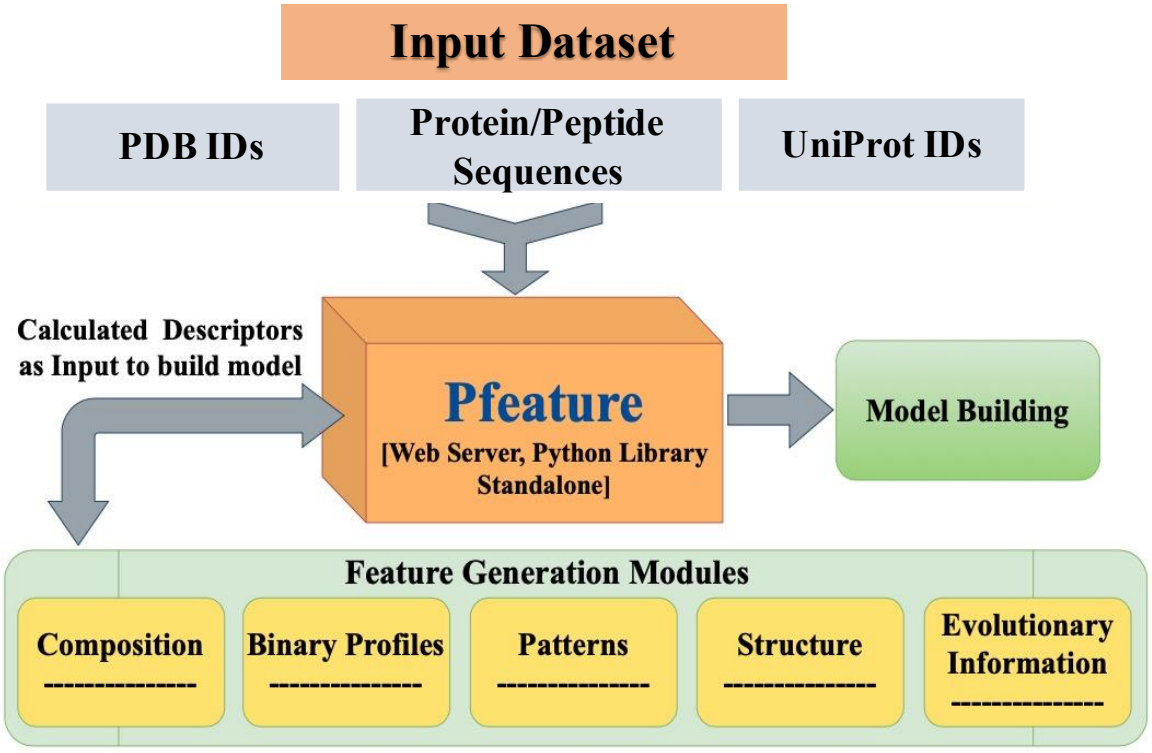
Pfeature: A platform for computing wide range of protein features and building prediction models

---

Pfeature Manual

Raghava's Group



Department of Computational Biology,
Indraprastha Institute of Information Technology,
New Delhi, India

Overview of Pfeature

One of the major challenge in the field of bioinformatics is to identify structural and functional properties of a protein from its primary sequence. In the past number of methods have been developed to annotate a protein at sequence level as well as at residue level. Protein level annotation includes identification of a protein having a specific function like prediction of nuclear proteins, toxins, allergens, membrane proteins. Residue level annotation includes prediction of residues having a specific function or structure, for example prediction of interacting residues, secondary structure of residues. In order to annotation a protein, one need to develop models mainly using machine learning techniques to classify or predict a protein having specific function. In order to develop machine learning based models one need to compute features of protein called protein descriptors. Numerous methods have been developed in the past to compute features of a protein that includes PROFEAT, PyBioMed, iFeature, protr, Rcpi, propy. Though existing methods provides wide range of options to compute large number of protein features. Still there are number of features which are not included in any existing methods particularly feature required for annotating protein at residues level. Residue level annotation means that you wants to predict function of residues in a protein like DNA interacting residues in a protein.

In order to facilitate the scientific community, we have developed a software package Pfeature, which computes more than 50,000 features, required for predicting the function of a protein and its residues. It has five major modules, namely composition, binary profiles, evolutionary information, structural, and model building. Our composition module compute all existing compositional features, plus novel features like Shannon entropy, residue repeats. The binary profile of amino acid sequences provides complete information, including the order of residues or type of residues suitable to predict the function of a protein at the residue level. Evolutionary module compute composition of PSSM profile generated using PSI-BLAST. Structural features of a protein can be computed from its secondary and tertiary structure using structural module. Model building module implement various machine learning tools for developing prediction models as well as feature processing options. It also allows generating overlapping patterns and feature from the whole protein or its parts. It is available in the form of a web server as at https://webs.iiitd.edu.in/raghava/pfeature/ , as well as a python library and standalone package.

This package has been developed at Prof. Gajendra P. S. Raghava's group, please contact at raghava@iiitd.ac.in or visit http://webs.iiitd.edu.in/raghava/ ; if you have any query

Table of Content

Chapter 1.0
Overview

1.1 What is Pfeature?

The Pfeature is a package developed for computing wide range of protein features for annotating proteins at residue as well as at residue level. It is developed by Raghava's Group, Indraprastha Institute of Information Technology, New Delhi, India. One of the challenge in developing prediction model using machine learning algorithms is to compute descriptors of a protein. Pfeature is a comprehensive and feature-rich package that can be used to accomplish various classification tasks. Pfeature can calculate features from sequence as well as the structure of the proteins or peptides. Pfeature comprises five influential modules for calculating features, such as composition-based, binary-profile based, evolutionary information based, structure-based, and pattern-based features. Pfeature also provides another module, named model building, which can generate classification and regression models using the generated features. We anticipate that this package can be employed for investigating issues regarding structures, and functions of various molecular data in the context of bioinformatics and systems biology. Moreover, to aid the scientific community, Pfeature is available as a webserver at https://webs.iiitd.edu.in/raghava/pfeature/, as a standalone package at https://webs.iiitd.edu.in/raghava/pfeature/pfeature_standalone.zip, and as a python-library at https://webs.iiitd.edu.in/raghava/pfeature/Pfeature_Library.zip. Pfeature is also available at GitHub repository at https://github.com/raghavagps/Pfeature.

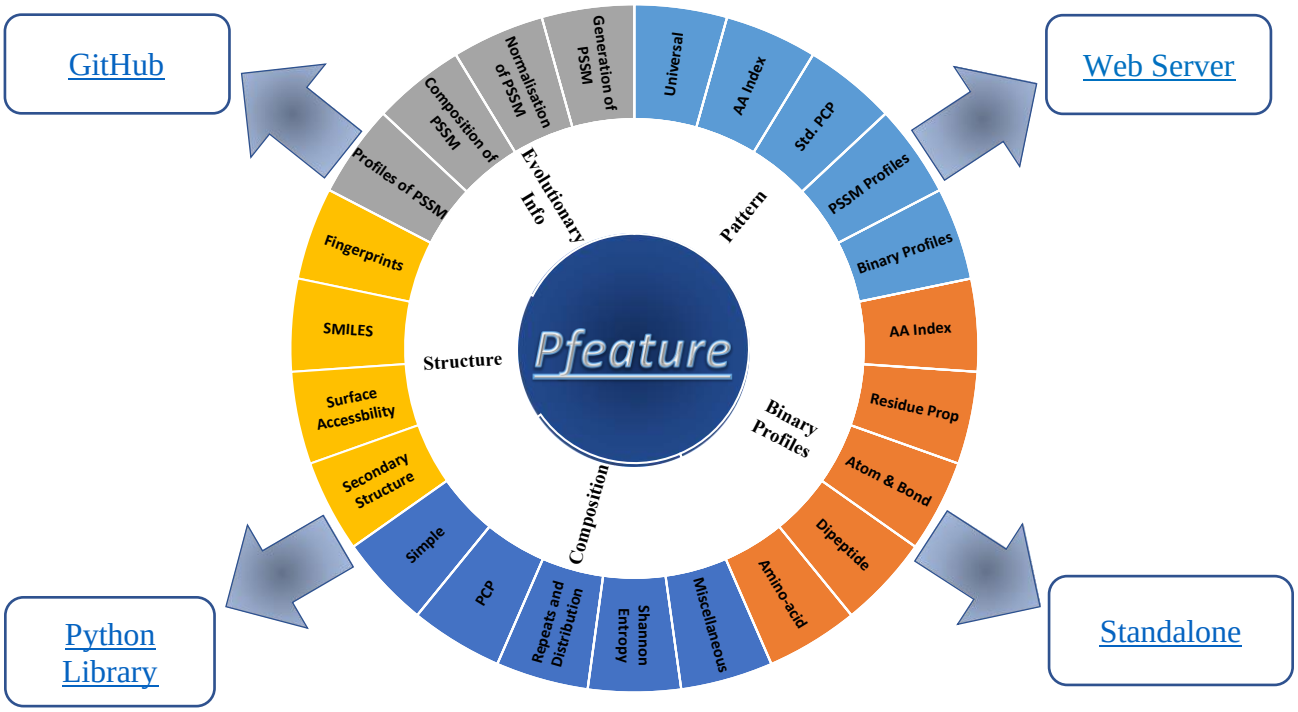


Figure 1.1: Modules and sub-modules of Pfeature

1.2 Who uses Pfeature?

The Pfeature can be useful for researchers, scientists, or students from various biomedical disciplines, who can use it to investigate and represent different information related to proteins/peptides data under consideration. Pfeature will help them to explore problems regarding the protein functions and its capabilities.

1.3 Objectives

The fundamental objective of Pfeature is to ease the process of feature calculation and prediction model generation, for researchers with very little or no knowledge of computer and machine learning.

Major functions of Pfeature:

- Calculation of composition and physicochemical property based features of proteins/peptides.
- Computation of binary profiles of proteins or protein segments
- Computation of evolutionary information of proteins in form of PSSM profiles.

- Identification of structural features of proteins from their structure
- Determination of pattern-based features of proteins
- Modules for developing classification and regression models
- Facility to extract protein sequence from following databases; PDB, UniProt ID.

1.4 Overview of Features

Following table shows the features integrated in Pfeature, it also shows dimension or vector size of each feature.

Table 1.1: Shows the major descriptors integrated in Pfeature; it also includes vector size of each descriptor/feature.

COMPOSITION BASED DESCRIPTORS					
Type of Composition	**Vector Size**	**Type of Composition**	**Vector Size**	**Type of Composition**	**Vector Size**
Amino acid	20	Shannon Entropy	1	Composition enhanced Transition Distribution	189
Dipeptide	400	Atom & Bond	9	Shannon Entropy at residue level	20
Tripeptide	8000	Physicochemical	30	Conjoint Triad Calculation	343
Atom	5	Pseudo Amino-acid	20 + λ	Shannon Entropy of Property	24
Bond	4	Distance Distribution	20	Amphiphilic Pseudo Amino-acid	20 + (λ*3)
Autocorrelation	3*γ	Quasi-Sequence Order	40 + (λ*2)	Repetitive Residue Information	20
AAIndex	γ	Sequence Order Coupling	λ*2		
BINARY PROFILE BASED DESCRIPTORS					
Amino acid	20*L	Dipeptide	400*L	Property	25*L
Atom	5*η	Bond	4*ε	Atom & Bond	(5*η)+(4*ε)
AAIndex	553*L				

EVOLUTIONARY INFORMATION BASED DESCRIPTORS (PSSM Profiles)					
Generation	L X 21	Normalization	L X 21	Composition	400
Profile	L X 21				
STRUCTURE					
Fingerprints	14532	SMILES	1	Surface Accessibility	9
Secondary Structure	3				

L: length of protein; λ: The number depends upon the choice of maxlag; γ: Number depends upon the choice of amino acid indices;η: Number of atoms; ε: Number of bonds

Chapter 2.0
Getting Started With Pfeature

This chapter is written to assist Pfeature users in using it effectively. It allow user to use this package in three flavors; i) python library, ii) web server and iii) standalone. If you are new user and wish to use it without doing any installation. We will advise you to use web server of Pfeature via following URL http://webs.iiitd.edu.in/raghava/pfeature/ . If you are interested in running Pfeature on large dataset at your local machine. We will advise you to use standalone version of Pfeature see https://webs.iiitd.edu.in/raghava/pfeature/stand.php. In case you are advance user and wish to call functions of Pfeature from your Python code. We will advise you to install Python-Library of Pfeature. Following section provides detail description of using these three modules.

2.1 Installation of Python-Library

The user can download the python-library of Pfeature from https://github.com/raghavagps/Pfeature/tree/master/PyLib or https://webs.iiitd.edu.in/raghava/pfeature/Pfeature_Library.zip. The python-library has been successfully tested on Mac, Linux, Windows and Centos operating systems.

Installation of Pfeature is simple as explained below:

On Microsoft Windows:

1. Download Pfeature.zip
2. extract or uncompress the Pfeature.zip
3. change directory to Pfeature
4. Run the command: python3 setup.py install

On Mac/Linux:

1. Download Pfeature.zip
2. unzip the Pfeature.zip
3. change directory to Pfeature
4. Run the command: python3 setup.py install or sudo python3 setup.py install

On Centos:

1. Download Pfeature.zip
2. unzip the Pfeature.zip
3. change directory to Pfeature
4. Run the command: python3 setup.py install

Pfeature python library demo for calculating amino acid composition of whole protein

```
>>> from Pfeature.pfeature import * # To import all the functions of Pfeature
>>> from Pfeature.pfeature import aac_wp # AAC for whole protein
>>> aac_wp("example.seq", "example.out")
```

```
PAAAAQAVAGLAPVAAEQMALWMRLL
PLLALLALWGPDPAAAFVNQHLCGSH
LVEALYLVCGERGFFYTPKTRREAED
LQVGQVELGGGPGAGSLQPLALEGSL
QKRGIVEQCCTSICSLYQLENYCN
MMFRSVIPVLLFLIPLLLSAQAANSLR
ACGPALMDMLRVACPNGFNSMFAKRGTL
GLFDYEDHLADLDSSESHHMNSLSSIRRD
FRGVVDSCCRKSCSFSTLRAYCDS
```

aac_wp

example.seq

```
AAC_A,AAC_C,AAC_D,AAC_E,AAC_F,AAC_G,AAC_H,AAC_I,AAC_K,AAC_L,AAC_M,AAC_N,AAC_P,AAC_Q,AAC_R,AAC_S,AAC_T,AAC_V,AAC_W,AAC_Y
38.46,0.0,0.0,3.85,0.0,3.85,0.0,0.0,0.0,15.38,7.69,0.0,7.69,7.69,3.85,0.0,0.0,7.69,3.85,0.0
19.23,3.85,3.85,0.0,3.85,7.69,7.69,0.0,0.0,23.08,0.0,3.85,11.54,3.85,0.0,3.85,0.0,3.85,3.85,0.0
7.69,3.85,3.85,15.38,7.69,7.69,0.0,0.0,0.0,3.85,11.54,0.0,0.0,0.0,3.85,0.0,11.54,0.0,7.69,7.69,0.0,7.69
7.69,0.0,0.0,7.69,0.0,26.92,0.0,0.0,0.0,0.0,23.08,0.0,0.0,0.0,7.69,11.54,0.0,7.69,0.0,7.69,0.0,0.0,0.0
0.0,16.67,0.0,8.33,0.0,4.17,0.0,8.33,4.17,8.33,0.0,8.33,0.0,12.5,4.17,8.33,4.17,4.17,0.0,8.33
11.11,0.0,0.0,0.0,7.41,0.0,0.0,0.0,7.41,0.0,25.93,7.41,3.7,7.41,3.7,7.41,11.11,0.0,7.41,0.0,0.0,0.0
14.29,7.14,3.57,0.0,7.14,10.71,0.0,0.0,0.0,3.57,10.71,10.71,7.14,7.14,0.0,7.14,3.57,3.57,3.57,0.0,0.0,0.0
3.45,0.0,17.24,6.9,3.45,3.45,10.34,3.45,0.0,0.0,13.79,3.45,3.45,3.45,0.0,0.0,0.0,6.9,20.69,0.0,0.0,0.0,0.0,0.0,3.45
4.17,16.67,8.33,0.0,8.33,4.17,0.0,0.0,0.0,4.17,4.17,0.0,0.0,0.0,0.0,0.0,12.5,20.83,4.17,8.33,0.0,0.0,4.17
```

example.out

Table 2.1: Usage of functions involved in Pfeature python library

Name of the Function	Function call
Composition Based Features	
Amino-acid Composition	aac_wp(inputfile,outputfile)
Amino-acid Composition of N-Terminal	aac_nt(inputfile,outputfile,number)
Amino-acid Composition of C-Terminal	aac_ct(inputfile,outputfile,number)
Amino-acid Composition of Rest	aac_rt(inputfile,outputfile,number1,number2)
Amino-acid Composition of N- and C-Terminal	aac_nct(inputfile,outputfile,number)

Amino-acid Composition of split	aac_st(inputfile,outputfile,number)
Dipeptide Composition	dpc_wp(inputfile,outputfile,order)
Dipeptide Composition of N-Terminal	dpc_nt(inputfile,outputfile,number,order)
Dipeptide Composition of C-Terminal	dpc_ct(inputfile,outputfile,number,order)
Dipeptide Composition of Rest	dpc_rt(inputfile,outputfile,number1,number2,order)
Dipeptide Composition of N- and C-Terminal	dpc_nct(inputfile,outputfile,number,order)
Dipeptide Composition of split	dpc_st(inputfile,outputfile,lambda,number_of_splits)
Tripeptide Composition	tpc_wp(inputfile,outputfile)
Tripeptide Composition of N-Terminal	tpc_nt(inputfile,outputfile,number)
Tripeptide Composition of C-Terminal	tpc_ct(inputfile,outputfile,number)
Tripeptide Composition of Rest	tpc_rt(inputfile,outputfile,number1,number2)
Tripeptide Composition of N- and C-Terminal	tpc_nct(inputfile,outputfile,number)
Tripeptide Composition of split	tpc_st(inputfile,outputfile,number)
Atom Composition	atc_wp(inputfile,outputfile)
Atom Composition of N-Terminal	atc_nt(inputfile,outputfile,number)
Atom Composition of C-Terminal	atc_ct(inputfile,outputfile,number)
Atom Composition of Rest	atc_rt(inputfile,outputfile,number1,number2)
Atom Composition of N- and C-Terminal	atc_nct(inputfile,outputfile,number)
Atom Composition of split	atc_st(inputfile,outputfile ,number)
Bond Composition	btc_wp(inputfile,outputfile)
Bond Composition of N-Terminal	btc_nt(inputfile,outputfile,number)
Bond Composition of C-Terminal	btc_ct(inputfile,outputfile,number)
Bond Composition of Rest	btc_rt(inputfile,outputfile,number1,number2)
Bond Composition of N- and C-Terminal	btc_nct(inputfile,outputfile,number)
Bond Composition of split	btc_st(inputfile,outputfile ,number)
Physico-Chemical Properties Composition	pcp_wp(inputfile,outputfile)
Physico-Chemical Properties Composition of N-Terminal	pcp_nt(inputfile,outputfile,number)
Physico-Chemical Properties Composition of C-Terminal	pcp_ct(inputfile,outputfile,number)
Physico-Chemical Properties Composition of Rest	pcp_rt(inputfile,outputfile,number1,number2)
Physico-Chemical Properties Composition of N- and C-Terminal	pcp_nct(inputfile,outputfile,number)
Physico-Chemical Properties Composition of split	pcp_st(inputfile,outputfile ,number)
Amino-acid index Composition	aai_wp(inputfile,outputfile)
Amino-acid index Composition of N-Terminal	aai_nt(inputfile,outputfile,number)
Amino-acid index Composition of C-Terminal	aai_ct(inputfile,outputfile,number)
Amino-acid index Composition of Rest	aai_rt(inputfile,outputfile,number1,number2)

