BIOLOGY Methods & Protocols

Biology Methods and Protocols, 2022, 1–11

https://doi.org/10.1093/biomethods/bpac008
Advance Access Publication Date: 30 March 2022
Methods Article

METHODS ARTICLE

PSSMCOOL: a comprehensive R package for generating evolutionary-based descriptors of protein sequences from PSSM profiles

Alireza Mohammadi[1,†], Javad Zahiri [2,3,*,†], Saber Mohammadi [1], Mohsen Khodarahmi[4,5,6] and Seyed Shahriar Arab[7]

[1]Bioinformatics and Computational Omics Lab (BioCOOL), Department of Biophysics, Faculty of Biological Sciences, Tarbiat Modares University, Tehran 14115111, Iran, [2]Department of Neuroscience, University of California San Diego, San Diego, CA 92093, USA, [3]Department of Pediatrics, University of California San Diego, La Jolla, CA 92093, USA, [4]Department of Radiology, Shahid Madani Hospital, Karaj 44693, Iran, [5]Bahar Medical Imaging Center, Karaj 3144615931, Iran, [6]Dr. Khodarahmi Medical Imaging Center, Karaj 3144615931, Iran and [7]Department of Biophysics, Faculty of Biological Sciences, Tarbiat Modares University, Tehran 14115111, Iran

[†]These authors contributed equally to this work
[*]Correspondence address. Department of Neuroscience, University of California San Diego, San Diego, CA, USA. E-mail: jzahiri@health.ucsd.edu

Abstract

Position-specific scoring matrix (PSSM), also called profile, is broadly used for representing the evolutionary history of a given protein sequence. Several investigations reported that the PSSM-based feature descriptors can improve the prediction of various protein attributes such as interaction, function, subcellular localization, secondary structure, disorder regions, and accessible surface area. While plenty of algorithms have been suggested for extracting evolutionary features from PSSM in recent years, there is not any integrated standalone tool for providing these descriptors. Here, we introduce PSSMCOOL, a flexible comprehensive R package that generates 38 PSSM-based feature vectors. To our best knowledge, PSSMCOOL is the first PSSM-based feature extraction tool implemented in R. With the growing demand for exploiting machine-learning algorithms in computational biology, this package would be a practical tool for machine-learning predictions.

Keywords: R package; PSSM; machine learning; feature extraction

Introduction

Position-specific scoring matrix (PSSM) is defined as a matrix that involves information about the probability of amino acids or nucleotides occurrence in each position, which is derived from a multiple sequence alignment. This matrix is similar to the substitution matrix but it is more intricate due to including the alignment position information. In such a matrix, the rows represent the position of residues in an alignment and the columns specify the name of residues. This representation can be reversed so that the rows and columns would determine the name of residues and their corresponding positions in the alignment, respectively. The values of this matrix are the residues' binary logarithm derived from multiple alignment scores. Briefly speaking, the procedure of building PSSM can be summarized as three main steps (Fig. 1A).

Received: 3 March 2021; **Revised:** 21 January 2022; **Editorial Decision:** 27 January 2022

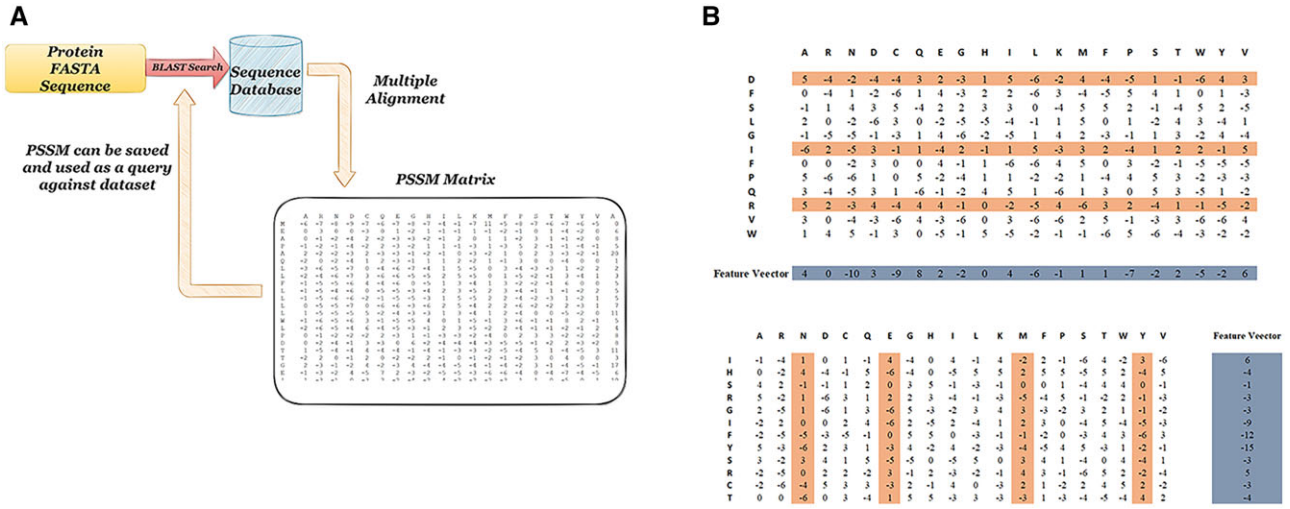


Figure 1: (**A**) The process used to build a PSSM. To build a PSSM, protein sequences are given to sequence databases such as NCBI as FASTA files for performing BLAST search. Having multiple alignments performed, a PSSM file can be obtained. The obtained PSSM can be used as a new query against the dataset. (**B**) Schematic presentation of row and column transformation. The feature vector specified as blue is obtained by summing the rows and columns highlighted in pink.

In these matrices, the positive numbers indicate that identical or similar sequences have been aligned and the negative numbers are indicators of a non-conserved alignment. This matrix, which can be considered a summary of the ensemble of corresponding sequences, is a quantified description for the conservation degree in each position of the alignment.

As far as the significance of PSSM is concerned, we investigated the studies that used the PSSM-based feature for predicting a protein attribute. By a thorough search on the literature in PubMed using PSSM and Prediction as keywords, we obtained 306 articles.

Moreover, the information conveyed through PSSMs is widely used in predicting various attributes of proteins ranging from the prediction of secondary and tertiary structures [1], protein–protein interactions [2], accessible surface area [3], flexibility [4], binding sites domains [5, 6], post-translational modification [7], protein localization [8], identifying the binding regions of protein–RNA [9], and protein–DNA [10] to the prediction of drug–target interaction [11]. Figure 2 shows the categorized papers based on their subjects that utilized PSSM-based features.

Feature extraction or feature encoding is a fundamental step in the construction of high-quality machine learning-based models. Specifically, this is a key step in various prediction problems in bioinformatics and computational biology [2, 12–14]. In the last two decades, a variety of new features have been proposed to train models for predicting several protein attributes. Such schemes are mostly based on sequence information or representation of diverse physicochemical properties. Although the features derived directly from protein sequences (such as amino acid compositions, conjoint triad, and k-mer composition) are regarded as essential factors for training efficient models, an increasing number of studies have shown that evolutionary information, which can be extracted from PSSMs, is much more informative than sequence-based information alone [2, 14, 15].

Several servers and software packages have been introduced to extract some specific descriptors from biological molecules, namely repRNA [16], repDNA [17], Pse-in-One [18, 19], Pse-Analysis [20], PseAAC [21], propy [22], PROFEAT [23], protr/ProtrWeb [24], and POSSUM [25]. All these tools, except for *POSSUM* and a recent version of *protr*, generate sequence-based and physiochemical features and commonly lack PSSM profile information. Moreover, more than 38 different PSSM-based feature types have been proposed in recent years. The POSSUM web server is the only tool devoted to the feature extraction from PSSMs. This tool provides 21 out of 38 PSSM-based feature types (unfortunately, the POSSUM tool was not accessible at the time of writing this manuscript). To the best of our knowledge, PSSMCOOL is the most comprehensive tool for extracting sequence-based evolutionary-related features from PSSMs.

Materials and methods

Various PSSM-derived features have been implemented as a comprehensive R package named PSSMCOOL. This R package includes 31 functions that extract 38 different PSSM-based features; that is, some of them are capable of generating more than one feature vector. These functions take a PSSM file for the protein of interest, as the input and output of the corresponding feature vector. In some functions, depending on the desired feature types, parameters are adjustable by users.

