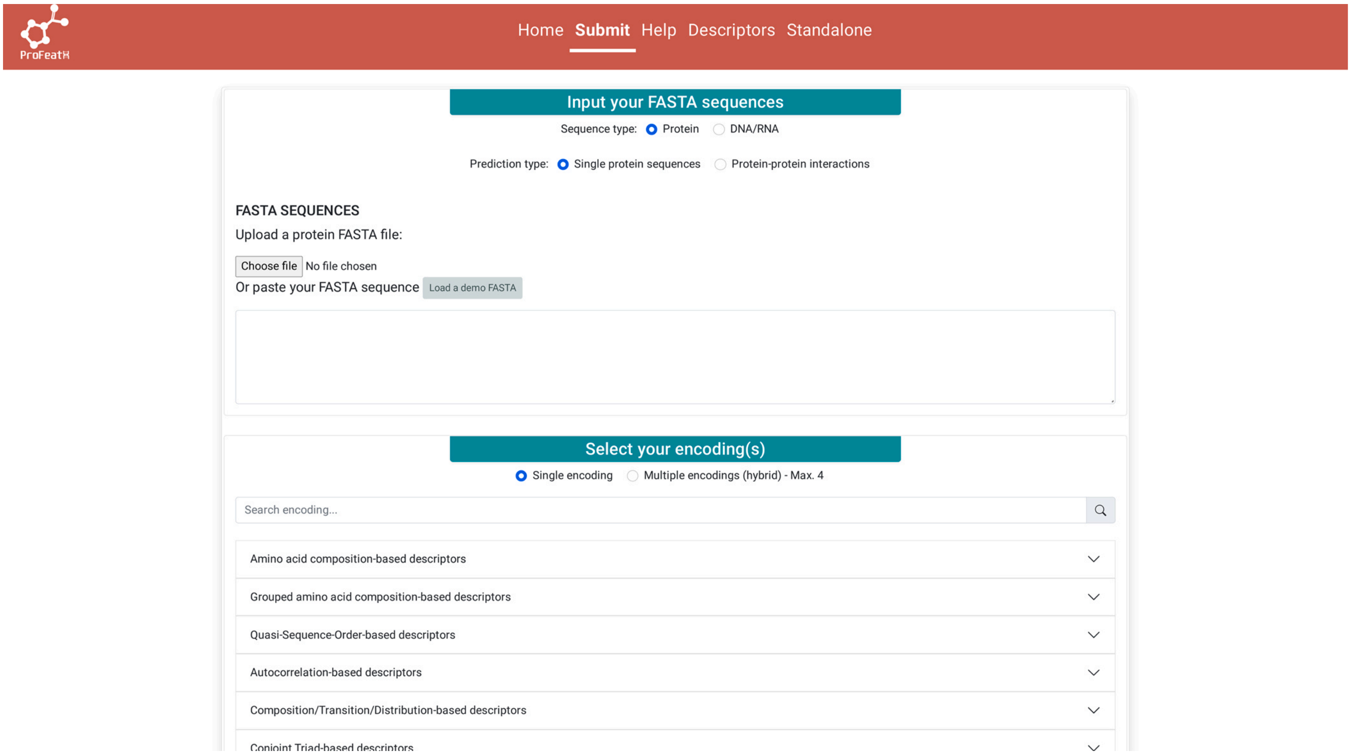
We developed the stand-alone version of ProFeatX, which reads amino acid sequences in FASTA format, removes characters that do not represent any of the 20 natural amino acids, and transforms them into vectors with same size using parallel computing into 50 different descriptors, some of them with customizable parameters (Supplementary material 1), where 19 of them require additional files (Table 1): SSEB, SSEC, SSPB and SSPAC require.ss2 files generated by PSIPRED [14],.spXout files generated by SPINE-X [10], or.spd33 files generated by SPIDER3 [12]; Disorder, DisorderB and DisorderC require.csv files generated by flDPnn [13]; TA, TAC, TAB, TAAC and ASA require.spXout files generated by SPINE-X or.spd33 files generated by SPIDER3; PSSM, PSSMAAC, BiPSSM, PSSMAC and PPSSM require.pssm files generated by the blastpgp command if using legacy BLAST [2], or the psiblast command if using BLAST+ [5]; KNNpeptide and KNNprotein require, in addition to the input FASTA file, a training FASTA file and a label file, where it lists the classification for each training sequence.