Amino-acid index Composition of N- and C-Terminal	aai_nct(inputfile,outputfile,number)
Amino-acid index Composition of split	aai_st(inputfile,outputfile ,number)
Repetitive Residue Information	rri_wp(inputfile,outputfile)
Repetitive Residue Information of N-Terminal	rri_nt(inputfile,outputfile,number)
Repetitive Residue Information of C-Terminal	rri_ct(inputfile,outputfile,number)
Repetitive Residue Information of Rest	rri_rt(inputfile,outputfile,number1,number2)
Repetitive Residue Information of N- and C-Terminal	rri_nct(inputfile,outputfile,number)
Repetitive Residue Information of split	rri_st(inputfile,outputfile ,number)
Distance Distribution of Residues	ddr_wp(inputfile,outputfile)
Distance Distribution of Residues for N-Terminal	ddr_nt(inputfile,outputfile)
Distance Distribution of Residues for C-Terminal	ddr_ct(inputfile,outputfile)
Distance Distribution of Residues for Rest	ddr_rt(inputfile,outputfile)
Distance Distribution of Residues for N- and C-Terminal	ddr_nct(inputfile,outputfile)
Distance Distribution of Residues for split	ddr_st(inputfile,outputfile, number of splits)
Physico-chemical properties repeat Composition	pri_wp(inputfile,outputfile)
Physico-chemical properties repeat Composition of N-Terminal	pri_nt(inputfile,outputfile,number)
Physico-chemical properties repeat Composition of C-Terminal	pri_ct(inputfile,outputfile,number)
Physico-chemical properties repeat Composition of Rest	pri_rt(inputfile,outputfile,number1,number2)
Physico-chemical properties repeat Composition of N- and C-Terminal	pri_nct(inputfile,outputfile,number)
Physico-chemical properties repeat Composition of Split	pri_st(inputfile,outputfile ,number)
Shannon Entropy	sep_wp(inputfile,outputfile)
Shannon Entropy of N-Terminal	sep_nt(inputfile,outputfile,number)
Shannon Entropy of C-Terminal	sep_ct(inputfile,outputfile,number)
Shannon Entropy of Rest	sep_rt(inputfile,outputfile,number1,number2)
Shannon Entropy of N- and C-Terminal	sep_nct(inputfile,outputfile,number)
Shannon Entropy of Split	sep_st(inputfile,outputfile ,number)
Shannon Entropy of Residue Level	ser_wp(inputfile,outputfile)
Shannon Entropy of Residue Level for N-Terminal	ser_nt(inputfile,outputfile,number)
Shannon Entropy of Residue Level for C-Terminal	ser_ct(inputfile,outputfile,number)
Shannon Entropy of Residue Level for Rest	ser_rt(inputfile,outputfile,number1,number2)

Shannon Entropy of Residue Level for N-Terminal	ser_nct(inputfile,outputfile,number)
Shannon Entropy of Residue Level for Split	ser_st(inputfile,outputfile ,number)
Shannon Entropy of Physicochemical Property	spc_wp(inputfile,outputfile)
Shannon Entropy of Physicochemical Property for N- Terminal	spc_nt(inputfile,outputfile,number)
Shannon Entropy of Physicochemical Property for C- Terminal	spc_ct(inputfile,outputfile,number)
Shannon Entropy of Physicochemical Property for Rest	spc_rt(inputfile,outputfile,number1,number2)
Shannon Entropy of Physicochemical Property for N- and C-Terminal	spc_nct(inputfile,outputfile,number)
Shannon Entropy of Physicochemical Property for Split	spc_st(inputfile,outputfile,number)
Autocorrelation	acr_wp(inputfile,outputfile,lag)
Autocorrelation of N-Terminal	act_nt(inputfile,outputfile,number,lag)
Autocorrelation of C-Terminal	acr_ct(inputfile,outputfile,number,lag)
Autocorrelation of Rest	acr_rt(inputfile,outputfile,number1,number2,lag)
Autocorrelation of N- and C-Terminal	acr_nct(inputfile,outputfile,number,lag)
Autocorrelation of Split	acr_st(inputfile,outputfile,number,lag)
Conjoint Triad Calculation	ctc_wp(inputfile,outputfile)
Conjoint Triad Calculation for N-Terminal	ctc_nt(inputfile,outputfile,number)
Conjoint Triad Calculation for C-Terminal	ctc_ct(inputfile,outputfile,number)
Conjoint Triad Calculation for Rest	ctc_rt(inputfile,outputfile,number1,number2)
Conjoint Triad Calculation for N- and C-Terminal	ctc_nct(inputfile,outputfile,number)
Conjoint Triad Calculation for Split	ctc_st(inputfile,outputfile,number)
Composition enhanced Transition Distribution	ctd_wp(inputfile,outputfile)
Composition enhanced Transition Distribution of N-Terminal	ctd_nt(inputfile,outputfile,number)
Composition enhanced Transition Distribution of C-Terminal	ctc_ct(inputfile,outputfile,number)
Composition enhanced Transition Distribution of Rest	ctd_rt(inputfile,outputfile,number1,number2)
Composition enhanced Transition Distribution of N- and C-Terminal	ctd_nct(inputfile,outputfile,number)
Composition enhanced Transition Distribution	ctd_st(inputfile,outputfile ,number)
Pseudo Amino-acid Composition	paac_wp(inputfile,outputfile,lamda,weight)
Pseudo Amino-acid Composition of N-Terminal	paac_nt(inputfile,outputfile,number,lamda,weight)
Pseudo Amino-acid Composition of C-Terminal	paac_ct(inputfile,outputfile,number,lamda,weight)

Pseudo Amino-acid Composition of Rest	paac_rt(inputfile,outputfile,number1,number2,lamda,weight)
Pseudo Amino-acid Composition of N- and C-Terminal	paac_nct(inputfile,outputfile,number,lamda,weight)
Pseudo Amino-acid Composition of split	paac_st(inputfile,outputfile,lambda,number)
Amphiphilic Pseudo Amino-acid Composition	apaac_wp(inputfile,outputfile,lamda,weight)
Amphiphilic Pseudo Amino-acid Composition of N- Terminal	apaac_nt(inputfile,outputfile,number,lamda,weight)
Amphiphilic Pseudo Amino-acid Composition of C- Terminal	apaac_ct(inputfile,outputfile,number,lamda,weight)
Amphiphilic Pseudo Amino-acid Composition of Rest	apaac_rt(inputfile,outputfile,number1,number2,lamda,weight)
Amphiphilic Pseudo Amino-acid Composition of N- and C-Terminal	apaac_nct(inputfile,outputfile,number,lamda,weight)
Amphiphilic Pseudo Amino-acid Composition of split	apaac_st(inputfile,outputfile,number1,number2,lamda,weight)
Quasi-Sequence Order	qos_wp(inputfile,outputfile,gap,weight)
Quasi-Sequence Order of N-Terminal	qos_nt(inputfile,outputfile,number,gap,weight)
Quasi-Sequence Order of C-Terminal	qos_ct(inputfile,outputfile,number,gap,weight)
Quasi-Sequence Order of Rest	qos_rt(inputfile,outputfile,number1,number2,gap,weight)
Quasi-Sequence Order of N- and C-Terminal	qos_nct(inputfile,outputfile,number,gap,weight)
Quasi-Sequence Order of split	qos_st(inputfile,outputfile,Number_of_splits,gap,weight)
Sequence Order Coupling	soc_wp(inputfile,outputfile,gap)
Sequence Order Coupling for N-Terminal	soc_nt(inputfile,outputfile,number,gap)
Sequence Order Coupling for C-Terminal	soc_ct(inputfile,outputfile,number,gap)
Sequence Order Coupling for Rest	soc_rt(inputfile,outputfile,number1,number2,gap)
Sequence Order Coupling for N- and C-Terminal	soc_nct(inputfile,outputfile,number,gap)
Sequence Order Coupling of split	soc_split(inputfile,outputfile,Number_of_splits,gap)
Binary Profile Based Features	
Binary Profile of amino acid	aab_wp(inputfile,outputfile)
Binary Profile of N-Terminal amino acid	aab_nt(inputfile,outputfile,number)
Binary Profile of C-Terminal of amino acid	aab_ct(inputfile,outputfile,number)
Binary Profile of rest of amino acid	aab_rt(inputfile,outputfile,number1,number2)
Binary Profile of N- and C-Terminal amino acid	aab_nct(inputfile,outputfile,number)
Binary Profile of split of amino acid	aab_st(inputfile,outputfile,number)
Binary Profile of Dipeptide	dpb_wp(inputfile,outputfile,order)
Dipeptide Binary Profile of N-Terminal	dbp_nt(inputfile,outputfile,number,order)
Dipeptide Binary Profile of C-Terminal	dpb_ct(inputfile,outputfile,number,order)

Dipeptide Binary Profile of Rest	dpb_rt(inputfile,outputfile,number1,number2,order)
Dipeptide Binary Profile of N- and C-Terminal	dbp_nct(inputfile,outputfile,number,order)
Dipeptide Binary Profile of Split	dpb_st(inputfile,outputfile,lag,Number_of_splits)
Atomic Binary Profile	atb_wp(inputfile,outputfile)
Atomic Binary Profile of N-Terminal	atb_nt(inputfile,outputfile,number)
Atomic Binary Profile of C-Terminal	atb_ct(inputfile,outputfile,number)
Atomic Binary Profile of rest	atb_rt(inputfile,outputfile,number1,number2)
Atomic Binary Profile of N- and C-Terminal	atb_nct(inputfile,outputfile,number)
Atomic Binary Profile of split	atb_st(inputfile,outputfile ,Number_of_splits)
Binary Profile of bond	btb_wp(inputfile,outputfile)
Bonds Binary Profile of N-Terminal	btb_nt(inputfile,outputfile,number)
Bonds Binary Profile of C-Terminal	btb_ct(inputfile,outputfile,number)
Bonds Binary Profile of rest	btc_rt(inputfile,outputfile,number1,number2)
Bonds Binary Profile of N- and C-Terminal	btb_nct(inputfile,outputfile,number)
Bonds Binary Profile of split	btb_st(inputfile,outputfile ,number_of_splits)
Binary Profile of Physicochemical Property	pcb_wp(inputfile,outputfile)
Binary Profile of Physicochemical Property of N-Terminal	pcb_nt(inputfile,outputfile,number)
Binary Profile of Physicochemical Property of C-Terminal	pcb_ct(inputfile,outputfile,number)
Binary Profile of Physicochemical Property of Rest	pcb_rt(inputfile,outputfile,number1,number2)
Binary Profile of Physicochemical Property of N- and C-Terminal	pcb_nct(inputfile,outputfile,number)
Binary Profile of Physicochemical Property of Split	pcb_st(inputfile,outputfile,number_of_splits)
Binary Profile of AAIndex	aib_wp(inputfile,outputfile)
Binary Profile of AAIndex of N-Terminal	aib_nt(inputfile,outputfile,number)
Binary Profile of AAIndex of C-Terminal	aib_ct(inputfile,outputfile,number)
Binary Profile of AAIndex of Rest	aib_rt(inputfile,outputfile,number1,number2)
Binary Profile of AAIndex of N- and C-Terminal	aib_nct(inputfile,outputfile,number)
Binary Profile of AAIndex of split	aib_st(inputfile,outputfile,number_of_splits)
PSSM Based Features	
PSSM composition	pssm_comp(inputfile,outputfile)
Normalization of PSSM profile using method 1	pssm_n1(inputfile,outputfile)
Normalization of PSSM profile using method 2	pssm_n2(inputfile,outputfile)
Normalization of PSSM profile using method 3	pssm_n3(inputfile,outputfile)
Normalization of PSSM profile using method 4	pssm_n4(inputfile,outputfile)

Pattern Based Features	
Binary Profile of generated pattern	pat_bin(inputfile,outputfile,window_length)
Physico-chemical properties composition of generated pattern	pat_pcp(inputfile,outputfile,window_length)
Pattern generated from sequences	pat_str(inputfile,outputfile,window_length)
Pattern generated from CSV file	pat_csv(inputfile,outputfile,window_length)
Amino acid index composition of generated pattern	pat_aai(inputfile,outputfile,window_length)

2.2 Webserver Implementation

Pfeature is available as webserver at URL: https://webs.iiitd.edu.in/raghava/pfeature/ . User can go to the link and can access the GUI of Pfeature, which is majorly divided into six modules, such as:

- Composition : It calculates composition based features using sequence information.
- Binary Profiles : It calculates binary profile based features using sequence information.
- Evolutionary Information : It calculates evolutionary information based features using sequence information.
- Structure : It calculates structural features using structure.
- Pattern : It calculates features by creating patterns of desired length using sequence information.
- Model Building : This module generate classification and regression model using generated features.

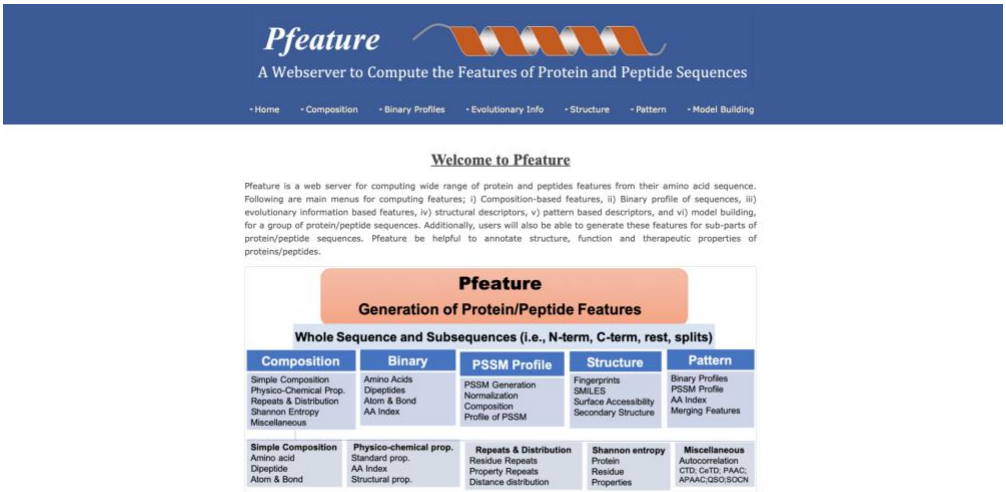


Figure 2.1: Homepage of webserver Pfeature

2.3 Usage of Standalone

We have also provided the standalone version for Pfeature, which can be used to calculate the features for big data. The standalone package of Pfeature allow users to computes individual as well as, all possible descriptors for a protein/peptide sequence. This document provide information about standalone version of Pfeature. The python based standalone package of Pfeature can be downloaded from https://github.com/raghavagps/Pfeature/tree/master/Standalone or https://webs.iiitd.edu.in/raghava/pfeature/pfeature_standalone.zip. This standalone contains three scripts, their description is as follows:

- pfeature_comp.py : To calculated composition based features
- pfeature_bin.py : To calculated binary profile based features
- pfeature_pssm.py : To calculated binary profile based features

In order to know the full usage of these scripts, user can add "-h or --help" as follows:

i) python3 pfeature_comp.py -h

```
usage: pfeature_comp.py [-h] -i INPUT [-o OUTPUT]
                        [-j {AAC,DPC,TPC,ATC,BTC,PCP,AAI,RRI,PRI,DDR,SEP,SER,SPC,ACR,CTC,CeTD,PAAC,APAAC,QSO,SOC,ALLCOMP}]
                        [-n N_TERMINAL] [-c C_TERMINAL] [-nct NC_TERMINAL]
                        [-rn REST_N] [-rc REST_C] [-rnc REST_NC] [-s SPLIT]
                        [-d LAG] [-w WEIGHT] [-t PWEIGHT]

Please provide following arguments

optional arguments:
  -h, --help            show this help message and exit
  -i INPUT, -I INPUT, --input INPUT
                        Input: protein or peptide sequence in FASTA format or single sequence per line in single letter code
  -o OUTPUT, -O OUTPUT, --output OUTPUT
                        Output: File for saving results by default pfeature_result.csv
  -j {AAC,DPC,TPC,ATC,BTC,PCP,AAI,RRI,PRI,DDR,SEP,SER,SPC,ACR,CTC,CeTD,PAAC,APAAC,QSO,SOC,ALLCOMP}, -J {AAC,DPC,TPC,ATC,BTC,PCP,AAI,RRI,PRI,DDR,SEP,SER,SPC,ACR,CTC,CeTD,PAAC,APAAC,QSO,SOC,ALLCOMP}, --job {AAC,DPC,TPC,ATC,BTC,PCP,AAI,RRI,PRI,DDR,SEP,SER,SPC,ACR,CTC,CeTD,PA
AC,APAAC,QSO,SOC,ALLCOMP}
                        Job Type:
                        AAC: Amino acid composition
                        DPC: Dipeptide composition
                        TPC: Tripeptide composition
                        ATC: Atomic composition
                        BTC: Bond composition
                        PCP: Physico-chemical properties composition
                        AAI: Amino-acid indices composition
                        RRI: Residue repeat information
                        PRI: Physico-chemical properties repeat information
                        DDR: Distance distribution of residues
                        SEP: Shannon entropy of protein
                        SER: Shannon entropy of residues
                        SPC: Shannon entropy of physico-chemical properties
                        ACR: Autocorrelation descriptors
                        CTC: Conjoint triad descriptors
                        CeTD: Composition enhanced transition distribution
                        PAAC: Pseudo amino acid composition
                        APAAC: Amphiphilic pseudo amino acid composition
                        QSO: Quasi sequence order
                        SOC: Sequence order coupling number
                        ALLCOMP:All composition features together except ACR and AAI
                        by default AAC
  -n N_TERMINAL, -N N_TERMINAL, --n_terminal N_TERMINAL
                        Window Length from N-terminal: by default 0
  -c C_TERMINAL, -C C_TERMINAL, --c_terminal C_TERMINAL
                        Window Length from C-terminal: by default 0
  -nct NC_TERMINAL, -Nct NC_TERMINAL, -NCt NC_TERMINAL, -NCT NC_TERMINAL, -nCt NC_TERMINAL, -ncT NC_TERMINAL, -nCT NC_TERMINAL, -NcT NC_TERMINAL, --nc_terminal NC_TERMINAL
                        Residues from N- and C-terminal: by default 0
  -rn REST_N, -Rn REST_N, -RN REST_N, -rN REST_N, --rest_n REST_N
                        Number of residues removed from N-terminal, by default 0
  -rc REST_C, -Rc REST_C, -RC REST_C, -rC REST_C, --rest_c REST_C
                        Number of residues removed from C-terminal, by default 0
  -rnc REST_NC, -RNC REST_NC, -Rnc REST_NC, -rNc REST_NC, -rnC REST_NC, -RNc REST_NC, -rNC REST_NC, -RnC REST_NC, --rest_nc REST_NC
                        Number of residues removed from N- and C-terminal, by default 0
  -s SPLIT, -S SPLIT, --split SPLIT
                        Number of splits a sequence divided into, by default 0
  -d LAG, -D LAG, --lag LAG
                        This represents the order of gap, lag or dipeptide, by default 1
  -w WEIGHT, -W WEIGHT, --weight WEIGHT
                        Weighting Factor for QSO: Value between 0 to 1, by default 0.1
  -t PWEIGHT, -T PWEIGHT, --pweight PWEIGHT
                        Weighting factor for pseudo and amphiphlic pseudo amino acid composition: Value between 0 to 1, by default 0.05
```