The implemented feature extraction algorithms are based on matrix transformations from the original PSSM profiles, which can be categorized into three types: Row transformations, column transformations (see Fig. 1B), and a mixture of row and column transformations (Table 1). For obtaining features derived from row transformation, we performed the following procedure: Two rows of PSSM were summed or subtracted or one or more rows were multiplied by a number. Similarly, by adding or reducing two or multiple columns, the features that were formed based on column transformation were obtained.

The 10 important features implemented within the PSSMCOOL package are summarized below. More details and formulas are provided on the online documentation.

PSSM-AC

This feature, which stands for auto-covariance transformation [33], calculates the j-th column average and subtracts this from the i-th and $i+g$-th rows of this column and finally, these numbers are multiplied (Fig. 3). The values of j vary between 1 and 20. By changing the i variable from 1 to $L-g$, the acquired

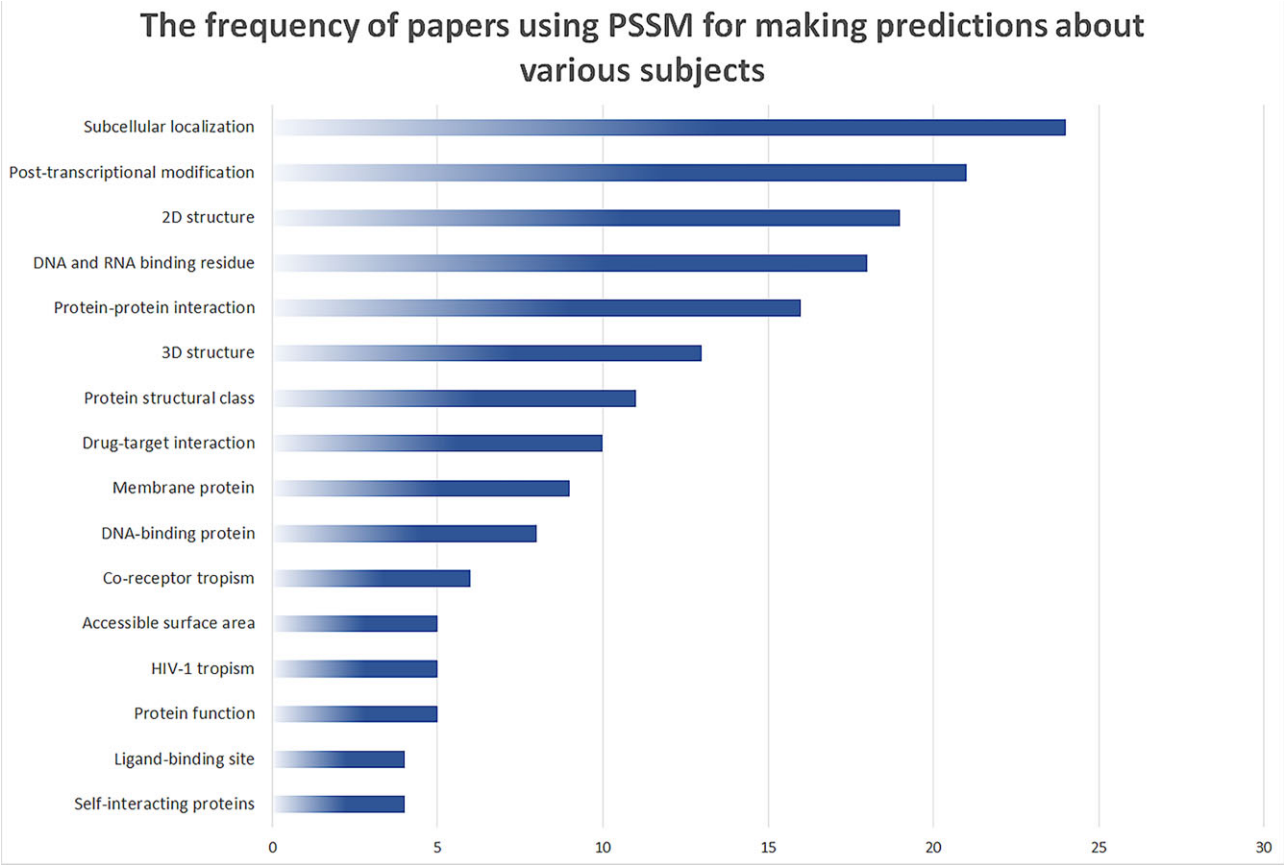


Figure 2: The frequency of categorized articles that employed PSSM-derived information in the years 1999–2021.

numbers are summed where L represents the length of the protein. The formula for generating this feature is provided in Equation (1).

$$\text{PSSM}-\text{AC}_{i,j}=\frac{1}{(L-g)}\sum_{i=1}^{L-g}\left(S_{i,j}-\frac{1}{L}\sum_{i=1}^{L}S_{i,j}\right)\left(S_{i+g,j}-\frac{1}{L}\sum_{i=1}^{L}S_{i,j}\right). \quad (1)$$

DPC–PSSM

This feature is related to dipeptide composition (DPC) [26] and originally was proposed for protein structural class prediction. For calculating this descriptor, the elements of two successive rows and two different columns are multiplied (see Fig. 4). This operation is performed on different rows and columns. Then, the computed values are summed and for every two successive rows, this sum is divided by $L-1$, where L is the protein length.

Trigram-PSSM

This feature is a feature vector with a length of 8000, which is extracted from PSSM [41]. If the elements of every three successive rows and three different columns of PSSM are multiplied and this operation is done for all three possible consecutive rows and eventually the acquired numbers are summed, we will have one of the elements of the final feature vector that corresponds to a specific combination of three amino acids out of 8000 possible combinations (Fig. 5). Equation (2) indicates how this feature can be generated.

$$T_{m,n,r}=\sum_{i=1}^{L-2}P_{i,m}P_{i+1,n}P_{i+2,r}. \quad (2)$$

PSe-PSSM

This feature originally was used to predict the membrane proteins and their types [47]. The PSe-PSSM feature vector is a vector with a length of 320 in which the 20 first numbers are the averages of 20 rows of PSSM [46]. The rest numbers of the final feature vector are computed as follows: For each column, the mean square of differences between the i-th and (i+lag)-th elements is computed for each column where lag can be any integer number between 1 and 15. Therefore, the length of the final feature vector will be $20\times15+20$. Figure 6 and Equation (3) show how this feature is generated.

$$\begin{gathered}p(k)=\frac{1}{(L-\text{lag})}\sum_{i=1}^{L-\text{lag}}(p_{i,j}-p_{i+\text{lag},j})^2\\ j=1,2,\ldots,\ 20,\text{lag}=1,2,\ldots,\ 15\\ k=20+j+20(\text{lag}-1)\end{gathered}. \quad (3)$$

K-separated-bigram-PSSM

This feature is almost identical to the DPC feature; in fact, the DPC feature is part of this feature (for $K=1$). As shown in Fig. 7, for every two different columns, it considers rows that have distance k [36].

Table 1: Implemented feature extraction algorithms and their application for predicting various problems in PSSMCOOL and a comparison between our package and POSSUM tool