3.2. Web version

The web server we developed lets the user encode sequences between one to four descriptors at once (e.g., to develop hybrid combinations), and they are able to encode protein-protein interactions between two sets of amino acid sequences in an all-to-all fashion, concatenating the features of each protein into a single vector. This web tool allows the user to upload a FASTA file of up to 50MB for single proteins encoding, and two FASTA files of up to 10MB for PPIs and for the 19 descriptors that require extra files (Fig. 1). For 12 of these 19 descriptors (excluding TA, TAC, TAB, TAC, ASA, KNNprotein and KNNpeptide), it can also generate the required files for up to 20 sequences if the user does not upload their own files. The input FASTA files can be amino acid sequences, or DNA/RNA sequences, which, using TransDecoder [11], identifies protein coding regions from these sequences and encodes them. The results can be downloaded in CSV, Excel, and pickle (.pkl) formats.

3.3. Benchmarking and comparison

Every descriptor has been benchmarked against each other with 1 and 8 threads. These tests focused on the time that it took to process a FASTA file with 1000 sequences, each one 640 amino acids long (Supplementary Material 2). When using both 1 core and 8 cores, the descriptor that takes the longest time is KNNprotein due to each input sequence having to calculate global alignments with the Needleman-Wunsch algorithm [19], against every sequence in the training file which, for testing purposes, is the same as the input file with 1000 sequences. AAI was the second descriptor that took the longest time because of its huge vector size, creating a



Input your FASTA sequences

Sequence type: ● Protein ○ DNA/RNA

Prediction type: ● Single protein sequences ○ Protein-protein interactions

FASTA SEQUENCES

Upload a protein FASTA file:

Choose file No file chosen

Or paste your FASTA sequence Load a demo FASTA

Select your encoding(s)

● Single encoding ○ Multiple encodings (hybrid) - Max. 4

Search encoding...

Amino acid composition-based descriptors

Grouped amino acid composition-based descriptors

Quasi-Sequence-Order-based descriptors

Autocorrelation-based descriptors

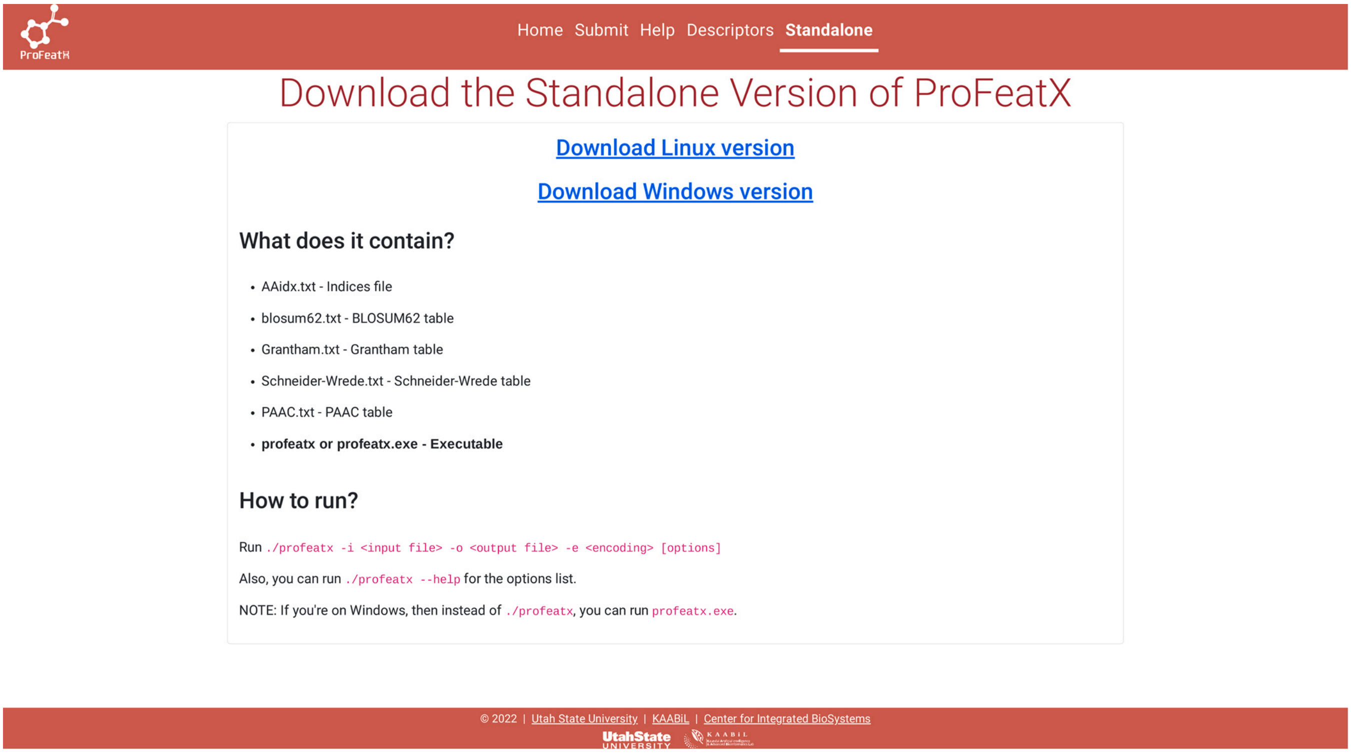
Composition/Transition/Distribution-based descriptors