Figure 2.2: Complete usage of script pfeature_comp.py

ii) python3 pfeature_bin.py -h

```
usage: pfeature_bin.py [-h] -i INPUT [-o OUTPUT]
                       [-j {AAB,DPB,ATB,BTB,PCB,AIB,ALLBIN}] [-n N_TERMINAL]
                       [-c C_TERMINAL] [-nct NC_TERMINAL] [-rn REST_N]
                       [-rc REST_C] [-rnc REST_NC] [-s SPLIT] [-d LAG]

Please provide following arguments

optional arguments:
  -h, --help            show this help message and exit
  -i INPUT, -I INPUT, --input INPUT
                        Input: protein or peptide sequence in FASTA format or single sequence per line in single letter code
  -o OUTPUT, -O OUTPUT, --output OUTPUT
                        Output: File for saving results by default pfeature_result.csv
  -j {AAB,DPB,ATB,BTB,PCB,AIB,ALLBIN}, -J {AAB,DPB,ATB,BTB,PCB,AIB,ALLBIN}, --job {AAB,DPB,ATB,BTB,PCB,AIB,ALLBIN}
                        Job Type:
                        AAB: Amino acid based binary profile
                        DPB: Dipeptide based binary profile
                        ATB: Atom based binary profile
                        BTB: Bond based binary profile
                        PCB: Physico-chemical properties based binary profile
                        AIB: Amino-acid indices based binary profile
                        ALLBIN:All binary profiles together except ATB and BTB
                        by default AAB
  -n N_TERMINAL, -N N_TERMINAL, --n_terminal N_TERMINAL
                        Window Length from N-terminal: by default 0
  -c C_TERMINAL, -C C_TERMINAL, --c_terminal C_TERMINAL
                        Window Length from C-terminal: by default 0
  -nct NC_TERMINAL, -Nct NC_TERMINAL, -NCt NC_TERMINAL, -NCT NC_TERMINAL, -nCt NC_TERMINAL, -ncT NC_TERMINAL, -nCT NC_TERMINAL, -NcT NC_TERMINAL, --nc_terminal NC_TERMINAL
                        Residues from N- and C-terminal: by default 0
  -rn REST_N, -Rn REST_N, -RN REST_N, -rN REST_N, --rest_n REST_N
                        Number of residues removed from N-terminal, by default 0
  -rc REST_C, -Rc REST_C, -RC REST_C, -rC REST_C, --rest_c REST_C
                        Number of residues removed from C-terminal, by default 0
  -rnc REST_NC, -RNC REST_NC, -Rnc REST_NC, -rNc REST_NC, -rnC REST_NC, -RNc REST_NC, -rNC REST_NC, -RnC REST_NC, --rest_nc REST_NC
                        Number of residues removed from N- and C-terminal, by default 0
  -s SPLIT, -S SPLIT, --split SPLIT
                        Number of splits a sequence divided into, by default 0
  -d LAG, -D LAG, --lag LAG
                        This represents the order of gap, lag or dipeptide, by default 1
```

Figure 2.3: Complete usage of script pfeature_bin.py

iii) python3 pfeature_pssm.py -h

```
usage: pfeature_pssm.py [-h] -i INPUT [-o OUTPUT] [-n {N0,N1,N2,N3,N4}]

Please provide following arguments

optional arguments:
  -h, --help            show this help message and exit
  -i INPUT, -I INPUT, --input INPUT
                        Input: protein or peptide sequence in FASTA format or single sequence per line in single letter code
  -o OUTPUT, -O OUTPUT, --output OUTPUT
                        Output: File for saving results by default pssm_profile.csv
  -n {N0,N1,N2,N3,N4}, -N {N0,N1,N2,N3,N4}, --normalization_method {N0,N1,N2,N3,N4}
                        Normalization Method:
                        N0: It provides pssm profile without any normalization
                        N1: It normalizes pssm profile based on 1/(1+e^-x) formula
                        N2: It normalizes pssm profile based on (x-min)/(max-min) formula
                        N3: It normalizes pssm profile based on ((x-min)/(max-min))*100 formula
                        N4: It normalizes pssm profile based on 1/(1+e^-(x/100) formula
                        By default it is N0
```

Figure 2.4: Complete usage of script pfeature_pssm.py

Minimum Usage of Standalone scripts

In the standalone scripts such as pfeature_comp.py, pfeature_bin.py, and pfeature_pssm.py, the user can run the python script only by providing the input file in the fasta or single line format as shown in the following figure. The information about the other parameters can be accessed using "-h" parameter.

```
python3 pfeature_comp.py -i protein.fa
OR
python3 pfeature_bin.py -i protein.fa
OR
python3 pfeature_pssm.py -i protein.fa
```

Figure 2.5: Minimum usage for scripts

Chapter 3.0
Composition Based Features

In this section, we have described python functions developed for amino acid composition based feature generation. These modules can be used for feature generation for protein sequences.

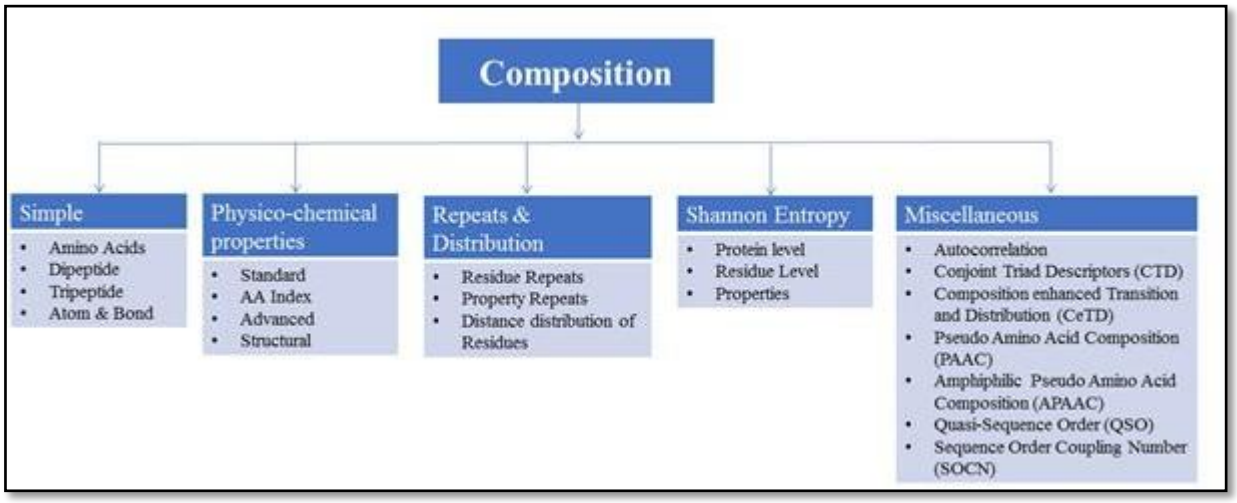


Figure 3.1: This flowchart shows different menus/submenus for computing different type of composition-based features of protein/peptide composition.

3.1 Simple

This sub-module comes under composition module, which is able to generate simple composition-based feature from protein/peptide sequences provided in either single line format or in FASTA format. This sub-module is called as simple composition because of the kind of features. It comprises of amino acid composition (20 features), dipeptide composition (400 features), tripeptide composition (8000 features), and atom & bond composition (9 features). Pfeature web site provides dynamic webpage to compute these features, moreover we have also provided the python library and standalone, as described in Chapter 2.

3.1.1 Amino Acids

This is a simplest feature, which is heavily used in literature for predicting function or structure of a protein. It computes the amino acid composition of each type of residue in a protein sequence. The compositions of all 20 natural amino acids were calculated using the following equation 1:

$$AAC_i = \frac{R_i}{L} \quad (1)$$

where $\boldsymbol{AAC_i}$ is amino acid composition of residue type $\boldsymbol{i}$; $\boldsymbol{R_i}$ and $\boldsymbol{L}$ number of residues of type $\boldsymbol{i}$ and length of sequence, respectively.

In order to compute the amino acid composition of different portions of an amino acid sequence, we have developed several python functions as described in **Table 2.1**; the user can call these functions in standalone or library. In the web server, users may select a part of the sequence for calculating protein features such as N-terminal, C-terminal, Splits, Rest.

3.1.2 Dipeptide and higher order dipeptides

Amino acid composition provides only number of different type of residues, but lacks information regarding the order of residues. Dipeptide composition is used to encapsulate the global information about each sequence, which gives a fixed vector of length 400 (20*20). This representation encompassed the information about amino acid composition along with the local order of amino acid. Traditionally, a dipeptide is made of consecutive residues (residue ***i*** and ***i+1***). In 2005, dipeptide of higher order were introduced (**J Biol Chem. 2005; 280:14427-32**). In case of higher order dipeptides, a dipeptide is made of ***i*** and ***i+2*** or ***i+3*** or ***i+4*** etc., instead of consecutive residues which is represented as dipeptide with order 1(See Figure 3.2, adapted from **J Biol Chem. 2005; 280:14427-32**).

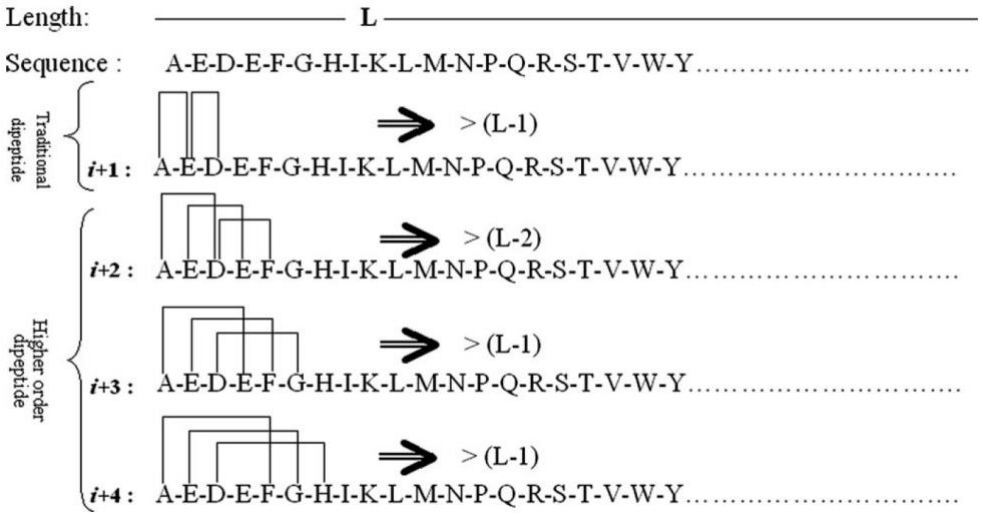


Figure 3.2: Graphical representation of traditional peptides and higher order dipeptides, figure is adapted from **J Biol Chem. 2005; 280:14427-32.**

In order to compute traditional dipeptide composition from a protein sequence following equation is used,

$$DPC_i^j = \frac{D_i^j}{L - j} \quad (2)$$

Where DPC^j_i is the fraction or composition of dipeptide of type ***i*** for jth order. D^j_i and ***L*** are the number of dipeptides of type ***i*** and length of a protein sequence, respectively. Here higher order dipeptide D^j_i is made of residue R_i and R_{i+j} where value of ***j*** is 2 or more. In case ***j*** is equal to 1 then dipeptide is called traditional dipeptide.

We have developed number of python functions to compute traditional and higher order dipeptide composition for different portions of an amino acid sequence. Their usage is provided in **Table 2.1** . In web server, user may select the portion of sequence for calculating protein features.

3.1.3 Tripeptide

Three consecutive amino acids form a tripeptide which provide local order in addition to simple composition. Both previous and next residues are used to form a tripeptide. There are total 8000 (20*20*20) possible tripeptides from 20 type of natural amino acid residue.

$$TPC_i = \frac{T_i}{L-2} \quad (3)$$

Where, TPC_i is tripeptide composition of tripeptide ***i***, out of possible 8000 tripeptides. T_i and ***L*** are number of tripeptides of type ***i*** and length of a protein sequence, respectively. In order to compute tripeptide composition in a sequence, following functions has been developed.

We have developed the python script for computing tripeptide composition for different portions of an amino acid sequence by implementing equation 3, which is available as python script, and library, whose usage is given in **Table 2.1**. In web server, the provision of selecting the specific portion of a sequence is not given, due to its computational complexity.

3.1.4 Atom & Bond

All amino acids are made up of atoms and bonds linking them. In this sub-module, we compute different type atom and bond composition. Atomic composition is fraction of Carbon, Hydrogen, Nitrogen, Oxygen and Sulphur atoms present in a protein sequence. For bond composition four types of bonds are considered total number of bonds (including aromatic), hydrogen bond, single bond and double bond. The number of each kind of bond is provided as bonds.csv file.

$$ATC_i = \frac{A_i}{N} \quad (4)$$

$$BTC_i = \frac{B_i}{N} \quad (5)$$

Where, ***ATC$_i$*** is the atomic composition of atom of type ***i***, ***A$_i$*** and ***N*** are number of atoms of type ***i*** and number of atoms in a protein sequence, respectively. Where, ***BTC$_i$*** is bond composition for bond of type ***i***, ***B$_i$*** and ***N*** are number of atoms of type ***i*** and number of atoms in a protein, respectively.

We have provided several python functions to compute atomic and bond composition for different portions of an amino acid sequence, and their usage is given in **Table 2.1** . In web server, user may select the portion of sequence for calculating protein features.

Table 3.1: List of Atoms and Bonds included in ATC & BTC Pfeature programs.

Atomic Composition	Bond Composition
Carbon Atom	Total Bonds
Hydrogen Atom	Hydrogen Bond
Nitrogen Atom	Single Bond
Oxygen Atom	Double Bond
Sulphur Atom	

3.2Physico-Chemical properties

We have used the Physico-chemical properties to represent a protein sequence. The values of each Physico-chemical property for all 20 amino acids were normalized between 0 and 1 using the standard conversion formula. The input vector has scalar values, each representing the average value of a distinct Physico-chemical property of residues (**Nucleic Acids Res. 2004; 32:W414-9**).

The sub-module is further divided into four categories as “Standard”, “AA Index”, “Advanced”, and “Structural”. Further, the provision to select the properties and portion of a sequence, is also given to the users. Each category is further explained below.

3.2.1 Standard Physico-chemical Properties

This sub-module calculates the fraction of each standard Physico-chemical property in given sequences. Following properties have been incorporated in Pfeature for calculating compositional features based on standard Physico-chemical properties.

Table 3.2: List of Physico-chemical properties included in Pfeature for computing features

Positively Charged	Aromaticity	Hydroxylic
Negatively Charged	Acidity	Sulphur Content
Neutral Charge	Basicity	Tiny
Polarity in residues	Neutral (pH)	Small
Non-polarity in residues	Hydrophobicity	Large
Aliphaticity	Hydrophilicity	
Cyclicity	Neutral towards water	

We used the following formula to calculate these features

$$PCP_i = \frac{P_i}{L} \quad (6)$$

Where, $\boldsymbol{PCP_i}$ is Physico-chemical properties composition of type $\boldsymbol{i}$; $\boldsymbol{P_i}$ and $\boldsymbol{L}$ are sum of property of type $\boldsymbol{i}$, and length of sequence, respectively.

In order to compute the composition of standard properties for different portions of an amino acid sequence, we have developed number of python functions provided as python scripts, library and standalone. **Table 2.1** summarizes the usage of each function. In web server user may select the portion of sequence and type of properties for calculating the features.

3.2.2 Amino Acid Index (AAIndex)

Amino Acid Index (AAIndex) is a database of amino acid indices, where AAIndex is a set of 20 numerical values representing various Physico-chemical and biochemical properties of amino acid residues. Current version 9.0 of database have total 566 AA indices (https://www.genome.jp/dbget/AAindex/list_of_indices). Pfeature allows user to compute composition of selected AA index via web interface or python function, using following equation:

$$AAIC_i = \frac{AAI_i}{L} \quad (7)$$

Where, $AAIC_i$ is AA index composition of residue type ***i***; AAI_i and ***L*** are sum of AA index value of type ***i*** and length of sequence, respectively.

In order to compute composition of AA index values for different portions of an amino acid sequence, we have developed number of python function and their usage is explained in **Table 2.1** . In web server user may select portion of sequence and type of properties for calculating the respective features.

3.2.3 Advanced properties

This module allows to compute composition of advanced properties like z1, z2, z3, z4 and z5 of a protein sequence. This "**Advanced**" module is similar to "**Standard**" module of computing the Physico-chemical properties.

In order to compute the composition of these advanced properties, for different portions of an amino acid sequence, we have developed python function as described in **Table 2.1** . In web server user may select portion of sequence and type of properties for calculating the respective features.

3.2.4 Structural Properties

This module allows to compute composition of advanced properties like secondary structure and surface accessibility of a protein sequence. This "**Structural**" module is similar to "**Standard**" module of computing Physico-chemical properties.

In order to compute the composition of these structural properties, for different portions of an amino acid sequence, we have developed python function as described in **Table 2.1** . In web server user may select portion of sequence and type of properties for calculating the respective features.

3.3Repeats & Distribution

Most of the existing composition modules describes the above mentioned measures or fraction of particular type of residue or residue property. One of the problems with existing features is that they do not measure the repetitive information of particular type of residue or their distribution. In this module, we have introduced the new features, which compute repeats and distribution of amino acids.

3.3.1 Residue Repeats

This sub-module is able to calculate the Repetitive Residue Information (RRI) for a peptide/protein sequence. RRI measures number of continuous runs of a residue type in a sequence, it can be calculated using following formula.

$$RRI_i = \frac{\sum_{j=1}^{N}(R_j)^2}{\sum_{j=1}^{N} R_j} \quad (8)$$

where RRI_i , N and R_j are residue repeat information, maximum number of occurrence, and number of runs/repeats in occurrence j respectively for residue type i, respectively.

Example: If a residue is a residue type occurs once at time then value of RRI will be one. For example amino acid alanine ***A*** occurs four times in following sequence "G**A**R**A**GRG**A**RDE**A**RT**A**G"; each time single run. It means ***N*** will be 5, RRI for A can be calculated using following formula

$$RRI_A = \frac{(1)^2+(1)^2+(1)^2+(1)^2+(1)^2}{1+1+1+1+1} = \frac{5}{5} = 1$$

In following sequence "G**AA**RGRG**AAA**RDERTG" amino acid ***A*** occurs two times, first time two runs and second time three runs. It means $N=2$, $R_1=2$ and $R_2=3$, ***RRI*** for ***A*** can be calculated using following equation

$$RRI_A = \frac{(2)^2+(3)^2}{2+3} = \frac{4+9}{5} = 2.6$$

In following sequence "GRGRGAA**AAA**RDERTG" amino acid ***A*** occurs once with 5 runs. It means $N=1$, and $R_1=5$; ***RRI*** for ***A*** can be calculated using following equation

$$RRI_A = \frac{(5)^2}{5} = \frac{25}{5} = 5$$

This means for a given residue type, minimum RRI will be 1 and maximum will be total number of that type of residues in sequence. This value measures multiple runs of a residue in a sequence.