	Descriptor name	Dimension	PSSMCOOOL	POSSUM	Reference	First usage
Row transformation	AAC-PSSM	20	✓	✓	[26]	Protein structural class
	AATP	420	✓	✓	[27]	Protein structural class
	AB-PSSM	400	✓	✓	[28]	Protein function
	CS_PSe_PSSM	700	✓	✗	[29]	Protein structural class
	D-FPSSM	20	✓	✓	[2]	Protein–protein interaction
	DISSULFID	[a]	✓	✗	[30]	Cysteine reactivity
	Kiderafactor	[a]	✓	✗	[31]	Ligand-binding site
	MEDP	420	✓	✓	[32]	Protein structural class
	PSSM-composition	400	✓	✓	[33]	Secreted effector proteins
	RPM-PSSM	400	✓	✓	[28]	Protein function
	S-FPSSM	400	✓	✓	[2]	Protein–protein interaction
	Smoothed-PSSM	[b]	✓	✓	[34]	RNA-binding sites
Column transformations	DMACA-PSSM	210	✓	✗	[35]	Protein types in Gram-negative bacteria
	DPC-PSSM	400	✓	✓	[36]	Protein fold recognition
	DWTPSSM	80	✓	✗	[37]	Protein crystallization prediction
	EEDP	400	✓	✓	[38]	Protein structural class
	k-separated-bigrams PSSM	400	✓	✓	[36]	Protein fold recognition
	LPC_PSSM	280	✓	✗	[39]	Protein structural class
	MBMGACPSSM	560	✓	✗	[32]	Protein structural class
	SCSH2	[b]	✓	✗	[14]	Protein–protein interaction
	SOMA_PSSM	160	✓	✗	[40]	Protein structural class
	TPC	400	✓	✓	[27]	Protein structural class
	tri-gram-PSSM	8000	✓	✓	[41]	Protein fold recognition
Combination of row and column transformations	AADP-PSSM	420	✓	✓	[26]	Protein structural class
	Average_Block	400	✓	✗	[42]	Protein classification
	Discrete cosin transform	400	✓	✗	[43]	Protein–protein interaction
	DP-PSSM	120	✓	✓	[44]	Subcellular localizations
	EDP	20	✓	✓	[38]	Protein structural class
	Gray_PSSM_PseAAC	100	✓	✗	[45]	Antifreeze proteins
	Pse-PSSM	[b]	✓	✓	[46, 47]	Membrane proteins
	PSSM400	400	✓	✗	[42]	Protein classification
	PSSM-AC	200	✓	✓	[33]	Secreted effector proteins
	PSSM_BLOCK	[b]	✓	✗	[48]	Protein self-interactions
	PSSM-CC	[b]	✓	✓	[33]	Secreted effector proteins
	PSSM_SEG	100	✓	✗	[49]	Protein fold recognition
	PSSM_SD	80	✓	✗	[49]	Protein fold recognition
	RPSSM	110	✓	✓	[50]	Protein structural classes
	Single_Average	400	✓	✗	[42]	Protein classification
	SVD_PSSM	20	✓	✗	[42]	Protein classification

[a]These features produce a matrix of features whose dimension varies based on the choice of the parameters.
[b]Feature vector dimension varies based on the choice of the parameters.

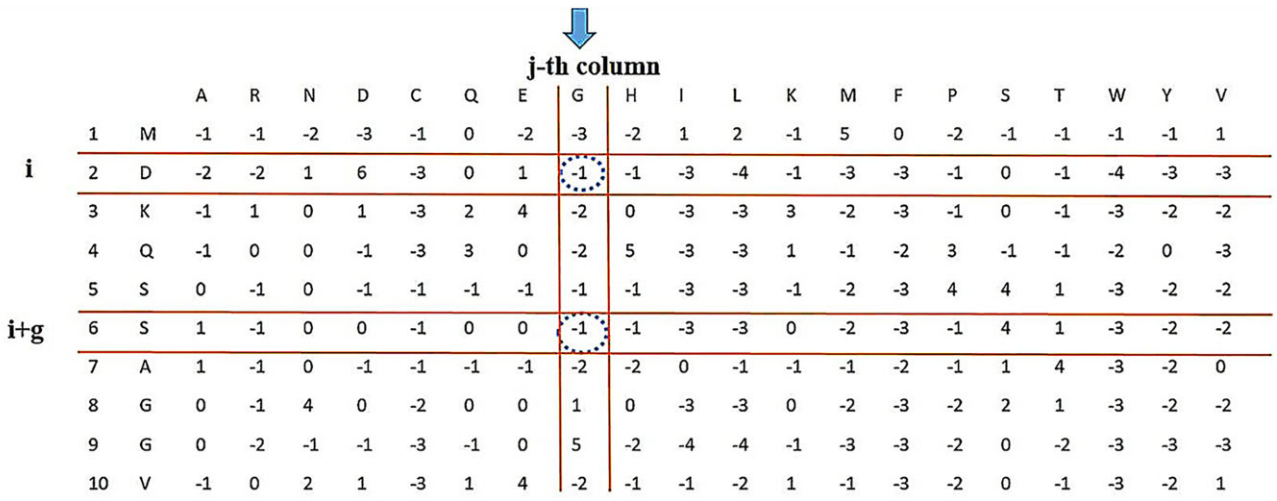
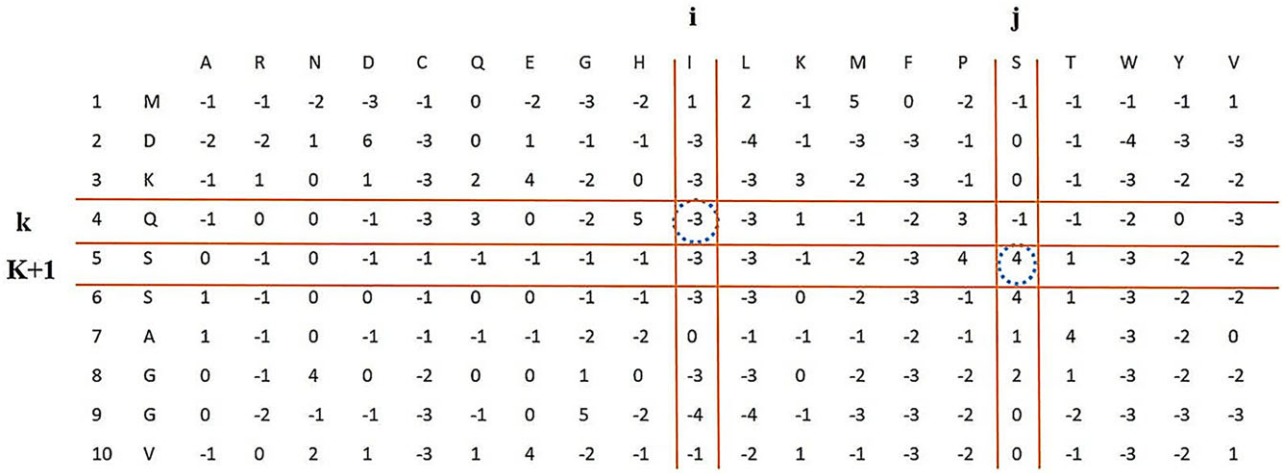
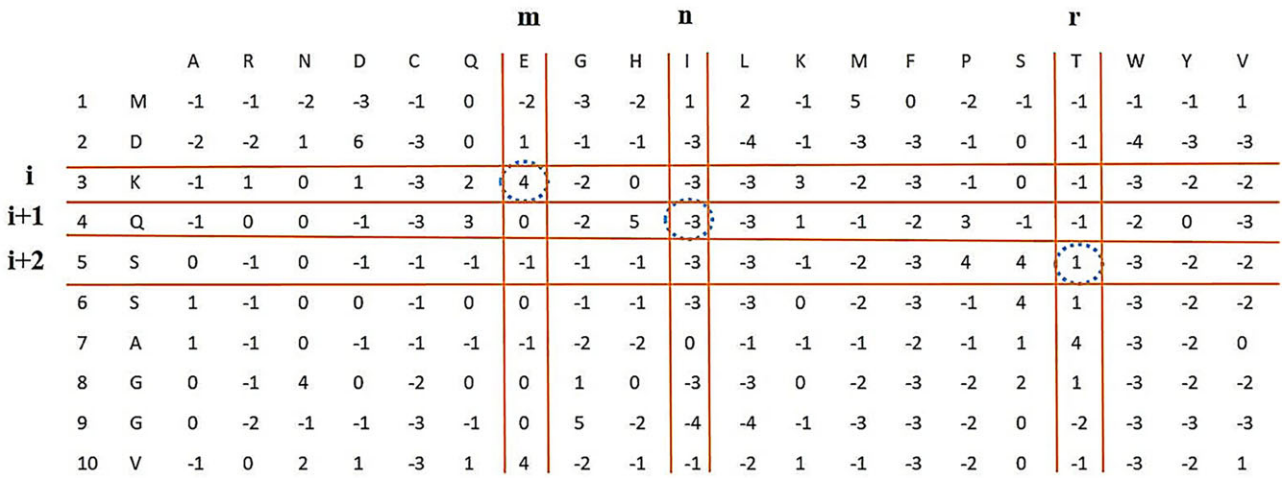


Figure 3: Extraction of PSSM_AC feature from PSSM. Here, the average of column j is −0.8, which is subtracted from −1 corresponding to the i, jth element, and −1 corresponding to the $i+g$, jth element. This results in −0.2 for both subtractions. The obtained numbers are multiplied to gain 0.04. This must be repeated in the range of $i=1$ to $L-g$ and the resulting numbers must be summed and finally divided by $L-g$.