Conjoint Triad-based descriptors

Fig. 1. A snapshot of the submission page of ProFeatX depicting the 50 encodings available (under 10 broad categories) for both the analysis of single proteins as well as PPI pairs. The encodings can also be done for a single feature or a combination (hybrid) of up to 4 features. Encodings are divided into 10 major categories as shown in the figure: Amino acid composition-based, Grouped amino acid composition, Quasi-sequence-order based, Autocorrelation-based, Composition/Transition/Distribution, Conjoint Triad-based, Pseudo-amino acid composition, Binary-based, Pseudo *k*-Tuple reduced amino acid composition-based, and Other encodings.

Table 2
Comparison of features and runtimes between different encoding tools.

Tool	Language	Total descriptors	Parallelization	Standalone	Web server	PPI support	Time (ms) - Stand-alone version
ProFeatX	C++	50	Yes	Yes	Yes	Yes, web version	764
iFeature	Python	38	No	Yes	Yes	No	4560
iFeatureOmega	Python	39	No	Yes	Yes	No	4842
iLearnPlus	Python	37	No	Yes	Yes	No	5102
MathFeature	Python	14	No	Yes	No	Script for joining files	159,653
Pfeature	Python	37	No	Yes	Yes	No	26,799
protr	R	25	No	Yes	Yes	No	4963
Rcpi	R	22	No	Yes	No	Function for joining tables	4943



Download the Standalone Version of ProFeatX

Download Linux version

Download Windows version

What does it contain?

- AAidx.txt - Indices file
- blosum62.txt - BLOSUM62 table
- Grantham.txt - Grantham table
- Schneider-Wrede.txt - Schneider-Wrede table
- PAAC.txt - PAAC table
- **profeatx or profeatx.exe - Executable**

How to run?

Run `./profeatx -i <input file> -o <output file> -e <encoding> [options]`

Also, you can run `./profeatx --help` for the options list.

NOTE: If you're on Windows, then instead of `./profeatx`, you can run `profeatx.exe`.

© 2022 | Utah State University | KAABiL | Center for Integrated BioSystems

Fig. 2. ProFeatX's download page of the stand-alone version. Instructions on how to run the tool locally or on the HPC are provided. Alternatively, the users can visit the 'Help' page of ProFeatX webserver for more details.

bottleneck writing the data into the output file. The only descriptor that did not improve when increasing the number of cores was ZS, presumably as a consequence of the overhead that represents sending a copy of the small Z-scale matrix (20 ×5) to every thread (Supplementary Material 3). The benchmarks did not consider the time required to generate the files required for the 19 descriptors that are based on PSSM, secondary structure and disorder.

ProFeatX was compared against six other protein encoding tools: iFeature, iFeatureOmega, iLearnPlus, MathFeature, Pfeature, protr, Rcpi (Table 2). ProFeatX is the only tool developed in C++, while iFeature, iFeatureOmega, iLearnPlus, MathFeature and Pfeature were developed in Python, and protr and Rcpi in R. The total descriptors were counted, where ProFeatX took the lead with 50 total methods for the standalone version, including multiple encodings that are not found in other suites. These are APAAS, CMV, EBGW, TVD, SSPB, SSPAC, TAC, TAB, TAAC, PSSMAAC, BiPSSM and PPSSM (Supplementary Material 4). Also, ProFeatX is the only tool that supports parallel computing. Every tool has a standalone version, but Rcpi and MathFeature do not have a web version. Moreover, ProFeatX supports encodings capability for PPIs, a unique feature useful to develop fast ML algorithms for the prediction of protein-protein interactions, including modeling for host-pathogen interactions. The tests were perfomed encoding a FASTA file with 22,487 sequences with lengths ranging from 32 to 2597 amino acids (Supplementary Material 5), using the AAC encoding method running on a single thread. For protr and Rcpi, we did a previous data cleaning, removing from each sequence all characters that did not belong to the natural 20 amino acids due to tool limitations. The results show that our tool was nearly 6 times faster than iFeature, which was the second in speed (Table 2). All benchmarks, including the descriptor vs descriptor ones, were executed on an AMD EPYC 7601 using 16 GB of RAM.A,.