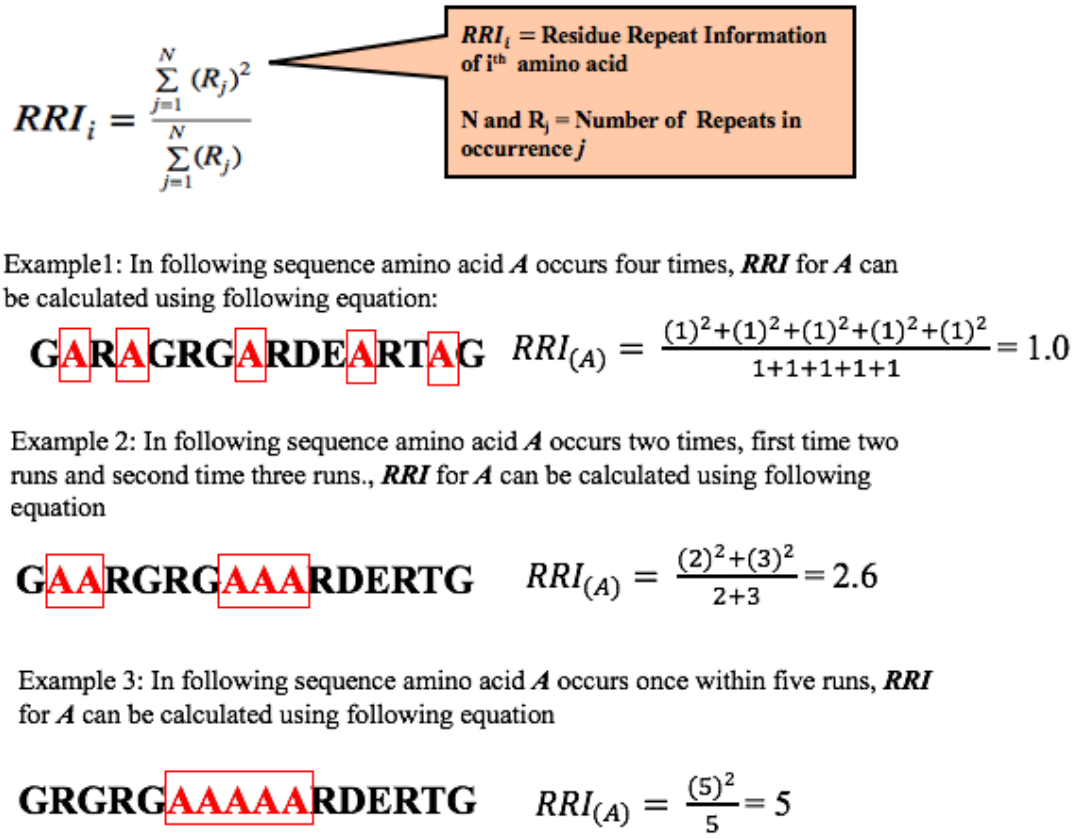


Figure 3.3: Calculation of Repetitive Residue Information (RRI) for a peptide/protein sequence.

In order to compute repetitive residue information of different portions of an amino acid sequence, we have developed number of python function, as described in **Table 2.1**. In web server user may select portion of sequence for calculating the features.

3.3.2 Property Repeats

This function calculates property repeat information (PRI) which gives the information of repetitiveness of each physicochemical property within a peptide/ protein sequence.

$$PRI_i = \frac{\sum_{j=1}^{N} (P_j)^2}{\sum_{j=1}^{N} P_j} \quad (9)$$

where PRI_i , N and P_j are property repeat information, maximum number of occurrence and number of runs/repeats in occurrence j respectively for property type i.

In order to compute the Physico-chemical property repeat information of different portions of an amino acid sequence such as N-terminal, C-terminal, Rest and Splits, we have developed number of python function, as described in **Table 2.1**. In web server user may select portion of sequence for calculating the features.

3.3.3 Distance distribution of Residues

This sub-module entitled as distance distribution of residues (DDOR) computes the distribution of residue on the basis of the distance from N-terminal, C-terminal and inter-distances between same residue within the given peptide/protein sequence.

$$DDOR_i = \frac{(R_{NT})^2 + \sum_{j=1}^{N}(R_j)^2 + (R_{CT})^2}{(L - F_i) + 1} \quad (10)$$

Where, $\boldsymbol{DDOR_i}$ is distance distribution of residue type ***i***, ***N*** is total number of inter-residue distances for type ***i***.

$\boldsymbol{R_{NT}}$= Residue distance from N-terminal

R_j= Inter-distance between residue type ***i***

R_{CT}= Residue distance from C-terminal

L= Total length of protein sequence

F_i= Frequency of residue type ***i***

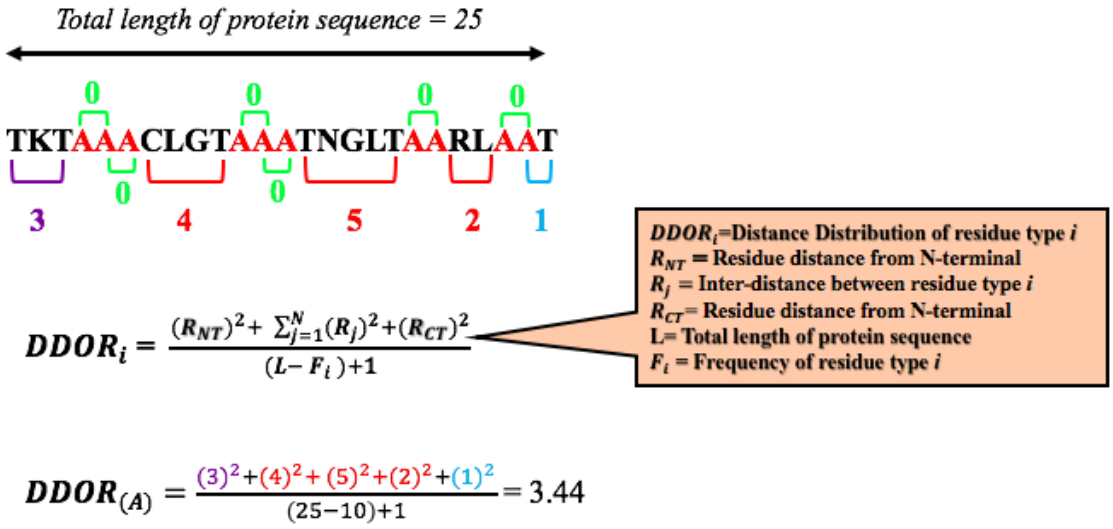


Figure 3.4: Calculation of Distance Distribution of Residue (DDOR) for peptide/protein sequence.

3.4Shannon Entropy

3.4.1 Protein Level

Shannon entropy for a protein/peptide sequence can be computed by the standard expression:

$$H(X) = -\sum_{i=1}^{20} pi \log_2 p_i \qquad (11)$$

Where, i is the amino acid in the sequence (i=A, C, D, … ,Y) and X is any protein/peptide sequence. See figure below for more details.

We have provided the user with several python functions in standalone and python library, to compute the Shannon entropy at protein level for different portions of the amino acid sequence, and their usage is given in **Table 2.1** . In web server, user may select the portion of sequence for calculating protein features.

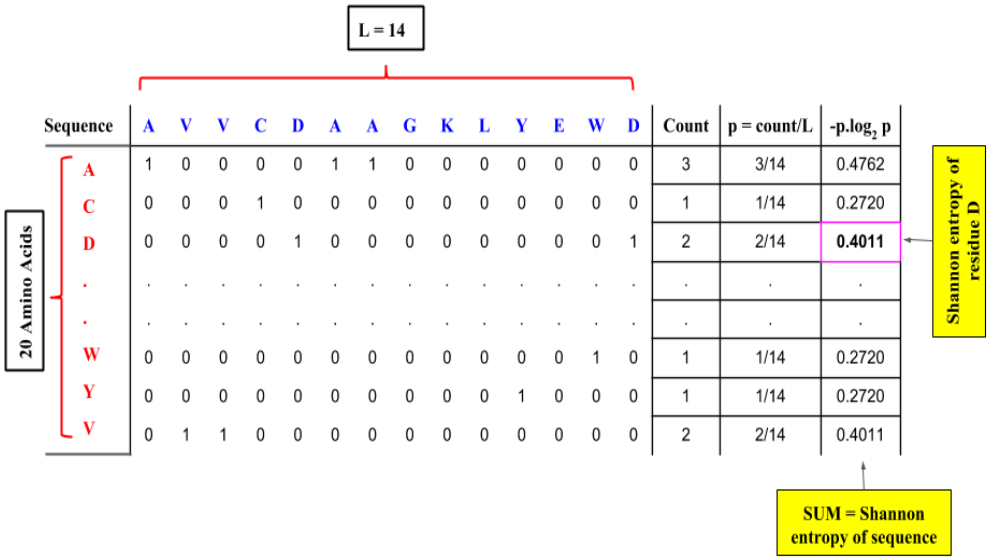


Figure 3.5: Calculation of Shannon entropy for a protein/ peptide sequence or sub-sequence at protein and residue levels.

3.4.2 Residue Level

As shown in the Figure 3.5, Shannon entropy of 20 natural amino acid residues are calculated using the following two equations:

$$p_i = \frac{C_i}{L} \qquad (12)$$

$$H_i = -p_i \log_2 p_i \qquad (13)$$

Where, C_i is the count of amino acid of type i in the sequence, L is the total length of sequence, and H_i is entropy of residue i.

In order to compute the Shannon entropy at residue level for different portions of an amino acid sequence, we have developed python scripts which is implemented in library and standalone, their usage is given in **Table 2.1**. In the web server, users may select a part of the sequence for calculating protein features such as N-terminal, C-terminal, Splits, Rest.

3.4.3 Properties Level

This function calculates the Shannon Entropy of a particular Physicochemical property in a sequence. Let the sequence be of length 'l' and has r_i instances of a property present in the sequence, then the Shannon Entropy $H_i(x)$ of a particular physicochemical property is calculated using the following formula:

$$H_i = -p_i \log(p_i) - (1 - p_i)\log(1 - p_i) \qquad (13)$$

where p_i is r_i/l.

We have provided the user with several python functions in standalone and python library, to compute the Shannon entropy at property level for different portions of the amino acid sequence, and their usage is given in **Table 2.1** . In web server, user may select the portion of sequence and can check the desired properties, for calculating the features.

3.5Miscellaneous

3.5.1 Autocorrelation

Autocorrelation descriptors are defined based on the distribution of amino acid properties along the sequence. The amino acid properties used here are various types of amino acid indices (http://www.genome.ad.jp/dbget/aaindex.html). Three type of autocorrelation descriptors are used here viz. Normalized Moreau-Broto, Moran and Geary autocorrelation descriptors as implemented in (**Dong, Jie, et al. *Journal of cheminformatics* 10.1 (2018): 16.**)

Conditions: dval<=min(L-1, 30) where L is the length of the sequence/ sub-sequence for which autocorrelation descriptors have to be calculated.

In the server as well as in library and standalone, we have defined several functions which is capable of run the same operation on different portion of the sequence, their calling is described in **Table 2.1.** In the webserver, user can provide the amino acid indices, dval, and portion of the sequence.

3.5.2 Conjoint Triad Descriptors (CTD)

Conjoint triad descriptors are proposed by J.W. Shen et.al. These descriptors explains the features of protein pairs based on the classification of amino acids. The 20 amino acids were clustered into several classes according to their dipoles and volumes of the side chains in the following manner (**Dong, Jie, et al. *Journal of cheminformatics* 10.1 (2018): 16.)**

Group 1: A, G, V
Group 2: I, L, F, P
Group 3: Y, M, T, S
Group 4: H, N, Q, W
Group 5: R, K
Group 6: D, E
Group 7: C

The conjoint triad descriptors considers the property of amino acid along with its adjacent amino acids as one single unit of three amino acids. Triad of three amino acids belonging to same group are identical in nature, such as RCE and KCD are identical in nature. Protein sequence can be represented as a binary space (V, F) where, V is the vector space of the sequence features, and each feature v_i represents a triad type; F is the frequency vector corresponding to V, and f_i is the frequency of type v_i appearing in the protein sequence. For the amino acids that have been catalogued into seven classes, the size of V should be 7×7×7; thus i = 1,2, ..., 343. Long protein would have a large value of f_i as compared to small sequences thus creating problem while comparing two heterogeneous proteins. Thus, we will normalize f_i in following manner:-

$$f\_norm_i = (f_i - \min(f_1, f_2, f_3 \ldots\ldots f_{343})) / \max(f_1, f_2, f_3 \ldots\ldots f_{343})$$

Table 2.1 explains the functions developed to calculate the conjoint triad descriptors for different portions of the sequence. The same facility is also provided in the webserver.

3.5.3 Composition enhanced Transition and Distribution (CeTD)

First step is to encode (convert) the peptide/protein sequence on the basis of their group value. All the values are present in aa_aatr_group.csv file. Then occurrence (composition) of each residue within is calculated using formula:

$$Composition = \frac{Frequency\ of\ same\ Residue * 100}{Length\ of\ peptide\ sequence} \qquad (14)$$

Table 3.9: **Distribution of residues in three groups with respect to attributes**

Attribute	Group 1	Group 2	Group 3
Hydrophobicity	R,K,E,D,Q,N	G,A,S,T,P,H,Y	C,L,V,I,M,F,W
Normalized Vander Waals volume	G,A,S,T,P,D	N,V,E,Q,I,L	M,H,K,F,R,Y,W
Polarity	L,I,F,W,C,M,V,Y	P,A,T,G,S	H,Q,R,K,N,E,D
Polarizability	G,A,S,D,T	C,P,N,V,E,Q,I,L	K,M,H,F,R,Y,W
Charge	K,R	A,N,C,Q,G,H,I,L,M,F,P,S,T,W,Y,V	D,E
Secondary structure	E,A,L,M,Q,K,R,H	V,I,Y,C,W,F,T	G,N,P,S,D
Solvent accessibility	A,L,F,C,G,I,V,W	R,K,Q,E,N,D	M,S,P,T,H,Y

There are 9- possibilities that two residues lying next to each other. This is called enhanced transition (E-Transition). 11,12,13,21,22,23,31,32,33 are the 9 possibilities.

Distribution is the measure of presence of particular residue in 5 quartile (0%, 25%, 50%, 75%, 100%) of the peptide sequence.

The facility to calculate the Composition enhanced Transition and Distribution (CeTD) for the portions of the sequence is provided in the library, standalone and webservers. The usage of these functions can be seen from **Table 2.1.**

3.5.4 Pseudo Amino Acid Composition (PAAC)

This group of descriptors has been proposed by K.C. Chou. Let $H_1^o(i)$ be hydrophobicity values for i = 1,2,3,.....20, $H_2^o(i)$ be the hydrophilicity values for i = 1,2,3,.....20, and $M^o(i)$ be the side chain masses of the 20 natural amino acids. They are converted to the following quantities by a standard conversion:

$$\boldsymbol{H_1(i)} = \frac{\boldsymbol{H_1^o(i) - \frac{1}{20}\sum_{i=1}^{20} H_1^o}}{\sqrt{\frac{\sum_{i=1}^{20}[H_1^o(i) - \frac{1}{20}\sum_{i=1}^{20} H_1^o(i)]^2}{20}}} \qquad (15)$$

Where, $\boldsymbol{H_1^o}$ (i) and M^o(i) are normalized asH_1(i) and M(i) in the same manner.

The user can calculate the pseudo amino acid composition for different portions of the input sequence, by using library and standalone as described in **Table 2.1.** Similarly, webserver of Pfeature also provide the facility to run the same operation on different portions of the sequence.

3.5.5 Amphiphilic Pseudo Amino Acid Composition (APAAC)

Amphiphilic Pseudo-Amino Acid Composition (APAAC) is described as:

$$\boldsymbol{P_c} = \frac{\boldsymbol{f_c}}{\sum_{\boldsymbol{r=1}}^{\boldsymbol{20}} \boldsymbol{f_r} + \boldsymbol{w}\sum_{\boldsymbol{j=1}}^{\boldsymbol{2\lambda}} \boldsymbol{\tau_j}} \quad (1 < c < 20) \qquad (16(i))$$

$$\boldsymbol{P_c} = \frac{\boldsymbol{\omega\tau_u}}{\sum_{\boldsymbol{r=1}}^{\boldsymbol{20}} \boldsymbol{f_r} + \boldsymbol{w}\sum_{\boldsymbol{j=1}}^{\boldsymbol{2\lambda}} \boldsymbol{\tau_j}} \quad (21 < u < 20+2\lambda) \qquad (16(ii))$$

where w is the weighting factor which is set as (w= 0.05),as described in Chou's work (Chou, 2001).

Amphiphilic pseudo amino acid composition can be calculated for different portions of amino acid sequence, using library, standalone and webserver of Pfeature.

3.5.6 Quasi-Sequence Order (QSO)

The quasi-sequence-order descriptors are proposed by K.C. Chou, et.al. Quasi-sequence-order Descriptors obtained from the distance matrix between the 20 amino acids. Schneider-Wrede physicochemical distance matrix (Schneider and Wrede, 1994) and the chemical distance matrix by Grantham (Grantham, 1974) are used by Kuo-Chen Chou.

For each type of amino acid, a quasi-sequence-order descriptor can be described as:

$$X_r = \frac{f_r}{\sum_{r=1}^{20} f_r + w\sum_{d=1}^{nlag} \tau_d} \quad r = 1,2,....20 \qquad (17)$$

where fr is the normalized occurrence of amino acid type r,and w is a weighting factor (w = 0.1), $nlag$ and $\tau_d$is the same which was described above. These are the first 20 quasi-sequence-order descriptors. The other 30 quasi-sequence-order descriptors are defined as:

$$X_d = \frac{w\tau_d - 20}{\sum_{r=1}^{20} f_r + w\sum_{d=1}^{nlag} \tau_d} \quad d = 21,22,....30 + nlag \qquad (18)$$

Table 2.1 exhibits the function calling to calculate the quasi-sequence-order descriptors for different portions of input sequence.

3.5.7 Sequence Order Coupling Number (SOC)

The *d*-th rank sequence-order-coupling number is described as:

$$\tau_d = \sum_{i=1}^{N-d} (d_{i,i+d})^2 \quad d = 1,2,3,.....,nlag \qquad (19)$$

where *d i, i+d* is the number in a given distance matrix explaining a distance between the two amino acids *i* and *i+d*, nlag is the maximum value of the lag, and N denotes the length of a protein or peptide sequence.

In order to compute the sequence order coupling number for different portions of amino acid sequence, we have developed python scripts which is implemented in library and standalone, their usage is given in **Table 2.1**. In the web server, users may select a part of the sequence for calculating protein features such as N-terminal, C-terminal, Splits, Rest.

Note: The length of the protein or peptide sequence must be not less than the maximum value of *nlag*.

Chapter 4.0
Binary Profiles Based Features

In this section, we have described the python functionalities developed for amino acid binary profile-based feature generation. These modules can be used for feature generation for protein sequences to apply machine learning techniques for further analysis, for instances, to explore the position specificity in interaction studies.

4.1 Amino Acids

This function generates binary equivalent of each residues. The following table consists of 20-vector binary profile for each residue. Peptide/protein sequences are replaced by their equivalent binary profile.