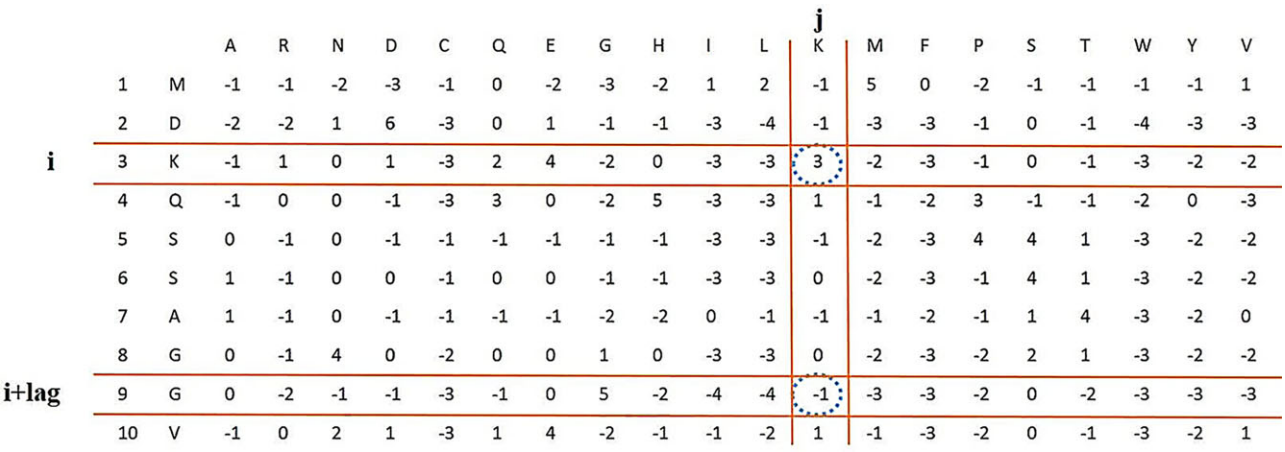
			A	R	N	D	C	Q	E	G	H	i I	L	K	M	F	P	j S	T	W	Y	V
	1	M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
	2	D	-2	-2	1	6	-3	0	1	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
	3	K	-1	1	0	1	-3	2	4	-2	0	-3	-3	3	-2	-3	-1	0	-1	-3	-2	-2
k	4	Q	-1	0	0	-1	-3	3	0	-2	5	-3	-3	1	-1	-2	3	-1	-1	-2	0	-3
K+1	5	S	0	-1	0	-1	-1	-1	-1	-1	-1	-3	-3	-1	-2	-3	4	4	1	-3	-2	-2
	6	S	1	-1	0	0	-1	0	0	-1	-1	-3	-3	0	-2	-3	-1	4	1	-3	-2	-2
	7	A	1	-1	0	-1	-1	-1	-1	-2	-2	0	-1	-1	-1	-2	-1	1	4	-3	-2	0
	8	G	0	-1	4	0	-2	0	0	1	0	-3	-3	0	-2	-3	-2	2	1	-3	-2	-2
	9	G	0	-2	-1	-1	-3	-1	0	5	-2	-4	-4	-1	-3	-3	-2	0	-2	-3	-3	-3
	10	V	-1	0	2	1	-3	1	4	-2	-1	-1	-2	1	-1	-3	-2	0	-1	-3	-2	1

Figure 4: Extraction of DPC_PSSM feature from PSSM. As shown in the figure, the values of two consecutive rows from different columns are multiplied (−12) and summed for the range of $k=1$ to $k=L-1$. The finally obtained number must be divided by $L-1$.



			A	R	N	D	C	Q	m E	G	H	n I	L	K	M	F	P	S	r T	W	Y	V
	1	M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
	2	D	-2	-2	1	6	-3	0	1	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
i	3	K	-1	1	0	1	-3	2	4	-2	0	-3	-3	3	-2	-3	-1	0	-1	-3	-2	-2
i+1	4	Q	-1	0	0	-1	-3	3	0	-2	5	-3	-3	1	-1	-2	3	-1	-1	-2	0	-3
i+2	5	S	0	-1	0	-1	-1	-1	-1	-1	-1	-3	-3	-1	-2	-3	4	4	1	-3	-2	-2
	6	S	1	-1	0	0	-1	0	0	-1	-1	-3	-3	0	-2	-3	-1	4	1	-3	-2	-2
	7	A	1	-1	0	-1	-1	-1	-1	-2	-2	0	-1	-1	-1	-2	-1	1	4	-3	-2	0
	8	G	0	-1	4	0	-2	0	0	1	0	-3	-3	0	-2	-3	-2	2	1	-3	-2	-2
	9	G	0	-2	-1	-1	-3	-1	0	5	-2	-4	-4	-1	-3	-3	-2	0	-2	-3	-3	-3
	10	V	-1	0	2	1	-3	1	4	-2	-1	-1	-2	1	-1	-3	-2	0	-1	-3	-2	1

Figure 5: Extraction of Trigram_PSSM feature from PSSM. The extraction of this feature is similar to DPC–PSSM extraction but instead of using two consecutive rows, the values of three consecutive rows in three different columns must be multiplied and summed. For the example provided here, the result of the multiplication is −12. This multiplication should be done for the range of $i=1$ to $L-2$ for each combination of three columns and the obtained values must be summed.

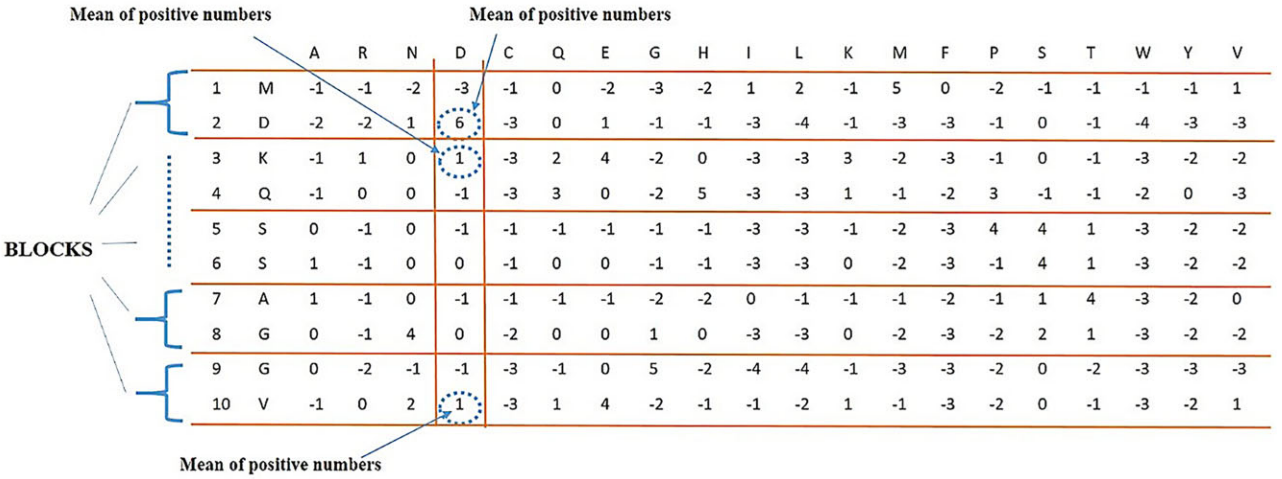


			A	R	N	D	C	Q	E	G	H	I	L	j K	M	F	P	S	T	W	Y	V
	1	M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
	2	D	-2	-2	1	6	-3	0	1	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
i	3	K	-1	1	0	1	-3	2	4	-2	0	-3	-3	3	-2	-3	-1	0	-1	-3	-2	-2
	4	Q	-1	0	0	-1	-3	3	0	-2	5	-3	-3	1	-1	-2	3	-1	-1	-2	0	-3
	5	S	0	-1	0	-1	-1	-1	-1	-1	-1	-3	-3	-1	-2	-3	4	4	1	-3	-2	-2
	6	S	1	-1	0	0	-1	0	0	-1	-1	-3	-3	0	-2	-3	-1	4	1	-3	-2	-2
	7	A	1	-1	0	-1	-1	-1	-1	-2	-2	0	-1	-1	-1	-2	-1	1	4	-3	-2	0
	8	G	0	-1	4	0	-2	0	0	1	0	-3	-3	0	-2	-3	-2	2	1	-3	-2	-2
i+lag	9	G	0	-2	-1	-1	-3	-1	0	5	-2	-4	-4	-1	-3	-3	-2	0	-2	-3	-3	-3
	10	V	-1	0	2	1	-3	1	4	-2	-1	-1	-2	1	-1	-3	-2	0	-1	-3	-2	1

Figure 6: Extraction of PSe_PSSM feature from PSSM. The first 20 values in this feature vector are the averages of 20 columns of PSSM. The remaining 300 values are computed by the mean square of differences between the i-th and i+lag-th rows for each column (lag values vary between 1 and 15). For $i=3$ and $i+\text{lag}=9$, the squared difference would be $(3-(-1))^2=4$. If $\text{lag}=6$, this will be calculated for the range of $i=1$ to $i=L-\text{lag}$, and the resulting values must be summed and divided by $L-\text{lag}$.