3.4. Basic usage

In order to use the standalone version (Fig. 2), an example of the basic usage is as follows:

```
./profeatx -i input.fasta -o output.tsv -t 8 -e AAC.
```

Here, input.fasta would be the input file, output.tsv would be the output file, 8 would be the number of threads, and AAC would be the descriptor. Depending on the descriptor, more arguments can be used. For the full list, please refer to the Supplementary Material 1, which can also be found at http://bioinfo.usu.edu/profeatx/descriptors/, or execute the command./profeatx -help.

In order to use the web version of ProFeatX, the detailed step-by-step instructions are also provided on the 'Help' page of the web server at http://bioinfo.usu.edu/profeatx/help/, providing examples of input data and multiple output formats available.

4. Conclusion

In this work, we developed a fast and easy to use tool for encoding proteins, which can be extremely useful for researchers that are creating machine learning algorithms dealing with proteins. It is especially convenient if they have multiple thousands of sequences to be used as input, and a computer that supports multiple threads. To our knowledge, there are no other web tools that allow users to encode multiple descriptors on protein-protein interactions and with such high number of available methods and parameter customizability.

We expect ProFeatX to become a widely known tool and help a lot of scientists in their machine learning research. In future versions we will increment the number of available encodings. Also, our tool will support file generation for the stand-alone version, so it executes the needed programs if installed such as PSIPRED and SPINE-X if these already exist in the local machine, and PPI support, not only by concatenation, but other methods such as Euclidean distance or multiplication. ProFeatX has a user-friendly interface and is expected to be widely used as a powerful tool to develop more diverse and strong ML classifiers, and help advance research in bioinformatics, computational biology, and systems biology.

Funding

The authors acknowledge the support to this study from the faculty start-up funds to RK from the Center for Integrated BioSystems (CIB) / Department of Plants, Soils, and Climate, USU. This research was also partially supported by Utah Agricultural Experiment Station (UAES) and approved as journal paper number 9637. The funding body did not play any role in the design of this study; the collection, analysis, or interpretation of data; or in the writing of this manuscript.

CRediT authorship contribution statement

DG: Methodology, Data curation, Software writing, webserver development, draft manuscript., RK: Conceptualization, Supervision, Validation, Writing - review & editing.

Data availability

All the encoded features inside ProFeatX have been implemented on a webserver which is freely available at http://bioinfo.usu.edu/profeatx/. The stand-alone version of the package can be downloaded from http://bioinfo.usu.edu/profeatx/download/. ProFeatX is implemented in C++ and all the code is available on GitHub at https://github.com/usubioinfo/profeatx.

Declaration of Competing Interest

The authors declare that there is no conflict of interest.

Acknowledgements

The authors acknowledge the help and support from the members of the KAABiL lab for testing the ProfeatX tool and providing feedback. The authors sincerely thank the anonymous referees for all the suggestions and help in improving the research article.

Appendix A. Supporting information

Supplementary data associated with this article can be found in the online version at doi:10.1016/j.csbj.2022.12.044.

References

[1] AlQuraishi M. Machine learning in protein structure prediction. Curr. Opin. Chem. Biol. 2021;65:1–8. https://doi.org/10.1016/j.cbpa.2021.04.005

[2] Altschul S. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 1997;25(17):3389–402. https://doi.org/10.1093/nar/25.17.3389

[3] Bonetta R, Valentino G. Machine learning techniques for protein function prediction. Proteins: Struct., Funct. Bioinform. 2020;88(3):397–413. https://doi.org/10.1002/prot.25832

[4] Bonidia RP, Domingues DS, Sanches DS, de Carvalho ACPLF. MathFeature: feature extraction package for DNA, RNA and protein sequences based on mathematical descriptors. Brief. Bioinform. 2022;23(1). https://doi.org/10.1093/bib/bbab434

[5] Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL. BLAST+: architecture and applications. BMC Bioinform. 2009;10(1):421. https://doi.org/10.1186/1471-2105-10-421

[6] Cao D-S, Xiao N, Xu Q-S, Chen AF. Rcpi: R/Bioconductor package to generate various descriptors of proteins, compounds and their interactions. Bioinformatics 2015;31(2):279–81. https://doi.org/10.1093/bioinformatics/btu624

[7] Chen Z, Zhao P, Li F, Leier A, Marquez-Lago TT, Wang Y, Webb GI, Smith AI, Daly RJ, Chou K-C, Song J. iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. Bioinformatics 2018;34(14):2499–502. https://doi.org/10.1093/bioinformatics/bty140