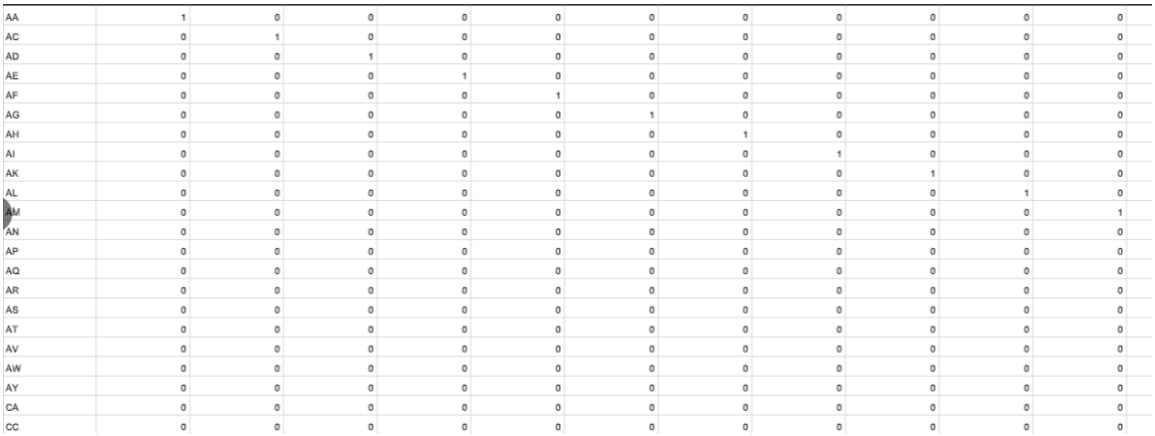
```
A : 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
C : 0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
D : 0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
E : 0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
F : 0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
G : 0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
H : 0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
I : 0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0
K : 0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0
L : 0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0
M : 0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0
N : 0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0
P : 0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
Q : 0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0
R : 0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0
S : 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0
T : 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0
V : 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0
W : 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0
Y : 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
X : 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Figure 4.1: Representation of residues in binary profile

The generation of binary profile for sequence with length L will result into the vector size of 20*L. Binary profile for different portions of the sequence can be calculated using standalone, library or webserver.

4.2Dipeptides

The Dipeptide binary profiles are generated by this function by replacing residues by their equivalent 400-size vector. The snapshot is as below.



AA	1	0	0	0	0	0	0	0	0	0	0
AC	0	1	0	0	0	0	0	0	0	0	0
AD	0	0	1	0	0	0	0	0	0	0	0
AE	0	0	0	1	0	0	0	0	0	0	0
AF	0	0	0	0	1	0	0	0	0	0	0
AG	0	0	0	0	0	1	0	0	0	0	0
AH	0	0	0	0	0	0	1	0	0	0	0
AI	0	0	0	0	0	0	0	1	0	0	0
AK	0	0	0	0	0	0	0	0	1	0	0
AL	0	0	0	0	0	0	0	0	0	1	0
AM	0	0	0	0	0	0	0	0	0	0	1
AN	0	0	0	0	0	0	0	0	0	0	0
AP	0	0	0	0	0	0	0	0	0	0	0
AQ	0	0	0	0	0	0	0	0	0	0	0
AR	0	0	0	0	0	0	0	0	0	0	0
AS	0	0	0	0	0	0	0	0	0	0	0
AT	0	0	0	0	0	0	0	0	0	0	0
AV	0	0	0	0	0	0	0	0	0	0	0
AW	0	0	0	0	0	0	0	0	0	0	0
AY	0	0	0	0	0	0	0	0	0	0	0
CA	0	0	0	0	0	0	0	0	0	0	0
CC	0	0	0	0	0	0	0	0	0	0	0

Figure 4.2: Binary profiles for few possible dipeptides

Here gap is also taken in account. If no gap is present then 0 value should pass by user and otherwise needed gap should be entered.

The generation of dipeptide binary profile for sequence with length L will result into the vector size of 400*(L-1). Binary profile for different portions of the sequence can be calculated using standalone, library or webserver.

4.3Atom & Bond

This function computes the binary profile corresponding to atomic and bond composition of each amino acid residue of the peptide sequence. Atomic composition is percentage of Carbon, Hydrogen, Nitrogen, Oxygen and Sulphur atoms present in a peptide sequence. These five atoms form a size 5 binary vector. Their combinations form binary profile of each residue. For example residue of Alanine(A) contains 13 atoms in total. Thus binary profile of ‘A’ will be of size 13*5=65.

Bond binary profile is made based upon canonical smile (from PubChem) for each amino acid. Four kinds of bond considered c(cyclic), benzene ring(b), single bond(-) and double bond (=). Corresponding to these bonds binary vector is created.

4.4Residue Properties

This method generates the output as a binary profile for each sequence, which explains if a particular physicochemical property is present in a sequence. A single residue is represented by a vector of length 25, where each value is corresponding to a particular physicochemical property, if a particular residue is having the property then that position will be assigned as 1 else 0. Hence, if a sequence is given with length L, the output vector will be of size 25*L.

Physico-chemical property based binary profile for different portions of the sequence can be calculated using the standalone, library or webserver of Pfeature.

4.5AA Index

This sub-module gives the binary profile of input AA Indices. If normalised score of AAIndex value of a particular residue is negative, the function assigns ‘0’ to that residue otherwise assigns ‘1’. This method gives the binary profile for 553 amino acid indices. The resulting vector is of size 553*L, where L is the length of the protein/peptide sequence.

AAIndex-based binary profile for different portions of the sequence can be calculated using the standalone, library or webserver of Pfeature.

Chapter 5.0
Evolutionary Information Based Features

In this section, we have described the sub-modules under the module **"Evolutionary Information"** developed for amino acid evolutionary information based feature generation. These modules can be used for feature generation for protein sequences to apply machine learning techniques in the analysis where position of residues plays a significant role.

Figure 5.1: This flowchart shows Evolutionary Information based features

5.1 Generation of PSSM

This matrix is generated by using psi-blast against databases (nr or swissprot). The resultant matrix consists information of evolutionary conservation of elements of type x(i,j), where j refers to the amino acid residue at position i.

5.2 Normalization of PSSM

Various normalization operations are there to normalize the generated PSSM profile. Each normalization is explained as follows:

- **pssm_n1** : Due to the large number of variation in the value of PSSM matrix, it is necessary to normalize it. Each element of matrix is normalized by $\frac{1}{1+e^{-x}}$
- **pssm_n2** : This is the second technique to normalize the elements of PSSM matrix using the formula $\frac{(\text{num} - \text{min})}{(\text{max} - \text{min})}$
- **pssm_n3** : This is the third technique to normalize the matrix using the formula $\frac{(\text{num} - \text{min})}{(\text{max} - \text{min})} * 100$
- **pssm_n4** : This is the fourth technique to normalize the PSSM profile using the formula $\frac{1}{1+e^{-x/100}}$

5.3 Composition of PSSM

This function results the vector of 400 size. It calculates the frequency of amino acid composition corresponding to residue of peptide/protein sequence. Each column consists of 20 values.

5.4 Profile of PSSM

This matrix is generated by using psi-blast against databases (nr or swissprot). The resultant matrix consists information of evolutionary conservation of elements of type x(i,j), where j is a residue at position ‘i’.

In order to generate PSSM profile of different portions of an amino acid sequence, we have developed number of python function which is described in **Table 2.1**. In web server user may select portion of sequence for calculating PSSM based features.

Chapter 6.0
Structure Based Features

In this section, we have described the sub-modules under the module **"Structure"** developed to calculate structure based feature using tertiary structures of protein/peptide. These modules can be used for feature generation for protein structures to apply machine learning techniques for further analysis.

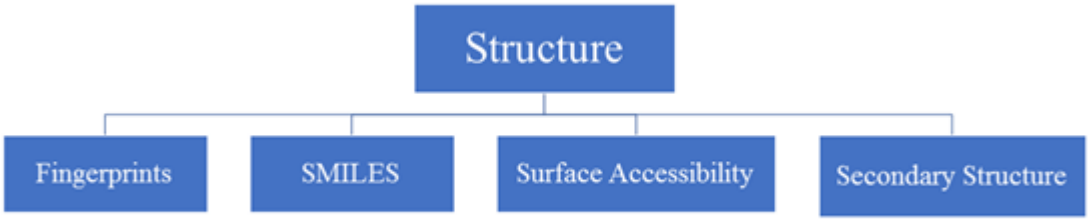


Figure 6.1: Sub-modules for Structure

6.1 Fingerprints

This module was developed to calculate different types of fingerprints descriptors. The fingerprints were calculated using PaDEL software, which is java based software. PaDEL software provides 10 different types of fingerprints types which in total provide 14,532 fingerprint values. These fingerprints are calculated using mainly The Chemistry Development Kit (CDK).

Along with CDK, other fingerprints present are Pubchem fingerprints, MACCS fingerprints, Klekota-Roth fingerprints. Fingerprints have been used as an important type of feature in various prediction methods developed previously in literature.

Usage: Here user needs to upload its molecular structure in PDB file format for calculating the fingerprints.

6.2 SMILES

SMILES stands for Simplified Molecular Input Line Entry System. It is a type of line notation for representing various molecules and reactions. It contains the same information as the extended data tables consists of. One of the advantage of using it is that it is easy to understand since it is a linguistic construct rather than a computer data structure. Also, the SMILES format takes 50-70% less space in comparison to other way of representing the information as well as required lesser time for processing the information. SMILES

notation is represented by series of characters and no spaces are present in between the characters.
SMILES notation follows five simple rules required for its encoding which are corresponding to atoms, bonds, branches, ring closures and disconnections. Detailed description of the SMILES notations can be obtained at http://www.daylight.com/dayhtml/doc/theory/theory.smiles.html .

Usage: Here, SMILES format were generated using openbabel software, where users are required to upload their structure in PDB file format in the SMILES page of pfeature in order to get the desired output.

6.3 Surface Accessibility

Accessible molecular surface or solvent-exposed area is defined as the area of an atom which can be touched by water molecule. Contact surface area and atoms chemical properties play an important role in modeling side chain conformations in proteins, structure and functional annotation of biological molecules. Here we have developed a module, which calculates the Relative Accessibility Area (RSA) using NACCESS software. The software requires PDB structure as an input and calculates relative accessible area. The output provided by the software shows value in percentage. In general values ranges between 0-100%; however, for some residues values go beyond 100%. In general, residues showing value less than 20% are said to buried whereas residues showing value above 20% are said to be exposed

Usage: Here, user needs to upload their structure in PDB file format and the server will calculate the relative accessibility area as an output.

6.4 Secondary Structure

Secondary structure refers to the interaction of hydrogen bond donor and acceptor residues of the repeating peptide unit. It plays an important role in protein structure prediction and protein folding. The two most important element of secondary structure are alpha helix and beta sheet. However, coils are also considered as an important type of secondary structure in many cases. There are many software presents in the literature which has been developed to predict the type of secondary structure. Since secondary structure elements represents an important type of feature, we have developed a module which calculates the percent average secondary structure element present in the input structure file. We have used DSSP software, which assigns the secondary structure state of the residue.

Usage: In order to calculate the percent average secondary structure element, user needs to upload the PDB file on to the server.

Chapter 7.0
Features Using Patterns

In this section, we have described the sub-modules under the module **"Pattern"** developed to generate the patterns. These modules can be used for feature generation for protein sequences or matrices to apply machine learning techniques.

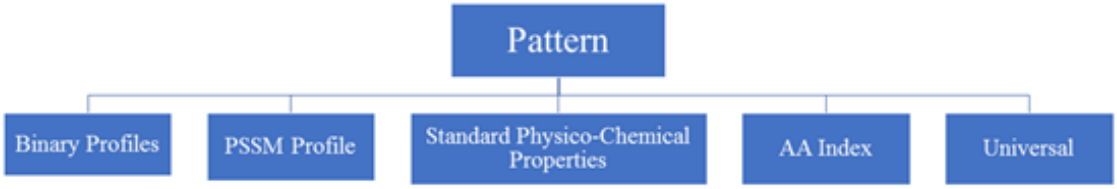


Figure 7.1 Sub-modules under module Pattern

7.1 Binary Profiles

This function is used to compute binary profile for the patterns of protein and peptide sequences. The patterns generated are in different window size. The window size will always be an odd number to generate equal size of the patterns. An extra 'X' is added in the starting and end of the sequence to make equal size patterns. The binary pattern is generated for the pattern generated.

7.2 PSSM Profile

This function is used to compute PSSM for patterns of protein and peptide sequences. Here the patterns are generated from PSSM matrix in different window size. The window size will always be an odd number to generate equal size of patterns. Here extra 'X' is added in the starting and end of the sequence to make the equal size patterns, so the vector size will be 21.

7.3 Standard Physico-Chemical Properties

This function generates patterns of desired length within sequences and then calculates the standard physicochemical properties (refer to section 3.2.1) of each generated pattern.

7.4 AA Index

This function generates patterns of desired length and then calculates average desired AA Index value for each generated pattern.

7.5 Universal

This function will generate patterns for any type of string like secondary structure, surface accessibility. These patterns are generated in sliding window manner and are of defined length. Additional ‘X’ is added on both the sides of the peptide sequence which results in the generation of equal length pattern.

Chapter 8.0
Portion of a Sequence

This module represents the operations applied on the length of peptide sequence. It has been shown in the literature that specific part or portion of the protein or peptide plays the major role in the functioning of that protein/peptide. Hence, to handle that situation we have provided five operations in our web-server, that same is shown in the Figure 8.1.

Figure 8.1: Portion of the sequence

8.1 Whole Amino Acid Sequence

This option allows users to compute features of a protein from whole sequence. This option is important for user when user wish to understand overall property of a protein or peptide. Most of methods developed in past use whole amino acid sequence of a protein.

8.2 N-Terminal

It has been observed in past that N-terminal of a protein is responsible for its function. For example most of classical secretory proteins contain a signal peptide. A short peptide (16-30 amino acids) present at the N-terminus of the majority of proteins that are destined towards the secretory pathway. Signal peptides are not only found in N-terminal of secretory proteins but found in number of other class of protein. Pfeature allow user to compute wide range of features in selected region (N-terminal) of a protein. One of the advantage of in selecting region is that user can generate both composition as well as binary profile as length of selected region is fixed **(BMC Bioinformatics 2007, 8:263 & BMC Bioinformatics 2010, 11:S19)**.

8.3 C-Terminal

It has been observed in past that C-terminal of a protein is responsible for its function. Normally. N-terminus of a protein often contains targeting signals, the C-terminus can

contain retention signals for protein sorting. The most common endoplasmic reticulum retention signal is the amino acid sequence KDEL or HDEL at the C-terminus. This keeps the protein in the endoplasmic reticulum and prevents it from entering the secretory pathway. Pfeature allow user to compute wide range of features in selected region (C-terminal) of a protein. One of the advantage of in selecting region is that user can generate both composition as well as binary profile as length of selected region is fixed **(BMC Bioinformatics 2007, 8:263 & BMC Bioinformatics 2010, 11:S19)**.

8.4 Split

One of the major problems with composition-based features is that that it present protein by limited features, it give only average features of whole sequence. In order to increase number of features to capture more information from a protein, split amino acid composition (SAAC) has been introduced (**J Biol Chem. 2006;281:5357-63)**. In this concept, amino acid is splitted in two or more than two portions then feature of each portion is computed separately. For example, is number of splits is three then sequence will be divided in three portions (each portion have nearly same length). If whole protein has 20 (composition) features then splitted composition provides 60 (20 X 3) features.

8.5 Rest

As shown in above section both terminals (N- & C-) have important information so Pfeature have provision to compute feature of N-terminal or C-terminal. In order to capture information or generating feature from remaining portion of proteins (after removing N-terminal and C-terminal residues). In case of rest option user need to select number of residues from N-terminal and C-terminal to be removed from protein for calculating features from rest of protein.

Chapter 9.0
Complete List of Features

In this section, we have elaborate and compared the features calculated by Pfeature and other available resources. Pfeature is able to calculate more than 72,000 composition features from the primary sequence of protein or peptide. In the table 3, we have described the group and type of features, kinds of sub-sequences, their dimension vectors, and methods which support the respective features.

Table 9.1: Brief description of features calculated by Pfeature

Type of Features	Description	Features	Dimension Vectors	Supported By
COMPOSITION: SIMPLE				
AAC	Amino acid Composition	Whole	20	{a,b,c,d,e}
		N-Terminal	20	{a}
		C-Terminal	20	{a}
		Rest	20	{a}
		Split	20*N	{a}
DPC	Dipeptide Composition	Whole	400	{a,b,c,d,e}
		N-Terminal	400	{a}
		C-Terminal	400	{a}
		Rest	400	{a}
		Split	400*N	{a}
TPC	Tripeptide Composition	Whole	8000	{a,b,c,d}
		N-Terminal	8000	{a}
		C-Terminal	8000	{a}
		Rest	8000	{a}
		Split	8000*N	{a}
ABC	Atom and Bond Composition	Whole	9	{a}
		N-Terminal	9	{a}
		C-Terminal	9	{a}
		Rest	9	{a}
		Split	9*N	{a}
COMPOSITION: PHYSICO-CHEMICAL PROPERTIES				
PCP	Physico-Chemical properties composition	Whole	19	{a,b,c,d,e}
		N-Terminal	19	{a}

		C-Terminal	19	{a}
		Rest	19	{a}
		Split	19*N	{a}
AAI	Amino Acid Index Composition	Whole	553	{a,b,c}
		N-Terminal	553	{a}
		C-Terminal	553	{a}
		Rest	553	{a}
		Split	553*N	{a}
PCP_adv	Advanced Physico-Chemical properties composition	Whole	5	{a,b,c,d,e}
		N-Terminal	5	{a}
		C-Terminal	5	{a}
		Rest	5	{a}
		Split	5*N	{a}
PCP_str	Structural Physico-Chemical properties composition	Whole	6	{a,b,c,d,e}
		N-Terminal	6	{a}
		C-Terminal	6	{a}
		Rest	6	{a}
		Split	6	{a}
COMPOSITION: REPEATS & DISTRIBUTION				
RRI	Repetitive Residue Information	Whole	20	{a}
		N-Terminal	20	{a}
		C-Terminal	20	{a}
		Rest	20	{a}