			A	R	N	D	C	Q	E	G	H	m: I	L	K	M	F	P	n: S	T	W	Y	V
	1	M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
	2	D	-2	-2	1	6	-3	0	1	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
i	3	K	-1	1	0	1	-3	2	4	-2	0	-3	-3	3	-2	-3	-1	0	-1	-3	-2	-2
	4	Q	-1	0	0	-1	-3	3	0	-2	5	-3	-3	1	-1	-2	3	-1	-1	-2	0	-3
	5	S	0	-1	0	-1	-1	-1	-1	-1	-1	-3	-3	-1	-2	-3	4	4	1	-3	-2	-2
	6	S	1	-1	0	0	-1	0	0	-1	-1	-3	-3	0	-2	-3	-1	4	1	-3	-2	-2
	7	A	1	-1	0	-1	-1	-1	-1	-2	-2	0	-1	-1	-1	-2	-1	1	4	-3	-2	0
	8	G	0	-1	4	0	-2	0	0	1	0	-3	-3	0	-2	-3	-2	2	1	-3	-2	-2
i+K	9	G	0	-2	-1	-1	-3	-1	0	5	-2	-4	-4	-1	-3	-3	-2	0	-2	-3	-3	-3
	10	V	-1	0	2	1	-3	1	4	-2	-1	-1	-2	1	-1	-3	-2	0	-1	-3	-2	1

Figure 7: Extraction of K-separated-bigram-PSSM feature from PSSM. This feature can be considered as an extension of the DPC feature. For each combination of two columns, the sum of multiplication of the i-th row corresponding to one column and the $i+k$-th row corresponding to the other column is computed where i varies between 1 and $L-K$. Here, for $i=3$ and $K=6$, the multiplication would be 0.



			A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
	1	M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
	2	D	-2	-2	1	6	-3	0	1	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
	3	K	-1	1	0	1	-3	2	4	-2	0	-3	-3	3	-2	-3	-1	0	-1	-3	-2	-2
	4	Q	-1	0	0	-1	-3	3	0	-2	5	-3	-3	1	-1	-2	3	-1	-1	-2	0	-3
	5	S	0	-1	0	-1	-1	-1	-1	-1	-1	-3	-3	-1	-2	-3	4	4	1	-3	-2	-2
	6	S	1	-1	0	0	-1	0	0	-1	-1	-3	-3	0	-2	-3	-1	4	1	-3	-2	-2
	7	A	1	-1	0	-1	-1	-1	-1	-2	-2	0	-1	-1	-1	-2	-1	1	4	-3	-2	0
	8	G	0	-1	4	0	-2	0	0	1	0	-3	-3	0	-2	-3	-2	2	1	-3	-2	-2
	9	G	0	-2	-1	-1	-3	-1	0	5	-2	-4	-4	-1	-3	-3	-2	0	-2	-3	-3	-3
	10	V	-1	0	2	1	-3	1	4	-2	-1	-1	-2	1	-1	-3	-2	0	-1	-3	-2	1

Figure 8: Extraction of AB-PSSM feature from PSSM. The first feature vector is obtained by placing 20 vectors corresponding to each block next to each other. For having these vectors, the row vectors (with length 20) related to each block are added together and the resulting vector is divided by the length of that block. For computing the second feature vector, the average of positive numbers in each column related to each block is calculated. Then, 20 values corresponding to 20 blocks are placed next to each other. By performing this procedure for each individual column, a feature vector with a length of 400 could be obtained.

		A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
1	M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1
2	D	-2	-2	1	6	-3	0	1	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3
3	K	-1	1	0	1	-3	2	4	-2	0	-3	-3	3	-2	-3	-1	0	-1	-3	-2	-2
4	Q	-1	0	0	-1	-3	3	0	-2	5	-3	-3	1	-1	-2	3	-1	-1	-2	0	-3
5	S	0	-1	0	-1	-1	-1	-1	-1	-1	-3	-3	-1	-2	-3	4	4	1	-3	-2	-2
6	S	1	-1	0	0	-1	0	0	-1	-1	-3	-3	0	-2	-3	-1	4	1	-3	-2	-2
7	A	1	-1	0	-1	-1	-1	-1	-2	-2	0	-1	-1	-1	-2	-1	1	4	-3	-2	0
8	G	0	-1	4	0	-2	0	0	1	0	-3	-3	0	-2	-3	-2	2	1	-3	-2	-2
9	G	0	-2	-1	-1	-3	-1	0	5	-2	-4	-4	-1	-3	-3	-2	0	-2	-3	-3	-3
10	V	-1	0	2	1	-3	1	4	-2	-1	-1	-2	1	-1	-3	-2	0	-1	-3	-2	1

Figure 9: Extraction of PSSM400 feature from PSSM. To calculate this feature, a sub-matrix representing the conservation of each standard amino acid will be computed. To obtain this sub-matrix, for each standard amino acid (here, the serine amino acid), all the corresponding columns are extracted. By calculating the average of columns in the extracted sub-matrix, a vector of length 20 will be acquired for each standard amino acid type. By putting the vectors (with the length of 20) for all 20 amino acids, the final feature vector with a length of 400 could be obtained.