[8] Chen Z, Zhao P, Li C, Li F, Xiang D, Chen Y-Z, Akutsu T, Daly RJ, Webb GI, Zhao Q, Kurgan L, Song J. iLearnPlus: a comprehensive and automated machine-learning platform for nucleic acid and protein sequence analysis, prediction and visualization. Nucleic Acids Res. 2021;49(10). https://doi.org/10.1093/nar/gkab122

[9] Chen Z, Liu X, Zhao P, Li C, Wang Y, Li F, Akutsu T, Bain C, Gasser RB, Li J, Yang Z, Gao X, Kurgan L, Song J. iFeatureOmega: an integrative platform for engineering, visualization and analysis of features from molecular sequences, structural and ligand data sets. Nucleic Acids Res. 2022;50(W1):W434–47. https://doi.org/10.1093/nar/gkac351

[10] Faraggi E, Zhang T, Yang Y, Kurgan L, Zhou Y. SPINE X: Improving protein secondary structure prediction by multistep learning coupled with prediction of solvent accessible surface area and backbone torsion angles. J. Comput. Chem. 2012;33(3):259–67. https://doi.org/10.1002/jcc.21968

[11] Haas BJ, Papanicolaou A, Yassour M, Grabherr M, Blood PD, Bowden J, Couger MB, Eccles D, Li B, Lieber M, MacManes MD, Ott M, Orvis J, Pochet N, Strozzi F, Weeks N, Westerman R, William T, Dewey CN, Regev A. De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis. Nat. Protocols 2013;8(8):1494–512. https://doi.org/10.1038/nprot.2013.084

[12] Heffernan R, Yang Y, Paliwal K, Zhou Y. Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility. Bioinformatics 2017;33(18):2842–9. https://doi.org/10.1093/bioinformatics/btx218

[13] Hu G, Katuwawala A, Wang K, Wu Z, Ghadermarzi S, Gao J, Kurgan L. flDPnn: Accurate intrinsic disorder prediction with putative propensities of disorder functions. Nat. Commun. 2021;12(1):4438. https://doi.org/10.1038/s41467-021-24773-7

[14] Jones DT. Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. 1999;292(2):195–202. https://doi.org/10.1006/jmbi.1999.3091

[15] Kaundal R, Loaiza CD, Duhan N, Flann N. deepHPI: a comprehensive deep learning platform for accurate prediction and visualization of host–pathogen protein–protein interactions. Brief. Bioinform. 2022;23(3). https://doi.org/10.1093/bib/bbac125

[16] Mahapatra S, Gupta VR, Sahu SS, Panda G. Deep Neural Network and Extreme Gradient Boosting Based Hybrid Classifier for Improved Prediction of Protein-Protein Interaction. IEEE/ACM Trans Comput Biol Bioinform. 2022;19(1):155–65. https://doi.org/10.1109/TCBB.2021.3061300

[17] Mazurenko S, Prokop Z, Damborsky J. Machine learning in enzyme engineering. ACS Catal. 2020;10(2):1210–23. https://doi.org/10.1021/acscatal.9b04321

[18] Nagata K, Randall A, Baldi P. SIDEpro: a novel machine learning approach for the fast and accurate prediction of side-chain conformations. Proteins: Struct., Funct. Bioinform. 2012;80(1):142–53. https://doi.org/10.1002/prot.23170

[19] Needleman SB, Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. 1970;48(3):443–53. https://doi.org/10.1016/0022-2836(70)90057-4

[20] Ofer D, Brandes N, Linial M. The language of proteins: NLP, machine learning & protein sequences. Comput. Struct. Biotechnol. J. 2021;19:1750–8. https://doi.org/10.1016/j.csbj.2021.03.022

[21] Pande A, Patiyal S, Lathwal A, Arora C, Kaur D, Dhall A, Mishra G, Kaur H, Sharma N, Jain S, Usmani SS, Agrawal P, Kumar R, Kumar V, Raghava GPS. Computing wide range of protein/peptide features from their sequence and structure. BioRxiv 2019:599126https://doi.org/10.1101/599126

[22] Sun T, Zhou B, Lai L, Pei J. Sequence-based prediction of protein protein interaction using a deep-learning algorithm. BMC Bioinform. 2017;18(1):277. https://doi.org/10.1186/s12859-017-1700-2

[23] Xiao N, Cao D-S, Zhu M-F, Xu Q-S. protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences. Bioinform. (Oxford, England) 2015;31(11):1857–9. https://doi.org/10.1093/bioinformatics/btv042