		Split	20*N	{a}
PRI	Repeat of Physico-chemical Properties	Whole	19	{a}
		N-Terminal	19	{a}
		C-Terminal	19	{a}
		Rest	19	{a}
		Split	19*N	{a}
DDR	Distance Distribution of Residues	Whole	20	{a}
		N-Terminal	20	{a}
		C-Terminal	20	{a}
		Rest	20	{a}
		Split	20*N	{a}

COMPOSITION: SHANNON ENTROPY				
SEP	Shannon Entropy at Protein Level	Whole	1	{a}
		N-Terminal	1	{a}
		C-Terminal	1	{a}
		Rest	1	{a}
		Split	1*N	{a}
SER	Shannon Entropy at Residue Level	Whole	20	{a}
		N-Terminal	20	{a}
		C-Terminal	20	{a}
		Rest	20	{a}
		Split	20*N	{a}
SPC	Shannon Entropy at Property Level	Whole	19	{a}
		N-Terminal	19	{a}
		C-Terminal	19	{a}
		Rest	19	{a}
		Split	19*N	{a}
COMPOSITION: MISCELLANEOUS				
ACR	Autocorrelation Descriptors	Whole	1659	{a,b,c,d,e}
		N-Terminal	1659	{a}
		C-Terminal	1659	{a}
		Rest	1659	{a}
		Split	1659*N	{a}
CTC	Conjoint Triad Descriptors	Whole	343	{a,b,c,d,e}
		N-Terminal	343	{a}
		C-Terminal	343	{a}
		Rest	343	{a}
		Split	343*N	{a}
CeTD	Composition enhanced Transition Distribution	Whole	189	{a,b,c,d,e}
		N-Terminal	189	{a}
		C-Terminal	189	{a}
		Rest	189	{a}
		Split	189*N	{a}
PAAC	Pseudo Amino Acid Composition	Whole	$20 + \lambda$	{a,b,c,d,e}
		N-Terminal	$20 + \lambda$	{a}
		C-Terminal	$20 + \lambda$	{a}
		Rest	$20 + \lambda$	{a}
		Split	$N*(20 + \lambda)$	{a}

APAAC	Amphiphilic Pseudo Amino Acid Composition	Whole	20 + (λ*3)	{a,b,c,d,e}
		N-Terminal	20 + (λ*3)	{a}
		C-Terminal	20 + (λ*3)	{a}
		Rest	20 + (λ*3)	{a}
		Split	N*(20 + (λ*3))	{a}
QSO	Quasi-Sequence Order	Whole	40 + (λ*2)	{a,b,c,d,e}
		N-Terminal	40 + (λ*2)	{a}
		C-Terminal	40 + (λ*2)	{a}
		Rest	40 + (λ*2)	{a}
		Split	N*(40 + (λ*2))	{a}
(SOCN)	Sequence Order Coupling Number	Whole	λ*2	{a,b,c,d,e}
		N-Terminal	λ*2	{a}
		C-Terminal	λ*2	{a}
		Rest	λ*2	{a}
		Split	N*λ*2	{a}
BINARY PROFILES				
AAB	Amino Acid Binary Profile	Whole	20*L	{a,b}
		N-Terminal	20*L	{a}
		C-Terminal	20*L	{a}
		Rest	20*L	{a}
		Split	N*(20*L)	{a}
DPB	Dipeptide Binary Profile	Whole	400*L	{a}
		N-Terminal	400*L	{a}
		C-Terminal	400*L	{a}
		Rest	400*L	{a}
		Split	N*(400*L)	{a}
ABB	Atom and Bond Binary Profile	Whole	(5*η)+(4*ε)	{a}
		N-Terminal	(5*η)+(4*ε)	{a}
		C-Terminal	(5*η)+(4*ε)	{a}
		Rest	(5*η)+(4*ε)	{a}
		Split	N*((5*η)+(4*ε))	{a}
PCB	Physico-Chemical Properties Binary Profile	Whole	25*L	{a}
		N-Terminal	25*L	{a}
		C-Terminal	25*L	{a}
		Rest	25*L	{a}
		Split	N*25*L	{a}
AIB	Amino Acid Index Binary Profile	Whole	553*L	{a}
		N-Terminal	553*L	{a}
		C-Terminal	553*L	{a}
		Rest	553*L	{a}

		Split	N*553*L	{a}
EVOLUTIONARY INFORMATION				
G_PSSM	Generation of PSSM	Whole	L X 21	{a}
N_PSSM	Normalization of PSSM	Whole	L X 21	{a}
C_PSSM	Composition of PSSM	Whole	400	{a}
P_PSSM	Profile of PSSM	Whole	L X 21	{a}
		N-Terminal	L X 21	{a}
		C-Terminal	L X 21	{a}
		Rest	L X 21	{a}
STRUCTURE				
FIN	Fingerprints	Whole	14532	{a}
SMI	SMILES	Whole	1	{a}
SA	Surface Accessibility	Whole	9	{a}
SS	Secondary Structure	Whole	3	{a}
PATTERN				
Binary Profile	Binary Profile generated using patterns of window length (ω)	Whole	L X (21*ω)	{a}
PSSM Profile	PSSM Profile generated using patterns of window length (ω)	Whole	L X (21*ω)	{a}
Physico-Chemical Properties	Physico-Chemical Properties, calculated using patterns of window length (ω)	Whole	L X (30*ω)	{a}
AA Index	Amino acid index composition, calculated using patterns of window length (ω)	Whole	L X 1	{a}
Universal	Generation of patterns of window length (ω)	Whole	L X ω	{a}
MODEL BUILDING				
Merging Features	Merge the two files into single file	2 CSV files	R X M	{a}
Feature Relevance	Mean based method to get the relevance of each feature	Positive and Negative Dataset	F X 9	{a}

a: Pfeature, b: ifeatue, c: PyBioMed, d: PyDPI, e: PROFEAT; L: length of protein; N: Number of splits; λ: The number depends upon the choice of maxlag; η: Number of atoms; ε: Number of bonds; R: Number of Rows; M: Total number of features in two files; F: Total number of features

Chapter 10.0
Feature Headers

10.0 List of Descriptors and Abbreviations

Amino Acid Composition (AAC): Total descriptor 20

AAC_A → Amino acid composition of Alanine

AAC_C → Amino acid composition of Cysteine

AAC_D → Amino acid composition of Aspartic acid

AAC_E → Amino acid composition of Glutamic acid

AAC_F → Amino acid composition of Phenylalanine

AAC_G → Amino acid composition of Glycine

AAC_H → Amino acid composition of Histidine

AAC_I → Amino acid composition of Isoleucine

AAC_K → Amino acid composition of Lysine

AAC_L → Amino acid composition of Leucine

AAC_M → Amino acid composition of Methionine

AAC_N → Amino acid composition of Asparagine

AAC_P → Amino acid composition of Proline

AAC_Q → Amino acid composition of Glutamine

AAC_R → Amino acid composition of Arginine

AAC_S → Amino acid composition of Serine

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

AAC_T → Amino acid composition of Threonine

AAC_V → Amino acid composition of Valine

AAC_W → Amino acid composition of Tryptophan

AAC_Y → Amino acid composition of Tyrosine

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Dipeptide Composition (order 1, traditional) : 400 dipeptide composition

DPC1_AA → Composition of Alanine-Alanine

DPC1_AC → Composition of Alanine-Cysteine

DPC1_YW → Composition of Alanine-Cysteine

DPC1_YY → Composition of Alanine-Cysteine

Dipeptide Composition (order 2, alternate) : 400 dipeptide composition

DPC2_AA → Composition of Alanine-Alanine

DPC2_AC → Composition of Alanine-Cysteine

DPC2_YW → Composition of Alanine-Cysteine

DPC2_YY → Composition of Alanine-Cysteine

Dipeptide Composition (order 3, with gap of 2 residues) : 400 dipeptide composition

DPC3_AA → Composition of Alanine-Alanine

DPC3_AC → Composition of Alanine-Cysteine

DPC3_YW → Composition of Alanine-Cysteine

DPC3_YY → Composition of Alanine-Cysteine

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Tripeptide Composition: 8000 tripeptide composition

TPC_AAA → Composition of Alanine-Alanine-Alanine

TPC_AAC → Composition of Alanine-Alanine-Cysteine

TPC_AAD → Composition of Alanine-Alanine-Aspartic acid

TPC_AAE → Composition of Alanine-Alanine-Glutamic acid

TPC_AAF → Composition of Alanine-Alanine-Phenylalanine

TPC_AAG → Composition of Alanine-Alanine-Glycine

TPC_AAH → Composition of Alanine-Alanine-Histidine

TPC_AAI → Composition of Alanine-Alanine-Isoleucine

TPC_AAK → Composition of Alanine-Alanine-Lysine

TPC_AAL → Composition of Alanine-Alanine-Leucine

TPC_YYM → Composition of Tyrosine-Tyrosine-Methionine

TPC_YYN → Composition of Tyrosine-Tyrosine-Asparagine

TPC_YYP → Composition of Tyrosine-Tyrosine-Proline

TPC_YYQ → Composition of Tyrosine-Tyrosine-Glutamine

TPC_YYR → Composition of Tyrosine-Tyrosine-Arginine

TPC_YYS → Composition of Tyrosine-Tyrosine-Serine

TPC_YYT → Composition of Tyrosine-Tyrosine-Threonine

TPC_YYV → Composition of Tyrosine-Tyrosine-Valine

TPC_YYW → Composition of Tyrosine-Tyrosine-Tryptophan

Note: ‘N’ prefix is added to the header for choosing N-terminal residues, ‘C’ prefix is added to the header for choosing C-terminal residues, ‘R’ prefix is added to the header for choosing rest method, ‘NC’ prefix is added to the header if NC-terminal has chosen, ‘RNC’ prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of ‘sn,’ where n is the number of splits, is added on choosing the split option.

TPC_YYY → Composition of Tyrosine-Tyrosine- Tyrosine

Atom Type Composition: 5 descriptors

ATC_C → Atomic Composition of Carbon

ATC_H → Atomic Composition of Hydrogen

ATC_N → Atomic Composition of Nitrogen

ATC_O → Atomic Composition of Oxygen

ATC_S → Atomic Composition of Sulphur

Bond Type Composition: 4 descriptors

BTC_T → Composition of total bonds

BTC_H → Composition of Hydrogen bonds

BTC_S → Composition of Single bonds

BTC_D → Composition of Double bonds

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Physico-chemical properties: 30 descriptors

PCP_PC → Composition of positively charged residues

PCP_NC → Composition of positively charged residues

PCP_NE → Composition of neutral charged residues

PCP_PO → Composition of polar residues

PCP_NP → Composition of non-polar residues

PCP_AL → Composition of residues having aliphatic side chain

PCP_CY → Composition of residues having cyclic side chain

PCP_AR → Composition of aromatic residues

PCP_AC → Composition of acidic residues

PCP_BS → Composition of basic residues

PCP_NE_ph → Composition of neutral residues based on pH

PCP_HB → Composition of hydrophobic residues

PCP_HL → Composition of hydrophilic residues

PCP_NT → Composition of neutral residues

PCP_HX → Composition of hydroxylic residues

PCP_SC → Composition of residues having sulphur content

PCP_SS_HE → Composition of residue in secondary structure (Helix)

PCP_SS_ST → Composition of residue in secondary structure (Strands)

PCP_SS_CO → Composition of residue in secondary structure (Coil)

PCP_SA_BU → Composition of residue in solvent accessibility (Buried)

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

PCP_SA_EX → Composition of residue in solvent accessibility (Exposed)

PCP_SA_IN → Composition of residue in solvent accessibility (Intermediate)

PCP_TN → Composition of tiny residues

PCP_SM → Composition of small residues

PCP_LR → Composition of large residues

PCP_Z1 → Composition of residues having Z1 advanced Physico-chemical properties

PCP_Z2 → Composition of residues having Z2 advanced Physico-chemical properties

PCP_Z3 → Composition of residues having Z3 advanced Physico-chemical properties

PCP_Z4 → Composition of residues having Z4 advanced Physico-chemical properties

PCP_Z5 → Composition of residues having Z5 advanced Physico-chemical properties

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Amino Acid Index: 553 type descriptors

AAI_ANDN920101 → Composition of index ANDN920101

AAI_ARGP820101 → Composition of index ARGP820101

AAI_ARGP820102 → Composition of index ARGP820102

AAI_ARGP820103 → Composition of index ARGP820103

AAI_BEGF750101 → Composition of index BEGF750101

AAI_BEGF750102 → Composition of index BEGF750102

AAI_BEGF750103 → Composition of index BEGF750103

AAI_BHAR880101 → Composition of index BHAR880101

AAI_BIGC670101 → Composition of index BIGC670101

AAI_BIOV880101 → Composition of index BIOV880101

AAI_KARS160113 → Composition of index KARS160113

AAI_KARS160114 → Composition of index KARS160114

AAI_KARS160115 → Composition of index KARS160115

AAI_KARS160116 → Composition of index KARS160116

AAI_KARS160117 → Composition of index KARS160117

AAI_KARS160118 → Composition of index KARS160118

AAI_KARS160119 → Composition of index KARS160119

AAI_KARS160120 → Composition of index KARS160120

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

AAI_KARS160121 → Composition of index KARS160121

AAI_KARS160122 → Composition of index KARS160122

Note: ‘N’ prefix is added to the header for choosing N-terminal residues, ‘C’ prefix is added to the header for choosing C-terminal residues, ‘R’ prefix is added to the header for choosing rest method, ‘NC’ prefix is added to the header if NC-terminal has chosen, ‘RNC’ prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of ‘sn,’ where n is the number of splits, is added on choosing the split option.

Residue Repeats Index: 20 descriptors

RRI_A → Residue repeat index of Alanine

RRI_C → Residue repeat index of Cysteine

RRI_D → Residue repeat index of Aspartic acid

RRI_E → Residue repeat index of Glutamic acid

RRI_F → Residue repeat index of Phenylalanine

RRI_G → Residue repeat index of Glycine

RRI_H → Residue repeat index of Histidine

RRI_I → Residue repeat index of Isoleucine

RRI_K → Residue repeat index of Lysine

RRI_L → Residue repeat index of Leucine

RRI_M → Residue repeat index of Methionine

RRI_N → Residue repeat index of Asparagine

RRI_P → Residue repeat index of Proline

RRI_Q → Residue repeat index of Glutamine

RRI_R → Residue repeat index of Arginine

RRI_S → Residue repeat index of Serine

RRI_T → Residue repeat index of Threonine

RRI_V → Residue repeat index of Valine

RRI_W → Residue repeat index of Tryptophan

RRI_Y → Residue repeat index of Tyrosine

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Property Repeats Index: 25 descriptors corresponding to 25 Physico-chemical properties

PRI_PC → Residue repeat index for positive charged residues

PRI_PC → Residue repeat index for negative charged residues

PRI_NE → Residue repeat index for neutral charged residues

PRI_PO → Residue repeat index for polar residues

PRI_NP → Residue repeat index for non-polar residues

PRI_AL → Residue repeat index for residues having aliphatic side chain

PRI_CY → Residue repeat index for residues having cyclic side chain

PRI_AR → Residue repeat index for aromatic residues

PRI_AC → Residue repeat index for acidic residues

PRI_BS → Residue repeat index for basic residues

PRI_NE → Residue repeat index for neutral residues based on pH

PRI_HB → Residue repeat index for hydrophobic residues

PRI_HL → Residue repeat index for hydrophilic residues

PRI_NT → Residue repeat index for neutral residues

PRI_HX → Residue repeat index for hydroxylic residues

PRI_SC → Residue repeat index for residues having sulphur content

PRI_SS_HE → Residue repeat index for residues in secondary structure (Helix)

PRI_SS_ST → Residue repeat index for residues in secondary structure (Strands)

PRI_SS_CO → Residue repeat index for residues in secondary structure (Coil)

PRI_SA_BU → Residue repeat index for residues in solvent accessibility (Buried)

PRI_SA_EX → Residue repeat index for residues in solvent accessibility (Exposed)

PRI_SA_IN → Residue repeat index for residues in solvent accessibility (Intermediate)

PRI_TN → Residue repeat index for tiny residues

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

PRI_SM → Residue repeat index for small residues

PRI_LR → Residue repeat index for large residues

Note: ‘N’ prefix is added to the header for choosing N-terminal residues, ‘C’ prefix is added to the header for choosing C-terminal residues, ‘R’ prefix is added to the header for choosing rest method, ‘NC’ prefix is added to the header if NC-terminal has chosen, ‘RNC’ prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of ‘sn,’ where n is the number of splits, is added on choosing the split option.

Distance Distribution of Repeats: 20 type of residues

DDR_A → Distribution of Alanine

DDR_C → Distribution of Cysteine

DDR_D → Distribution of Aspartic acid

DDR_E → Distribution of Glutamic acid

DDR_F → Distribution of Phenylalanine

DDR_G → Distribution of Glycine

DDR_H → Distribution of Histidine

DDR_I → Distribution of Isoleucine

DDR_K → Distribution of Lysine

DDR_L → Distribution of Leucine

DDR_M → Distribution of Methionine

DDR_N → Distribution of Asparagine

DDR_P → Distribution of Proline

DDR_Q → Distribution of Glutamine

DDR_R → Distribution of Arginine

DDR_S → Distribution of Serine

DDR_T → Distribution of Threonine

DDR_V → Distribution of Valine

DDR_W → Distribution of Tryptophan

DDR_Y → Distribution of Tyrosine

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Shannon Entropy of a Protein: 1 Descriptor

SEP → Shannon entropy of whole protein

Shannon Entropy of a Residue: 20 Descriptors

SER_A → Shannon entropy of Alanine

SER_C → Shannon entropy of Cysteine

SER_D → Shannon entropy of Aspartic acid

SER_E → Shannon entropy of Glutamic acid

SER_F → Shannon entropy of Phenylalanine

SER_G → Shannon entropy of Glycine

SER_H → Shannon entropy of Histidine

SER_I → Shannon entropy of Isoleucine

SER_K → Shannon entropy of Lysine

SER_L → Shannon entropy of Leucine

SER_M → Shannon entropy of Methionine

SER_N → Shannon entropy of Asparagine

SER_P → Shannon entropy of Proline

SER_Q → Shannon entropy of Glutamine

SER_R → Shannon entropy of Arginine

SER_S → Shannon entropy of Serine

SER_T → Shannon entropy of Threonine

SER_V → Shannon entropy of Valine

SER_W → Shannon entropy of Tryptophan

SER_Y → Shannon entropy of Tyrosine

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Shannon Entropy of Properties:25 features corresponding to 25 physicochemical properties

SEP_PC → Shannon entropy of positive charged residues

SEP_PC → Shannon entropy of negative charged residues

SEP_NE → Shannon entropy of neutral charged residues

SEP_PO → Shannon entropy of polar residues

SEP_NP → Shannon entropy of non-polar residues

SEP_AL → Shannon entropy of residues having aliphatic side chain

SEP_CY → Shannon entropy of residues having cyclic side chain

SEP_AR → Shannon entropy of aromatic residues

SEP_AC → Shannon entropy of acidic residues

SEP_BS → Shannon entropy of basic residues

SEP_NE → Shannon entropy of neutral residues based on pH

SEP_HB → Shannon entropy of hydrophobic residues

SEP_HL → Shannon entropy of hydrophilic residues

SEP_NT → Shannon entropy of neutral residues

SEP_HX → Shannon entropy of hydroxylic residues

SEP_SC → Shannon entropy of residues having sulphur content

SEP_SS_HE → Shannon entropy of residue in secondary structure (Helix)

SEP _SS_ST → Shannon entropy of residue in secondary structure (Strands)

SEP _SS_CO → Shannon entropy of residue in secondary structure (Coil)

SEP _SA_BU → Shannon entropy of residue in solvent accessibility (Buried)

SEP _SA_EX → Shannon entropy of residue in solvent accessibility (Exposed)

SEP _SA_IN → Shannon entropy of residue in solvent accessibility (Intermediate)

SEP_TN → Shannon entropy of tiny residues

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

SEP_SM → Shannon entropy of small residues

SEP_LR → Shannon entropy of large residues

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Autocorrelation : 3 descriptors (Dong, Jie, et al. *Journal of cheminformatics* (2018),10.1:16)

ACR1_MB → Normalized Moreau-Broto autocorrelation descriptor with lag 1

ACR1_MO → Morgan autocorrelation descriptor with lag 1

ACR1_GE → Geary autocorrelation descriptor with lag 1

Conjoint Triad Descriptors: 343 descriptors (Dong, Jie, et al. *Journal of cheminformatics* (2018),10.1:16)

Group 1: A, G, V

Group 2: I, L, F, P

Group 3: Y, M, T, S

Group 4: H, N, Q, W

Group 5: R, K

Group 6: D, E

Group 7: C

CTC_111 → Normalize frequency of group1-group1-group1 (tri-group)