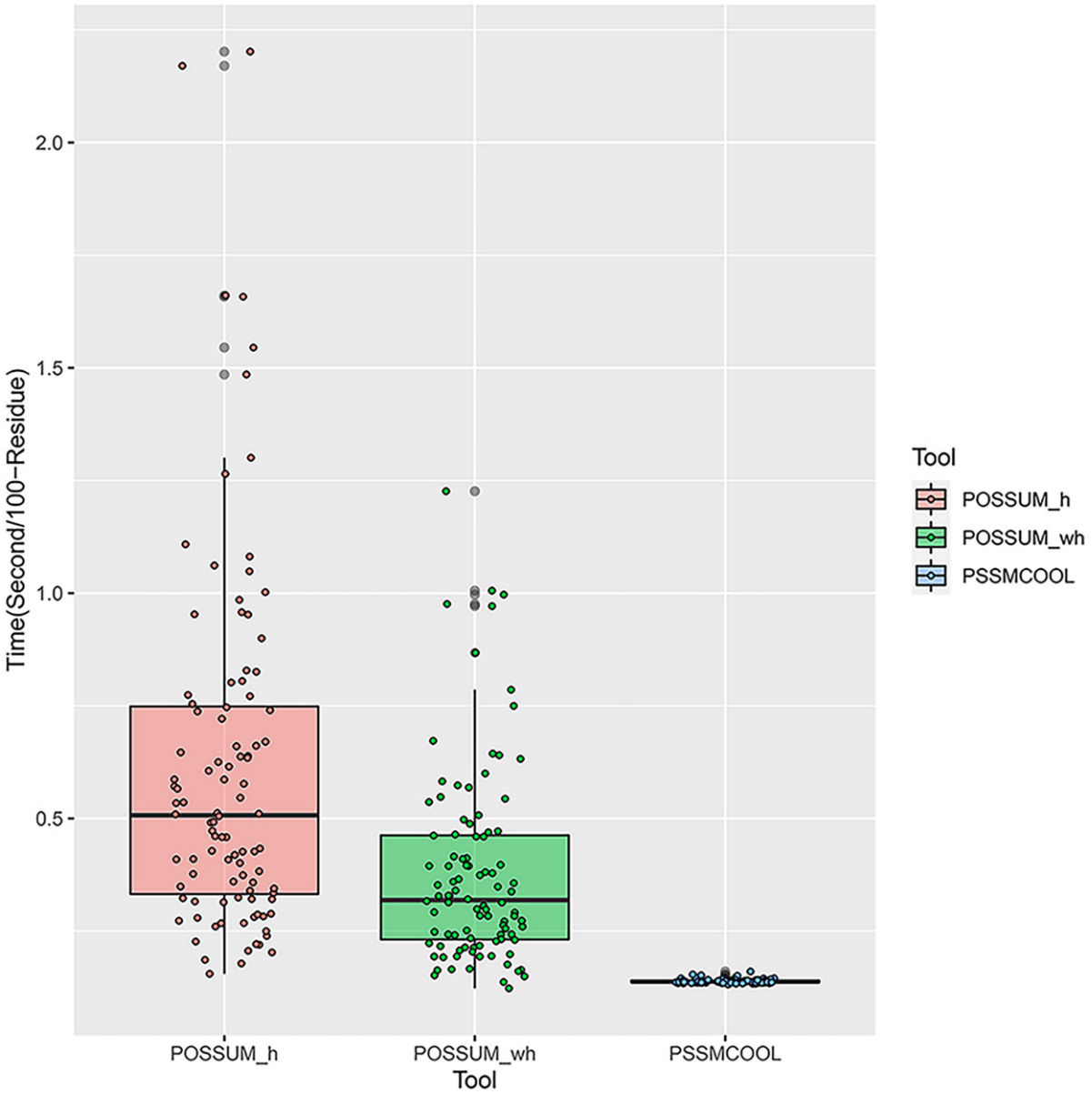


Figure 10: Comparison of run time between PSSMCOOL and POSSUM. POSSUM can be run in two modes. In the slower mode, it writes header for each extracted feature in the output files (POSSUM_h) and in the faster mode POSSUM writes features to the output file without headers (POSSUM_wh). Each point shows the average run time (in seconds) per 100 residues for each protein across all features.

AB-PSSM

The AB-PSSM feature was used for protein function prediction [28]. This feature consists of two types of feature vectors. At first, each protein sequence is divided into 20 equal parts, each of which is called a block. In each block, the row vectors of the PSSM related to that block are added together and the resulting final vector is divided by the length of that block, which is equal to 5% of protein length (see Fig. 8). Finally, concatenating these 20 vectors, the first feature vector of length 400 is obtained. For the second feature, in each block, the average of the positive numbers is computed for all 20 columns. Finally, these 400 averages will be used as the second feature vector.

CS-PSe-PSSM

This feature consists of a combination of several types of features; in general, the obtained feature vector would be of length 700 [29]. The sub-features that have been integrated as the single feature vector (CS-PSe-PSSM) are CSAAC, CSCM, segmented PsePSSM features, and segmented ACTPSSM.

SCSH2

This feature has been utilized for protein–protein interactions prediction [14]. To produce this feature vector, we need to extract the consensus sequence corresponding to the protein sequence based on the PSSM scores. Having placed these two sequences next to each other, a matrix with a dimension of $2\times L$ will be created. In the next step, each entry in this matrix is considered a node and connected to the two entries, which are immediately below it (except for the two entries in the last row). Finally, we will have a graph similar to a bipartite graph called the SCSH graph. Now in this graph, each path of length

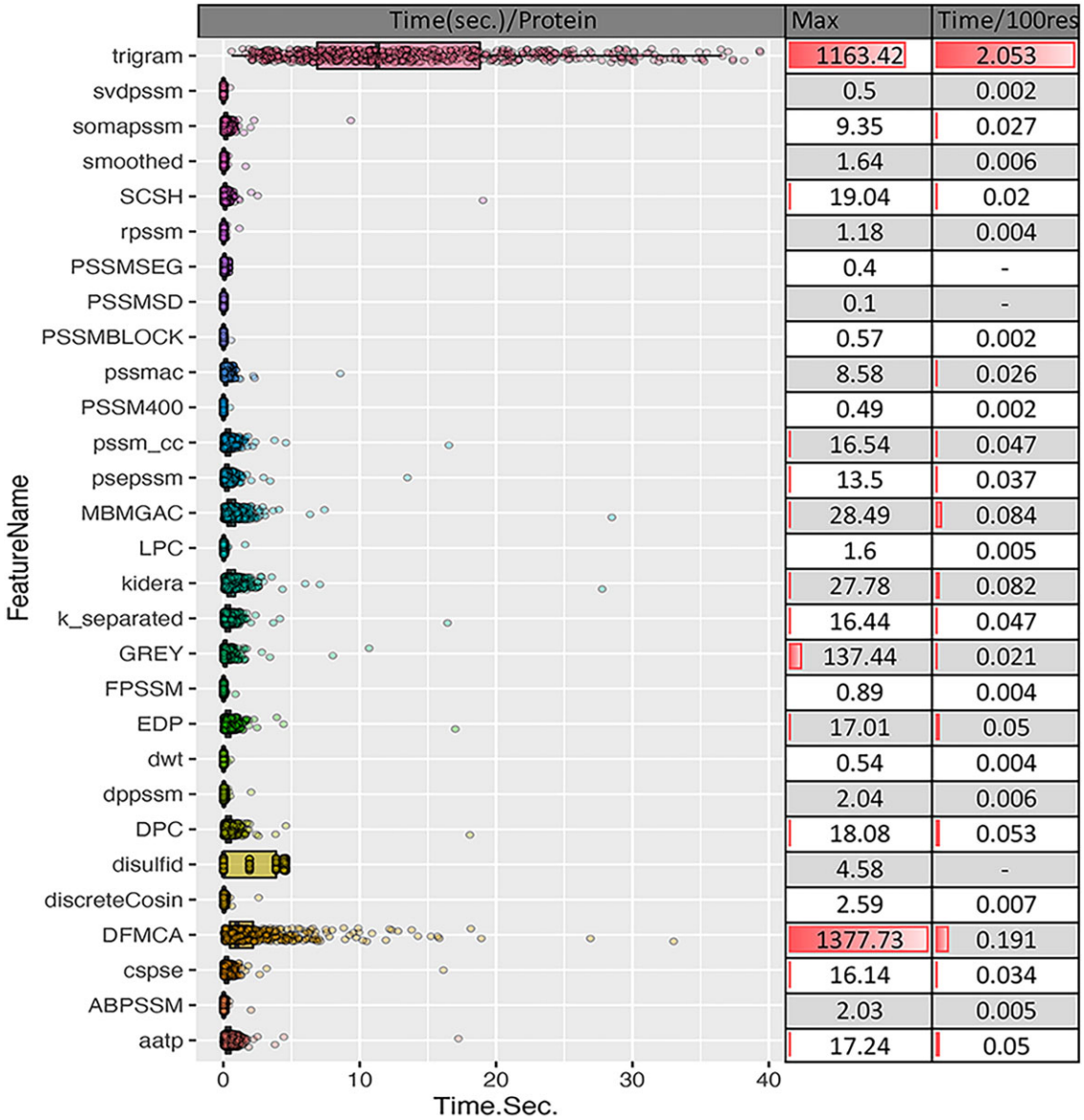


Figure 11: Feature extraction run time for features implemented in the PSSMCOOL. Trigram and DFMCA were the most computationally intensive features. However, the maximum run time corresponding to DFMCA did not exceed 23 min for a protein with >34 000 residues as the worst scenario. In addition, the average run time per 100 residues is 2.05 s for trigram and is <0.19 s for all other features. The configurations of the machine that was used for extracting features are as follows Windows 10 ×64; CPU: corei7 7700 HQ; RAM : 16 GB; and R version 4.1.2.

k specifies a $(k+1)$-mer. Finally, a k-mer composition feature vector can be obtained using this graph. k is equal to 2 in SCSH2.

PSSM400

This feature was employed in protein classification and protein–protein interaction prediction [42]. To generate this feature vector, for each of the standard amino acids, the corresponding rows in the PSSM are extracted and considered a sub-matrix (see Fig. 9). Now, for this sub-matrix, the column-wise average is considered the feature vector (a 20-dimensional vector). Finally, by putting together these feature vectors for all 20 amino acids, a feature vector of length 400 for each protein can be acquired.

SVD–PSSM

Singular value decomposition (SVD) is a general-purpose matrix factorization approach that has many useful applications in signal processing and statistics, as well as computational biology [42]. To compute this feature, SVD is applied to the PSSM representation of a protein for reducing its dimensionality. The final feature vector would be a 20-dimensional vector for all protein and peptide sequences with length ≥20.

Case study

We presented a case study and procedure that can be followed in order to use the PSSMCOOL for extracting features and building models for a prediction problem. For this case study, the

interactions between presynaptic proteins were extracted from the IntAct database [51]. As the first step, proteins with non-unique Uniprot accession numbers were discarded. For the positive set (protein–protein interactions), interactions from spoke expanded co-complexes and negative interactions were filtered out. Negative data set was constructed according to random pairing method as described in Refs. [2, 14, 52]. The final data set contained 1730 interactions (positive and negative) between 631 unique proteins. In this case study, "FPSSM2" function was used for feature extraction. Also, Bagged CART (treebag), and Single C5.0 Tree from caret package were used for classification. These two classifiers achieved 0.996 and 0.998 accuracy, respectively (R scripts corresponding to this case study are available at: https://github.com/BioCool-Lab/PSSMCOOL).

Run-time analysis

A set of human proteins was used to compare the time required for extracting features with PSSMCOOL and POSSUM. The human proteome was partitioned into 100 bins for assembling this set based on the protein lengths. Then, one random protein was selected from each bin and finally, a set comprised of 100 proteins was constructed. Figure 10 illustrates the performance of each tool for feature extraction in terms of run time. POSSUM can be run in two modes. In the slower mode, it writes header for each extracted feature in the output files (POSSUM_h) and in the faster mode, it writes features to the output file without headers (POSSUM_wh). Twenty-one features were used for making this comparison. Run time per 100 residues was calculated for each protein and these times were averaged across all these 21 features afterward. As Fig. 10 shows, the run times corresponding to PSSMCOOL are significantly lower than both POSSUM_h and POSSUM_wh. On average, PSSMCOOL only needs 0.14 s per 100 residues for feature extraction. It is worth mentioning that using POSSUM for several proteins requires writing command-line scripts, which does not seem to be very convenient for researchers who lack prior experience in Unix-based operating systems.

For almost all features implemented in PSSMCOOL, the corresponding run time is proportional to the protein length. However, this does not apply to three features, including disulfide, PSSMSEG, and PSSMSD, which are not dependent on the protein length. Their run time depends on the frequency of specific amino acids within the input proteins. Figure 11 shows the details of the run time corresponding to 29 different feature types in PSSMCOOL for 631 proteins used in the case study using a laptop with the following configuration; operating system: Windows 10 ×64; CPU: corei7 7700 HQ; RAM: 16 GB; R version 4.1.2. Evidently, trigram and DFMCA are the two most computationally intensive features ($P < 2.2E{-}22$; t-test). Nevertheless, the average run time for these two feature extractions was 2.05 and 0.19 s per 100 residues in the protein, respectively. Regarding the mean length of the human proteome, which is 553, on average feature extraction takes 1 s for each protein using a non-high-performing personal laptop.

Results and discussion

In this work, we present PSSMCOOL, a comprehensive, practical, and publicly accessible R package, developed to make the feature extraction of PSSMs feasible for researchers. Since it supplies 18 additional features, compared with the preceding available toolkit (POSSUM), it can greatly help for extracting features and developing new methods for the prediction of various protein attributes. The PSSMCOOL is freely accessible at: https://cran.r-project.org/web/packages/PSSMCOOL/index.html. Soaring data production has opened the door to the new applications of machine-learning methods in biology. One of the most significant steps toward the development of an efficient predictive model is feature extraction. The extraction of features by PSSMCOOL would be of great help for bioinformaticians who are interested in building predictive models for protein attribute prediction.

Availability of data and materials

Project name: PSSMCOOL;
Project home page: https://cran.r-project.org/web/packages/PSSMCOOL/index.html;
Operating systems: Windows, Linux, Mac;
Programming language: R;
Other requirements: R;
License: Not applicable;
Any restrictions to use by non-academics: No restrictions;
GitHub page: https://github.com/BioCool-Lab/PSSMCOOL.

Author contribution

The main idea of this work was represented by J.Z. A.M., J.Z., and S.M. implemented the package. M.K. and A.M. prepared the online documentation. S.S.A., M.K., and J.Z. reviewed and optimized the written R codes. Drafting and writing of the manuscript was carried out by all the authors. J.Z. supervised the work.

Conflict of interest statement. None declared.

References

1. Guo J, Chen H, Sun Z *et al.* A novel method for protein secondary structure prediction using dual-layer SVM and profiles. *Proteins Struct Funct Genet* 2004;**54**:738–43. Available from: http://doi.wiley.com/10.1002/prot.10634
2. Zahiri J, Yaghoubi O, Mohammad-Noori M *et al.* PPIevo: Protein–protein interaction prediction from PSSM based evolutionary information. *Genomics* 2013;**102**:237–42.
3. Chang DTH, Huang HY, Syu YT. Real value prediction of protein solvent accessibility using enhanced PSSM features. *BMC Bioinformatics* 2008;**9**:S12.Available from: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-9-S12-S12
4. De Brevern AG, Bornot A, Craveur P *et al.* PredyFlexy: Flexibility and local structure prediction from sequence. *Nucleic Acids Res* 2012;**40**:W317–22. Available from: http://www.dsimb.inserm.fr/dsimb_tools/
5. Kumar M, Gromiha MM, Raghava GPS. Identification of DNA-binding proteins using support vector machines and evolutionary profiles. *BMC Bioinformatics* 2007;**8**:1–10. Available from: https://link.springer.com/articles/10.1186/1471-2105-8-463
6. Xu R, Zhou J, Wang H *et al.* Identifying DNA-binding proteins by combining support vector machine and PSSM distance transformation. *BMC Syst Biol* 2015;**9**:1–12. Available from: https://link.springer.com/articles/10.1186/1752-0509-9-S1-S10
7. Dehzangi A, López Y, Lal SP *et al.* PSSM-Suc: Accurately predicting succinylation using position specific scoring matrix into bigram for feature extraction. *J Theor Biol* 2017;**425**:97–102.