CTC_112 → Normalize frequency of group1-group1-group2 (tri-group)

CTC_113→ Normalize frequency of group1-group1-group3 (tri-group)

CTC_775 → Normalize frequency of group7-group7-group5 (tri-group)

CTC_776 → Normalize frequency of group7-group7-group6 (tri-group)

CCT_777 → Normalize frequency of group7-group7-group7 (tri-group)

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Composition enhanced Transition and Distribution: 189 descriptors (Dubchak I, et al. *Proceedings of the National Academy of Sciences of the United States of America*)

Attributes	Group1	Group 2	Group 3
Hydrophobicity	R,K,E,D,Q,N	G,A,S,T,P,H,Y	C,L,V,I,M,F, W
Normalized Vander Waals volume	G,A,S,T,P,D	N,V,E,Q,I,L	M,H,K,F,R,Y, W
Polarity	L,I,F,W,C,M,V, Y	P,A,T,G,S	H,Q,R,K,N,E, D
Polarizability	G,A,S,D,T	C,P,N,V,E,Q,I,L	K,M,H,F,R,Y, W
Charge	K,R	A,N,C,Q,G,H,I,L,M,F,P,S,T,W, Y,V	D,E
Secondary structure	E,A,L,M,Q,K,R ,H	V,I,Y,C,W,F,T	G,N,P,S,D
Solvent accessibility	A,L,F,C,G,I,V, W	R,K,Q,E,N,D	M,S P,T,H,Y

- **Composition:** 21 Descriptors

CeTD_HB1 → Composition of group 1 residues for hydrophobicity attribute

CeTD_HB2 → Composition of group 2 residues for hydrophobicity attribute

CeTD_HB3 → Composition of group 3 residues for hydrophobicity attribute

CeTD_VW1 → Composition of group 1 residues for normalized vander waals volume attribute

CeTD_VW2 → Composition of group 2 residues for normalized vander waals volume attribute

CeTD_VW3 → Composition of group 2 residues for normalized vander waals volume attribute

CeTD_PO1 → Composition of group 1 residues for polarity attribute

CeTD_PO2 → Composition of group 2 residues for polarity attribute

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

CeTD_PO3 → Composition of group 3 residues for polarity attribute

CeTD_PZ1 → Composition of group 1 residues for polarizability attribute

CeTD_PZ2 → Composition of group 2 residues for polarizability attribute

CeTD_PZ3 → Composition of group 3 residues for polarizability attribute

CeTD_CH1 → Composition of group 1 residues for charge attribute

CeTD_CH2 → Composition of group 2 residues for charge attribute

CeTD_CH3 → Composition of group 3 residues for charge attribute

CeTD_SS1 → Composition of group 1 residues for secondary structure attribute

CeTD_SS2 → Composition of group 2 residues for secondary structure attribute

CeTD_SS3 → Composition of group 3 residues for secondary structure attribute

CeTD_SA1 → Composition of group 1 residues for solvent accessibility attribute

CeTD_SA2 → Composition of group 2 residues for solvent accessibility attribute

CeTD_SA3 → Composition of group 3 residues for solvent accessibility attribute

- **Transition:** 63 Descriptors

  CeTD_11_HB → Number of transitions takes place from group 1 residues to group 1 residues for hydrophobicity attribute

  CeTD_11_VW → Number of transitions takes place from group 1 residues to group 1 residues for normalized vander waals volume attribute

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

CeTD_11_PO → Number of transitions takes place from group 1 residues to group 1 residues for polarity attribute

---

CeTD_12_HB → Number of transitions takes place from group 1 residues to group 2 residues for hydrophobicity attribute

CeTD_12_VW → Number of transitions takes place from group 1 residues to group 2 residues for normalized vander waals volume attribute

CeTD_12_PO → Number of transitions takes place from group 1 residues to group 2 residues for polarity attribute

---

CeTD_33_CH → Number of transitions takes place from group 3 residues to group 3 residues for charge attribute

CeTD_33_SS → Number of transitions takes place from group 3 residues to group 3 residues for secondary structure attribute

CeTD_33_SA → Number of transitions takes place from group 3 residues to group 3 residues for solvent accessibility attribute

- **Distribution:** 105 Descriptors

CeTD_0_p_HB1 → Number of group 1 residues for hydrophobicity present in 0% quartile

CeTD_25_p_HB1 → Number of group 1 residues for hydrophobicity present in 25% quartile

CeTD_50_p_HB1 → Number of group 1 residues for hydrophobicity present in 50% quartile

CeTD_75_p_HB1 → Number of group 1 residues for hydrophobicity present in 75% quartile

CeTD_100_p_HB1 → Number of group 1 residues for hydrophobicity present in 100% quartile

CeTD_0_p_VW1 → Number of group 1 residues for normalized vander waals volume present in 0% quartile

CeTD_25_p_VW1 → Number of group 1 residues for normalized vander waals volume present in 25% quartile

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

CeTD_50_p_VW1 → Number of group 1 residues for normalized vander waals volume present in 50% quartile

CeTD_75_p_VW1 → Number of group 1 residues for normalized vander waals volume present in 75% quartile

CeTD_100_p_VW1 → Number of group 1 residues for normalized vander waals volume present in 100% quartile

---

CeTD_0_p_HB2 → Number of group 2 residues for hydrophobicity present in 0% quartile

CeTD_25_p_HB2 → Number of group 2 residues for hydrophobicity present in 25% quartile

CeTD_50_p_HB2 → Number of group 2 residues for hydrophobicity present in 50% quartile

CeTD_75_p_HB2 → Number of group 2 residues for hydrophobicity present in 75% quartile

CeTD_100_p_HB2 → Number of group 2 residues for hydrophobicity present in 100% quartile

CeTD_0_p_VW2 → Number of group 2 residues for normalized vander waals volume present in 0% quartile

CeTD_25_p_VW2 → Number of group 2 residues for normalized vander waals volume present in 25% quartile

CeTD_50_p_VW2 → Number of group 2 residues for normalized vander waals volume present in 50% quartile

CeTD_75_p_VW2 → Number of group 2 residues for normalized vander waals volume present in 75% quartile

CeTD_100_p_VW2 → Number of group 2 residues for normalized vander waals volume present in 100% quartile

---

CeTD_0_p_SA3 → Number of group 2 residues for solvent accessibility present in 0% quartile

CeTD_25_p_ SA3 → Number of group 2 residues for solvent accessibility present in 25% quartile

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

CeTD_50_p_ SA3 → Number of group 2 residues for solvent accessibility present in 50% quartile

CeTD_75_p_ SA3 → Number of group 2 residues for solvent accessibility present in 75% quartile

CeTD_100_p_ SA3 → Number of group 2 residues for solvent accessibility present in 100% quartile

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Pseudo Amino Acid Composition (order 1, traditional): 21 descriptors (Chou KC, 2001, *Proteins*)

PAAC1_A → Pseudo amino acid composition of Alanine

PAAC1_C → Pseudo amino acid composition of Cysteine

PAAC1_D → Pseudo amino acid composition of Aspartic acid

PAAC1_E → Pseudo amino acid composition of Glutamic acid

PAAC1_F → Pseudo amino acid composition of Phenylalanine

PAAC1_G → Pseudo amino acid composition of Glycine

PAAC1_H → Pseudo amino acid composition of Histidine

PAAC1_I → Pseudo amino acid composition of Isoleucine

PAAC1_K → Pseudo amino acid composition of Lysine

PAAC1_L → Pseudo amino acid composition of Leucine

PAAC1_M → Pseudo amino acid composition of Methionine

PAAC1_N → Pseudo amino acid composition of Asparagine

PAAC1_P → Pseudo amino acid composition of Proline

PAAC1_Q → Pseudo amino acid composition of Glutamine

PAAC1_R → Pseudo amino acid composition of Arginine

PAAC1_S → Pseudo amino acid composition of Serine

PAAC1_T → Pseudo amino acid composition of Threonine

PAAC1_V → Pseudo amino acid composition of Valine

PAAC1_W → Pseudo amino acid composition of Tryptophan

PAAC1_Y → Pseudo amino acid composition of Tyrosine

PAAC1_lam1 → Sequence correlation factor for lambda 1

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Pseudo Amino Acid Composition (order 2, alternate): 22 descriptors

PAAC2_A → Pseudo amino acid composition of Alanine

PAAC2_C → Pseudo amino acid composition of Cysteine

PAAC2_D → Pseudo amino acid composition of Aspartic acid

PAAC2_E → Pseudo amino acid composition of Glutamic acid

PAAC2_F → Pseudo amino acid composition of Phenylalanine

PAAC2_G → Pseudo amino acid composition of Glycine

PAAC2_H → Pseudo amino acid composition of Histidine

PAAC2_I → Pseudo amino acid composition of Isoleucine

PAAC2_K → Pseudo amino acid composition of Lysine

PAAC2_L → Pseudo amino acid composition of Leucine

PAAC2_M → Pseudo amino acid composition of Methionine

PAAC2_N → Pseudo amino acid composition of Asparagine

PAAC2_P → Pseudo amino acid composition of Proline

PAAC2_Q → Pseudo amino acid composition of Glutamine

PAAC2_R → Pseudo amino acid composition of Arginine

PAAC2_S → Pseudo amino acid composition of Serine

PAAC2_T → Pseudo amino acid composition of Threonine

PAAC2_V → Pseudo amino acid composition of Valine

PAAC2_W → Pseudo amino acid composition of Tryptophan

PAAC2_Y → Pseudo amino acid composition of Tyrosine

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

PAAC2_lam1 → Sequence correlation factor for lambda 1

PAAC2_lam2 → Sequence correlation factor for lambda 2

Pseudo Amino Acid Composition (order 3, With gap of 2 residues): 23 descriptors

PAAC3_A → Pseudo amino acid composition of Alanine

PAAC3_C → Pseudo amino acid composition of Cysteine

PAAC3_D → Pseudo amino acid composition of Aspartic acid

PAAC3_E → Pseudo amino acid composition of Glutamic acid

PAAC3_F → Pseudo amino acid composition of Phenylalanine

PAAC3_G → Pseudo amino acid composition of Glycine

PAAC3_H → Pseudo amino acid composition of Histidine

PAAC3_I → Pseudo amino acid composition of Isoleucine

PAAC3_K → Pseudo amino acid composition of Lysine

PAAC3_L → Pseudo amino acid composition of Leucine

PAAC3_M → Pseudo amino acid composition of Methionine

PAAC3_N → Pseudo amino acid composition of Asparagine

PAAC3_P → Pseudo amino acid composition of Proline

PAAC3_Q → Pseudo amino acid composition of Glutamine

PAAC3_R → Pseudo amino acid composition of Arginine

PAAC3_S → Pseudo amino acid composition of Serine

PAAC3_T → Pseudo amino acid composition of Threonine

PAAC3_V → Pseudo amino acid composition of Valine

PAAC3_W → Pseudo amino acid composition of Tryptophan

PAAC3_Y → Pseudo amino acid composition of Tyrosine

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

PAAC3_lam1 → Sequence correlation factor for lambda 1

PAAC3_lam2 → Sequence correlation factor for lambda 2

PAAC3_lam3 → Sequence correlation factor for lambda 3

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Amphiphilic Pseudo Amino Acid Composition (order 1, traditional): 23 descriptors

APAAC1_A → Amphiphilic pseudo amino acid composition of Alanine

APAAC1_C → Amphiphilic pseudo amino acid composition of Cysteine

APAAC1_D → Amphiphilic pseudo amino acid composition of Aspartic acid

APAAC1_E → Amphiphilic pseudo amino acid composition of Glutamic acid

APAAC1_F → Amphiphilic pseudo amino acid composition of Phenylalanine

APAAC1_G → Amphiphilic pseudo amino acid composition of Glycine

APAAC1_H → Amphiphilic pseudo amino acid composition of Histidine

APAAC1_I → Amphiphilic pseudo amino acid composition of Isoleucine

APAAC1_K → Amphiphilic pseudo amino acid composition of Lysine

APAAC1_L → Amphiphilic pseudo amino acid composition of Leucine

APAAC1_M → Amphiphilic pseudo amino acid composition of Methionine

APAAC1_N → Amphiphilic pseudo amino acid composition of Asparagine

APAAC1_P → Amphiphilic pseudo amino acid composition of Proline

APAAC1_Q → Amphiphilic pseudo amino acid composition of Glutamine

APAAC1_R → Amphiphilic pseudo amino acid composition of Arginine

APAAC1_S → Amphiphilic pseudo amino acid composition of Serine

APAAC1_T → Amphiphilic pseudo amino acid composition of Threonine

APAAC1_V → Amphiphilic pseudo amino acid composition of Valine

APAAC1_W → Amphiphilic pseudo amino acid composition of Tryptophan

APAAC1_Y → Amphiphilic pseudo amino acid composition of Tyrosine

APAAC1_HB_lam1 → Sequence correlation factor for hydrophobicity with lambda 1

APAAC1_HL_lam1 → Sequence correlation factor for hydrophilicity with lambda 1

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

APAAC1_SC_lam1 → Sequence correlation factor for side chain mass with lambda 1

Amphiphilic Pseudo Amino Acid Composition (order 2, alternate): 26 descriptors

APAAC2_A → Amphiphilic pseudo amino acid composition of Alanine

APAAC2_C → Amphiphilic pseudo amino acid composition of Cysteine

APAAC2_D → Amphiphilic pseudo amino acid composition of Aspartic acid

APAAC2_E → Amphiphilic pseudo amino acid composition of Glutamic acid

APAAC2_F → Amphiphilic pseudo amino acid composition of Phenylalanine

APAAC2_G → Amphiphilic pseudo amino acid composition of Glycine

APAAC2_H → Amphiphilic pseudo amino acid composition of Histidine

APAAC2_I → Amphiphilic pseudo amino acid composition of Isoleucine

APAAC2_K → Amphiphilic pseudo amino acid composition of Lysine

APAAC2_L → Amphiphilic pseudo amino acid composition of Leucine

APAAC2_M → Amphiphilic pseudo amino acid composition of Methionine

APAAC2_N → Amphiphilic pseudo amino acid composition of Asparagine

APAAC2_P → Amphiphilic pseudo amino acid composition of Proline

APAAC2_Q → Amphiphilic pseudo amino acid composition of Glutamine

APAAC2_R → Amphiphilic pseudo amino acid composition of Arginine

APAAC2_S → Amphiphilic pseudo amino acid composition of Serine

APAAC2_T → Amphiphilic pseudo amino acid composition of Threonine

APAAC2_V → Amphiphilic pseudo amino acid composition of Valine

APAAC2_W → Amphiphilic pseudo amino acid composition of Tryptophan

APAAC2_Y → Amphiphilic pseudo amino acid composition of Tyrosine

APAAC2_HB_lam1 → Sequence correlation factor for hydrophobicity with lambda 1

APAAC2_HL_lam1 → Sequence correlation factor for hydrophilicity with lambda 1

Note: ‘N’ prefix is added to the header for choosing N-terminal residues, ‘C’ prefix is added to the header for choosing C-terminal residues, ‘R’ prefix is added to the header for choosing rest method, ‘NC’ prefix is added to the header if NC-terminal has chosen, ‘RNC’ prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of ‘sn,’ where n is the number of splits, is added on choosing the split option.

APAAC2_SC_lam1 → Sequence correlation factor for side chain mass with lambda 1

APAAC2_HB_lam2 → Sequence correlation factor for hydrophobicity with lambda 2

APAAC2_HL_lam2 → Sequence correlation factor for hydrophilicity with lambda 2

APAAC2_SC_lam2 → Sequence correlation factor for side chain mass with lambda 2

Amphiphilic Pseudo Amino Acid Composition (order 3, With gap of 2 residues): 29 descriptors

APAAC3_A → Amphiphilic pseudo amino acid composition of Alanine

APAAC3_C → Amphiphilic pseudo amino acid composition of Cysteine

APAAC3_D → Amphiphilic pseudo amino acid composition of Aspartic acid

APAAC3_E → Amphiphilic pseudo amino acid composition of Glutamic acid

APAAC3_F → Amphiphilic pseudo amino acid composition of Phenylalanine

APAAC3_G → Amphiphilic pseudo amino acid composition of Glycine

APAAC3_H → Amphiphilic pseudo amino acid composition of Histidine

APAAC3_I → Amphiphilic pseudo amino acid composition of Isoleucine

APAAC3_K → Amphiphilic pseudo amino acid composition of Lysine

APAAC3_L → Amphiphilic pseudo amino acid composition of Leucine

APAAC3_M → Amphiphilic pseudo amino acid composition of Methionine

APAAC3_N → Amphiphilic pseudo amino acid composition of Asparagine

APAAC3_P → Amphiphilic pseudo amino acid composition of Proline

APAAC3_Q → Amphiphilic pseudo amino acid composition of Glutamine

APAAC3_R → Amphiphilic pseudo amino acid composition of Arginine

APAAC3_S → Amphiphilic pseudo amino acid composition of Serine

APAAC3_T → Amphiphilic pseudo amino acid composition of Threonine

APAAC3_V → Amphiphilic pseudo amino acid composition of Valine

APAAC3_W → Amphiphilic pseudo amino acid composition of Tryptophan

APAAC3_Y → Amphiphilic pseudo amino acid composition of Tyrosine

APAAC3_HB_lam1 → Sequence correlation factor for hydrophobicity with lambda 1

APAAC3_HL_lam1 → Sequence correlation factor for hydrophilicity with lambda 1

APAAC3_SC_lam1 → Sequence correlation factor for side chain mass with lambda 1

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

APAAC3_HB_lam2 → Sequence correlation factor for hydrophobicity with lambda 2

APAAC3_HL_lam2 → Sequence correlation factor for hydrophilicity with lambda 2

APAAC3_SC_lam2 → Sequence correlation factor for side chain mass with lambda 2

APAAC3_HB_lam3 → Sequence correlation factor for hydrophobicity with lambda 3

APAAC3_HL_lam3 → Sequence correlation factor for hydrophilicity with lambda 3

APAAC3_SC_lam3 → Sequence correlation factor for side chain mass with lambda 3

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Quasi-Sequence Order (order 1, traditional): 42 Descriptors (Chou KC, 2000, Biochemical and Biophysical Research Communications)

QSO1_SC_A → Quasi-sequence order with Schneider matrix for Alanine

QSO1_SC_C → Quasi-sequence order with Schneider matrix for Cysteine

QSO1_SC_D → Quasi-sequence order with Schneider matrix for Aspartic acid

QSO1_SC_E → Quasi-sequence order with Schneider matrix for Glutamic acid

QSO1_SC_F → Quasi-sequence order with Schneider matrix for Phenylalanine

QSO1_SC_G → Quasi-sequence order with Schneider matrix for Glycine

QSO1_SC_H → Quasi-sequence order with Schneider matrix for Histidine

QSO1_SC_I → Quasi-sequence order with Schneider matrix for Isoleucine

QSO1_SC_K → Quasi-sequence order with Schneider matrix for Lysine

QSO1_SC_L → Quasi-sequence order with Schneider matrix for Leucine

QSO1_SC_M → Quasi-sequence order with Schneider matrix for Methionine

QSO1_SC_N → Quasi-sequence order with Schneider matrix for Asparagine

QSO1_SC_P → Quasi-sequence order with Schneider matrix for Proline

QSO1_SC_Q → Quasi-sequence order with Schneider matrix for Glutamine

QSO1_SC_R → Quasi-sequence order with Schneider matrix for Arginine

QSO1_SC_S → Quasi-sequence order with Schneider matrix for Serine

QSO1_SC_T → Quasi-sequence order with Schneider matrix for Threonine

QSO1_SC_V → Quasi-sequence order with Schneider matrix for Valine

QSO1_SC_W → Quasi-sequence order with Schneider matrix for Tryptophan