8. Mundra P, Kumar M, Kumar KK *et al.* Using pseudo amino acid composition to predict protein subnuclear localization: Approached with PSSM. *Pattern Recognit Lett* 2007;**28**:1610–5.
9. Liu Y, Gong W, Yang Z *et al.* SNB-PSSM: A spatial neighbor-based PSSM used for protein–RNA binding site prediction. *Journal of Molecular Recognition* 2021;**34**:e2887. Available from: https://onlinelibrary.wiley.com/doi/abs/10.1002/jmr.2887
10. Ahmad S, Sarai A. PSSM-based prediction of DNA binding sites in proteins. *BMC Bioinformatics* 2005;**6**:33.Available from: https://pubmed.ncbi.nlm.nih.gov/15720719/
11. Mousavian Z, Khakabimamaghani S, Kavousi K. Drug–target interaction prediction from PSSM based evolutionary information. *J Pharmacol Toxicol Methods* 2016;**78**:42–51. Available from: https://pubmed.ncbi.nlm.nih.gov/26592807/
12. Chou KC. Some remarks on protein attribute prediction and pseudo amino acid composition. *J Theor Biol* 2011;**273**:236–47.
13. Emamjomeh A, Goliaei B, Zahiri J. Predicting protein–protein interactions between human and hepatitis C virus via an ensemble learning method. *Mol Biosyst* 2014;**10**:3147–54. Available from: https://pubs.rsc.org/en/content/articlehtml/2014/mb/c4mb00410h
14. Zahiri J, Mohammad-Noori M, Ebrahimpour R *et al.* LocFuse: Human protein–protein interaction prediction via classifier fusion using protein localization information. *Genomics* 2014;**104**:496–503.
15. An Y, Wang J, Li C *et al.* Comprehensive assessment and performance improvement of effector protein predictors for bacterial secretion systems III, IV and VI. *Brief Bioinform* 2018;**19**:148–61. Available from: https://academic.oup.com/bib/article-lookup/doi/10.1093/bib/bbw100
16. Liu B, Liu F, Fang L *et al.* repRNA: A web server for generating various feature vectors of RNA sequences. *Mol Genet Genomics* 2016;**291**:473–81. Available from: https://link.springer.com/article/10.1007/s00438-015-1078-7
17. Liu B, Liu F, Fang L *et al.* repDNA: A Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects. *Bioinformatics* 2015;**31**:1307–9. Available from: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btu820
18. Liu B, Liu F, Wang X *et al.* Pse-in-One: A web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nucleic Acids Res* 2015;**43**:W65–71. Available from: https://academic.oup.com/nar/article/43/W1/W65/2467922
19. Liu B, Wu H, Chou K-C. Pse-in-One 2.0: An improved package of web servers for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nucleic Acids Res* 2017;**09**:67–91. Available from: http://creativecommons.org/licenses/by/4.0/
20. Liu B, Wu H, Zhang D *et al.* Pse-analysis: A python package for DNA/RNA and protein/peptide sequence analysis based on pseudo components and kernel methods. *Oncotarget* 2017;**8**:13338–43. Available from: /pmc/articles/PMC5355101/
21. Shen HB, Chou KC. PseAAC: A flexible web server for generating various kinds of protein pseudo amino acid composition. *Anal Biochem* 2008;**373**:386–8.
22. Cao D-S, Xu Q-S, Liang Y-Z. Propy: A tool to generate various modes of Chou's PseAAC. *Bioinformatics* 2013;**29**:960–2. Available from: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btt072
23. Li ZR, Lin HH, Han LY *et al.* PROFEAT: A web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic Acids Res* 2006;**34**:W32–7. Available from: http://jing.cz3.nus.edu.sg/cgi-bin/prof/prof.cgi
24. Xiao N, Cao D-S, Zhu M-F. protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences. *Bioinformatics* 2015;**31**:1857–9. Available from: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btv042
25. Wang J, Yang B, Revote J *et al.* POSSUM: A bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. *Bioinformatics* 2017;**33**:2756–8. Available from: https://academic.oup.com/bioinformatics/article/33/17/2756/3813283
26. Liu T, Zheng X, Wang J. Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile. *Biochimie* 2010;**92**:1330–4.
27. Zhang S, Ye F, Yuan X. Using principal component analysis and support vector machine to predict protein structural class for low-similarity sequences via PSSM. *J Biomol Struct Dyn* 2012;**29**:1138–46. Available from: https://www.tandfonline.com/doi/abs/10.1080/07391102.2011.672627
28. Jeong JC, Lin X, Chen XW. On position-specific scoring matrix for protein function prediction. *IEEE/ACM Trans Comput Biol Bioinform* 2011;**8**:308–15.
29. Liang Y, Liu S, Zhang S. Prediction of protein structural classes for low-similarity sequences based on consensus sequence and segmented PSSM. *Comput Math Methods Med* 2015;**2015**:1–9. Available from: https://pubmed.ncbi.nlm.nih.gov/26788119/
30. Mapes NJ, Rodriguez C, Chowriappa P. Residue adjacency matrix based feature engineering for predicting cysteine reactivity in proteins. *Comput Struct Biotechnol J* 2019;**17**:90–100.
31. Fang C, Noguchi T, Yamana H. Condensing position-specific scoring matrixs by the Kidera factors for ligand-binding site prediction. *Int J Data Min Bioinform* 2015;**12**:70–84.
32. Liang Y, Liu S, Zhang S. Prediction of protein structural class based on different autocorrelation descriptors of position specific scoring matrix. *Match* 2015;**73**:765–84.
33. Zou L, Nan C, Hu F. Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles. *Bioinformatics* 2013;**29**:3135–42.
34. Cheng CW, Su ECY, Hwang JK *et al.* Predicting RNA-binding sites of proteins using support vector machines and evolutionary information. *BMC Bioinformatics* 2008;**9**:S6.Available from: http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-9-S12-S6
35. Liang Y, Zhang S, Ding S. Accurate prediction of Gram-negative bacterial secreted protein types by fusing multiple statistical features from PSI-BLAST profile. *SAR QSAR Environ Res* 2018;**29**:469–81.
36. Saini H, Raicar G, Lal S *et al.* Protein fold recognition using genetic algorithm optimized voting scheme and profile bigram. *J Softw* 2016;**11**:756–67. Available from: http://repository.usp.ac.fj/9657/
37. Wang Y, Ding Y, Tang J *et al.* CrystalM: A multi-view fusion approach for protein crystallization prediction. *IEEE/ACM Trans Comput Biol Bioinform* 2021;**18**:325–35.
38. Zhang L, Zhao X, Kong L. Predict protein structural class for low-similarity sequences by evolutionary difference information into the general form of Chou[U+05F3]s pseudo amino acid composition. *J Theor Biol* 2014;**355**:105–10. Available from: https://www.sciencedirect.com/science/article/pii/S0022519314002173
39. Li L, Cui X, Yu S *et al.* PSSP-RFE: Accurate prediction of protein structural class by recursive feature extraction from PSI-BLAST

profile, physical-chemical property and functional annotations. *PLoS ONE* 2014;**9**:e92863.
40. Liang Y, Zhang S. Predict protein structural class by incorporating two different modes of evolutionary information into Chou's general pseudo amino acid composition. *J Mol Graph Model* 2017;**78**:110–7. Available from: https://www.sciencedirect.com/science/article/pii/S1093326317306770
41. Paliwal KK, Sharma A, Lyons J. A tri-gram based feature extraction technique using linear probabilities of position specific scoring matrix for protein fold recognition. *IEEE Trans Nanobiosci* 2014;**13**:44–50. Available from: https://ieeexplore.ieee.org/abstract/document/6750119/
42. Nanni L, Lumini A, Brahnam S. An empirical study of different approaches for protein classification. *Sci World J* 2014; **2014**:1.Available from: https://www.hindawi.com/journals/tswj/2014/236717/abs/
43. Wang L, You ZH, Xia SX *et al*. Advancing the prediction accuracy of protein–protein interactions by utilizing evolutionary information from position-specific scoring matrix and ensemble classifier. *J Theor Biol* 2017;**418**:105–10. Available from: https://www.sciencedirect.com/science/article/pii/S0022519317300036
44. Juan EYT, Li WJ, Jhang JH *et al*. Predicting protein subcellular localizations for Gram-negative bacteria using DP-PSSM and support vector machines. In: *Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems*, CISIS 2009, 2009, 836–41. Available from: https://ieeexplore.ieee.org/abstract/document/5066887/
45. Xiao X, Hui M, Liu Z. iAFP-Ense: An ensemble classifier for identifying antifreeze protein by incorporating grey model and PSSM into PseAAC. *J Membr Biol* 2016;**249**:845–54.
46. Yu DJ, Hu J, Wu XW *et al*. Learning protein multi-view features in complex space. *Amino Acids* 2013;**44**:1365–79. Available from: https://link.springer.com/article/10.1007/s00726-013-1472-6
47. Chou KC, Shen HB. MemType-2L: A web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. *Biochem Biophys Res Commun* 2007;**360**:339–45.
48. An JY, Zhang L, Zhou Y *et al*. Computational methods using weighed-extreme learning machine to predict protein self-interactions with protein evolutionary information. *J Cheminform* 2017;**9**:47.Available from: https://jcheminf.biomedcentral.com/articles/10.1186/s13321-017-0233-z
49. Dehzangi A, Paliwal K, Lyons J *et al*. A segmentation-based method to extract structural and evolutionary features for protein fold recognition. *IEEE/ACM Trans Comput Biol Bioinform* 2014;**11**:186–96. Available from: https://ieeexplore.ieee.org/abstract/document/6693731/
50. Ding S, Yan S, Qi S *et al*. A protein structural classes prediction method based on PSI-BLAST profile. *J Theor Biol* 2014;**353**: 19–23. Available from: https://www.sciencedirect.com/science/article/pii/S0300908413003271
51. Kerrien S, Aranda B, Breuza L *et al*. The IntAct molecular interaction database in 2012. *Nucleic Acids Res* 2012;**40**:D841–6. Available from: https://academic.oup.com/nar/article/40/D1/D841/2903045
52. Zahiri J, Bozorgmehr J, Masoudi-Nejad A. Computational prediction of protein–protein interaction networks: Algorithms and resources. *Curr Genomics* 2013;**14**:397–414. Available from: https://www.ingentaconnect.com/content/ben/cg/2013/00000014/00000006/art00008