QSO1_SC_Y → Quasi-sequence order with Schneider matrix for Tyrosine

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

QSO1_G_A → Quasi-sequence order with Grantham matrix for Alanine

QSO1_G_C → Quasi-sequence order with Grantham matrix for Cysteine

QSO1_G_D → Quasi-sequence order with Grantham matrix for Aspartic acid

QSO1_G_E → Quasi-sequence order with Grantham matrix for Glutamic acid

QSO1_G_F → Quasi-sequence order with Grantham matrix for Phenylalanine

QSO1_G_G → Quasi-sequence order with Grantham matrix for Glycine

QSO1_G_H → Quasi-sequence order with Grantham matrix for Histidine

QSO1_G_I → Quasi-sequence order with Grantham matrix for Isoleucine

QSO1_G_K → Quasi-sequence order with Grantham matrix for Lysine

QSO1_G_L → Quasi-sequence order with Grantham matrix for Leucine

QSO1_G_M → Quasi-sequence order with Grantham matrix for Methionine

QSO1_G_N → Quasi-sequence order with Grantham matrix for Asparagine

QSO1_G_P → Quasi-sequence order with Grantham matrix for Proline

QSO1_G_Q → Quasi-sequence order with Grantham matrix for Glutamine

QSO1_G_R → Quasi-sequence order with Grantham matrix for Arginine

QSO1_G_S → Quasi-sequence order with Grantham matrix for Serine

QSO1_G_T → Quasi-sequence order with Grantham matrix for Threonine

QSO1_G_V → Quasi-sequence order with Grantham matrix for Valine

QSO1_G_W → Quasi-sequence order with Grantham matrix for Tryptophan

QSO1_G_Y → Quasi-sequence order with Grantham matrix for Tyrosine

QSO1_SC1 → Quasi-sequence order with Schneider matrix with lag 1

Note: ‘N’ prefix is added to the header for choosing N-terminal residues, ‘C’ prefix is added to the header for choosing C-terminal residues, ‘R’ prefix is added to the header for choosing rest method, ‘NC’ prefix is added to the header if NC-terminal has chosen, ‘RNC’ prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of ‘sn,’ where n is the number of splits, is added on choosing the split option.

QSO1_G1 → Quasi-sequence order with Grantham matrix with lag 1

Quasi-Sequence Order (order 2, alternate): 44 Descriptors

QSO2_SCA → Quasi-sequence order with Schneider matrix for Alanine

QSO2_SCC → Quasi-sequence order with Schneider matrix for Cysteine

QSO2_SCD → Quasi-sequence order with Schneider matrix for Aspartic acid

QSO2_SCE → Quasi-sequence order with Schneider matrix for Glutamic acid

QSO2_SCF → Quasi-sequence order with Schneider matrix for Phenylalanine

QSO2_SCG → Quasi-sequence order with Schneider matrix for Glycine

QSO2_SCH → Quasi-sequence order with Schneider matrix for Histidine

QSO2_SCI → Quasi-sequence order with Schneider matrix for Isoleucine

QSO2_SCK → Quasi-sequence order with Schneider matrix for Lysine

QSO2_SCL → Quasi-sequence order with Schneider matrix for Leucine

QSO2_SCM → Quasi-sequence order with Schneider matrix for Methionine

QSO2_SCN → Quasi-sequence order with Schneider matrix for Asparagine

QSO2_SCP → Quasi-sequence order with Schneider matrix for Proline

QSO2_SCQ → Quasi-sequence order with Schneider matrix for Glutamine

QSO2_SCR → Quasi-sequence order with Schneider matrix for Arginine

QSO2_SCS → Quasi-sequence order with Schneider matrix for Serine

QSO2_SCT → Quasi-sequence order with Schneider matrix for Threonine

QSO2_SCV → Quasi-sequence order with Schneider matrix for Valine

Note: ‘N’ prefix is added to the header for choosing N-terminal residues, ‘C’ prefix is added to the header for choosing C-terminal residues, ‘R’ prefix is added to the header for choosing rest method, ‘NC’ prefix is added to the header if NC-terminal has chosen, ‘RNC’ prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of ‘sn,’ where n is the number of splits, is added on choosing the split option.

QSO2_SCW → Quasi-sequence order with Schneider matrix for Tryptophan

QSO2_SCY → Quasi-sequence order with Schneider matrix for Tyrosine

QSO2_GA → Quasi-sequence order with Grantham matrix for Alanine

QSO2_GC → Quasi-sequence order with Grantham matrix for Cysteine

QSO2_GD → Quasi-sequence order with Grantham matrix for Aspartic acid

QSO2_GE → Quasi-sequence order with Grantham matrix for Glutamic acid

QSO2_GF → Quasi-sequence order with Grantham matrix for Phenylalanine

QSO2_GG → Quasi-sequence order with Grantham matrix for Glycine

QSO2_GH → Quasi-sequence order with Grantham matrix for Histidine

QSO2_GI → Quasi-sequence order with Grantham matrix for Isoleucine

QSO2_GK → Quasi-sequence order with Grantham matrix for Lysine

QSO2_GL → Quasi-sequence order with Grantham matrix for Leucine

QSO2_GM → Quasi-sequence order with Grantham matrix for Methionine

QSO2_GN → Quasi-sequence order with Grantham matrix for Asparagine

QSO2_GP → Quasi-sequence order with Grantham matrix for Proline

QSO2_GQ → Quasi-sequence order with Grantham matrix for Glutamine

QSO2_GR → Quasi-sequence order with Grantham matrix for Arginine

QSO2_GS → Quasi-sequence order with Grantham matrix for Serine

QSO2_GT → Quasi-sequence order with Grantham matrix for Threonine

QSO2_GV → Quasi-sequence order with Grantham matrix for Valine

QSO2_GW → Quasi-sequence order with Grantham matrix for Tryptophan

Note: ‘N’ prefix is added to the header for choosing N-terminal residues, ‘C’ prefix is added to the header for choosing C-terminal residues, ‘R’ prefix is added to the header for choosing rest method, ‘NC’ prefix is added to the header if NC-terminal has chosen, ‘RNC’ prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of ‘sn,’ where n is the number of splits, is added on choosing the split option.

QSO2_GY → Quasi-sequence order with Grantham matrix for Tyrosine

QSO2_SC1 → Quasi-sequence order with Schneider matrix with lag 1

QSO2_G1 → Quasi-sequence order with Grantham matrix with lag 1

QSO2_SC2 → Quasi-sequence order with Schneider matrix with lag 2

QSO2_G2 → Quasi-sequence order with Grantham matrix with lag 2

Quasi-Sequence Order (order 3, with gap of 2 residues): 46 Descriptors

QSO3_SCA → Quasi-sequence order with Schneider matrix for Alanine

QSO3_SCC → Quasi-sequence order with Schneider matrix for Cysteine

QSO3_SCD → Quasi-sequence order with Schneider matrix for Aspartic acid

QSO3_SCE → Quasi-sequence order with Schneider matrix for Glutamic acid

QSO3_SCF → Quasi-sequence order with Schneider matrix for Phenylalanine

QSO3_SCG → Quasi-sequence order with Schneider matrix for Glycine

QSO3_SCH → Quasi-sequence order with Schneider matrix for Histidine

QSO3_SCI → Quasi-sequence order with Schneider matrix for Isoleucine

QSO3_SCK → Quasi-sequence order with Schneider matrix for Lysine

QSO3_SCL → Quasi-sequence order with Schneider matrix for Leucine

QSO3_SCM → Quasi-sequence order with Schneider matrix for Methionine

QSO3_SCN → Quasi-sequence order with Schneider matrix for Asparagine

QSO3_SCP → Quasi-sequence order with Schneider matrix for Proline

QSO3_SCQ → Quasi-sequence order with Schneider matrix for Glutamine

QSO3_SCR → Quasi-sequence order with Schneider matrix for Arginine

QSO3_SCS → Quasi-sequence order with Schneider matrix for Serine

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

QSO3_SCT → Quasi-sequence order with Schneider matrix for Threonine

QSO3_SCV → Quasi-sequence order with Schneider matrix for Valine

QSO3_SCW → Quasi-sequence order with Schneider matrix for Tryptophan

QSO3_SCY → Quasi-sequence order with Schneider matrix for Tyrosine

QSO3_GA → Quasi-sequence order with Grantham matrix for Alanine

QSO3_GC → Quasi-sequence order with Grantham matrix for Cysteine

QSO3_GD → Quasi-sequence order with Grantham matrix for Aspartic acid

QSO3_GE → Quasi-sequence order with Grantham matrix for Glutamic acid

QSO3_GF → Quasi-sequence order with Grantham matrix for Phenylalanine

QSO3_GG → Quasi-sequence order with Grantham matrix for Glycine

QSO3_GH → Quasi-sequence order with Grantham matrix for Histidine

QSO3_GI → Quasi-sequence order with Grantham matrix for Isoleucine

QSO3_GK → Quasi-sequence order with Grantham matrix for Lysine

QSO3_GL → Quasi-sequence order with Grantham matrix for Leucine

QSO3_GM → Quasi-sequence order with Grantham matrix for Methionine

QSO3_GN → Quasi-sequence order with Grantham matrix for Asparagine

QSO3_GP → Quasi-sequence order with Grantham matrix for Proline

QSO3_GQ → Quasi-sequence order with Grantham matrix for Glutamine

QSO3_GR → Quasi-sequence order with Grantham matrix for Arginine

QSO3_GS → Quasi-sequence order with Grantham matrix for Serine

QSO3_GT → Quasi-sequence order with Grantham matrix for Threonine

QSO3_GV → Quasi-sequence order with Grantham matrix for Valine

QSO3_GW → Quasi-sequence order with Grantham matrix for Tryptophan

Note: ‘N’ prefix is added to the header for choosing N-terminal residues, ‘C’ prefix is added to the header for choosing C-terminal residues, ‘R’ prefix is added to the header for choosing rest method, ‘NC’ prefix is added to the header if NC-terminal has chosen, ‘RNC’ prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of ‘sn,’ where n is the number of splits, is added on choosing the split option.

QSO3_GY → Quasi-sequence order with Grantham matrix for Tyrosine

QSO3_SC1 → Quasi-sequence order with Schneider matrix with lag 1

QSO3_G1 → Quasi-sequence order with Grantham matrix with lag 1

QSO3_SC2 → Quasi-sequence order with Schneider matrix with lag 2

QSO3_G2 → Quasi-sequence order with Grantham matrix with lag 2

QSO3_SC3 → Quasi-sequence order with Schneider matrix with lag 3

QSO3_G3 → Quasi-sequence order with Grantham matrix with lag 3

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Sequence Order Coupling Number (order 1, traditional): 2 descriptors

SOC1_SC1 → Sequence order coupling number with Schneider matrix for lag 1

SOC1_G1 → Sequence order coupling number with Grantham matrix for lag 1

Sequence Order Coupling Number (order 2, alternate): 4 descriptors

SOC2_SC1 → Sequence order coupling number with Schneider matrix for lag 1

SOC2_G1 → Sequence order coupling number with Grantham matrix for lag 1

SOC2_SC2 → Sequence order coupling number with Schneider matrix for lag 2

SOC2_G2 → Sequence order coupling number with Grantham matrix for lag 2

Sequence Order Coupling Number (order 3, with gap of 2 residues): 6 descriptors

SOC3_SC1 → Sequence order coupling number with Schneider matrix for lag 1

SOC3_G1 → Sequence order coupling number with Grantham matrix for lag 1

SCO3_SC2 → Sequence order coupling number with Schneider matrix for lag 2

SOC3_G2 → Sequence order coupling number with Grantham matrix for lag 2

SOC3_SC3 → Sequence order coupling number with Schneider matrix for lag 3

SOC3_G3 → Sequence order coupling number with Grantham matrix for lag 3

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Binary Profile Descriptor

Binary profile of Amino acids : Total features 20* window/protein length (N)

A1 → Presence/Absence (1 or 0) for Alanine at position 1

C1 → Presence/Absence (1 or 0) for Cysteine at position 1

D1 → Presence/Absence (1 or 0) for Aspartic acid at position 1

E1 → Presence/Absence (1 or 0) for Glutamic acid at position 1

F1 → Presence/Absence (1 or 0) for Phenylalanine at position 1

A2 → Presence/Absence (1 or 0) for Alanine at position 2

C2 → Presence/Absence (1 or 0) for Cysteine at position 2

D2 → Presence/Absence (1 or 0) for Aspartic acid at position 2

E1 → Presence/Absence (1 or 0) for Glutamic acid at position 2

F2 → Presence/Absence (1 or 0) for Phenylalanine at position 2

An → Presence/Absence (1 or 0) for Alanine at position n

Cn → Presence/Absence (1 or 0) for Cysteine at position n

Dn → Presence/Absence (1 or 0) for Aspartic acid at position n

En→ Presence/Absence (1 or 0) for Glutamic acid at position n

Fn → Presence/Absence (1 or 0) for Phenylalanine at position n

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Dipeptide profile of amino acids : Total features 20*20*window/protein length(n)-q

AA1 → Presence/Absence (1 or 0) for Alanine-Alanine at position 1

AC1 → Presence/Absence (1 or 0) for Alanine-Cysteine at position 1

AD1 → Presence/Absence (1 or 0) for Alanine-Aspartic acid at position 1

AE1 → Presence/Absence (1 or 0) for Alanine-Glutamic acid at position 1

AA2 → Presence/Absence (1 or 0) for Alanine-Alanine at position 2

AC2 → Presence/Absence (1 or 0) for Alanine-Cysteine at position 2

AD2 → Presence/Absence (1 or 0) for Alanine-Aspartic acid at position 2

AE2 → Presence/Absence (1 or 0) for Alanine-Glutamic acid at position 2

AAn → Presence/Absence (1 or 0) for Alanine-Alanine at position n

ACn → Presence/Absence (1 or 0) for Alanine-Cysteine at position n

ADn → Presence/Absence (1 or 0) for Alanine-Aspartic acid at position n

AEn → Presence/Absence (1 or 0) for Alanine-Glutamic acid at position n

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Atom and Bond profile: Total features 5*total number of atoms (n) + 4*total number of bonds (m)

C1 → Presence/Absence (1 or 0) for Carbon atom at position 1

H1 → Presence/Absence (1 or 0) for Hydrogen atom at position 1

N1 → Presence/Absence (1 or 0) for Nitrogen atom at position 1

O1 → Presence/Absence (1 or 0) for Oxygen atom at position 1

S1 → Presence/Absence (1 or 0) for Sulphur atom at position 1

C2 → Presence/Absence (1 or 0) for Carbon atom at position 2

H2 → Presence/Absence (1 or 0) for Hydrogen atom at position 2

N2 → Presence/Absence (1 or 0) for Nitrogen atom at position 2

O2 → Presence/Absence (1 or 0) for Oxygen atom at position 2

S2 → Presence/Absence (1 or 0) for Sulphur atom at position 2

C_n → Presence/Absence (1 or 0) for Carbon atom at n^{th} position

H_n → Presence/Absence (1 or 0) for Hydrogen atom at n^{th} position

N_n → Presence/Absence (1 or 0) for Nitrogen atom at n^{th} position

O_n → Presence/Absence (1 or 0) for Oxygen atom at n^{th} position

S_n → Presence/Absence (1 or 0) for Sulphur atom at n^{th} position

SI1 → Presence/Absence (1 or 0) for single bond at position 1

DO1 → Presence/Absence (1 or 0) for double bond at position 1

CY1 → Presence/Absence (1 or 0) for cyclic ring at position 1

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

BE1 → Presence/Absence (1 or 0) for benzene ring at position 1

SI2 → Presence/Absence (1 or 0) for single bond at position 2

DO2 → Presence/Absence (1 or 0) for double bond at position 2

CY2 → Presence/Absence (1 or 0) for cyclic ring at position 2

BE2 → Presence/Absence (1 or 0) for benzene ring at position 2

SI_m → Presence/Absence (1 or 0) for single bond at m^{th} position

DO_m → Presence/Absence (1 or 0) for double bond at m^{th} position

CY_m → Presence/Absence (1 or 0) for cyclic ring at m^{th} position

BE_m → Presence/Absence (1 or 0) for benzene ring at m^{th} position

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Residue Properties Profile: Total features 25*window/protein length(n)

PC1 → Presence/Absence (1 or 0) for positively charged residues at position 1

NC1 → Presence/Absence (1 or 0) for positively charged residues at position 1

NE1 → Presence/Absence (1 or 0) for neutral charged residues at position 1

PO1 → Presence/Absence (1 or 0) for polar residues at position 1

NP1 → Presence/Absence (1 or 0) for non-polar residues at position 1

AL1 → Presence/Absence (1 or 0) for residues having aliphatic side chain at position 1

CY1 → Presence/Absence (1 or 0) for residues having cyclic side chain at position 1

AR1 → Presence/Absence (1 or 0) for aromatic residues at position 1

AC1 → Presence/Absence (1 or 0) for acidic residues at position 1

BS1 → Presence/Absence (1 or 0) for basic residues at position 1

NE1 → Presence/Absence (1 or 0) for neutral residues based on pH at position 1

HB1 → Presence/Absence (1 or 0) for hydrophobic residues at position 1

HL1 → Presence/Absence (1 or 0) for hydrophilic residues at position 1

NT1 → Presence/Absence (1 or 0) for neutral residues at position 1

HX1 → Presence/Absence (1 or 0) for hydroxylic residues at position 1

SC1 → Presence/Absence (1 or 0) for residues having sulphur content at position 1

SS_HE1 → Presence/Absence (1 or 0) for secondary structure (Helix) residues at position 1

SS_ST1 → Presence/Absence (1 or 0) for secondary structure (Strands) residues at position 1

SS_CO1 → Presence/Absence (1 or 0) for secondary structure (Coil) residues at position 1

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

SA_BU1→ Presence/Absence (1 or 0) for solvent accessibility (Buried) residues at position 1

SA_EX1 → Presence/Absence (1 or 0) for solvent accessibility (Exposed) residues at position 1

SA_IN1 → Presence/Absence (1 or 0) for solvent accessibility (Intermediate) residues at position 1

TN1 → Presence/Absence (1 or 0) for tiny residues at position 1

SM1 → Presence/Absence (1 or 0) for small residues at position 1

LR1 → Presence/Absence (1 or 0) for large residues at position 1

PC2 → Presence/Absence (1 or 0) for positively charged residues at position 2

NC2 → Presence/Absence (1 or 0) for positively charged residues at position 2

NE2 → Presence/Absence (1 or 0) for neutral charged residues at position 2

PO2 → Presence/Absence (1 or 0) for polar residues at position 2

NP2 → Presence/Absence (1 or 0) for non-polar residues at position 2

AL2 → Presence/Absence (1 or 0) for residues having aliphatic side chain at position 2

CY2 → Presence/Absence (1 or 0) for residues having cyclic side chain at position 2

AR2 → Presence/Absence (1 or 0) for aromatic residues at position 2

AC2 → Presence/Absence (1 or 0) for acidic residues at position 2

BS2 → Presence/Absence (1 or 0) for basic residues at position 2

NE2 → Presence/Absence (1 or 0) for neutral residues based on pH at position 2

HB2 → Presence/Absence (1 or 0) for hydrophobic residues at position 2

HL2 → Presence/Absence (1 or 0) for hydrophilic residues at position 2

NT2 → Presence/Absence (1 or 0) for neutral residues at position 2

HX2 → Presence/Absence (1 or 0) for hydroxylic residues at position 2

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

SC2 → Presence/Absence (1 or 0) for residues having sulphur content at position 2

SS_HE2 → Presence/Absence (1 or 0) for secondary structure (Helix) residues at position 2

SS_ST2 → Presence/Absence (1 or 0) for secondary structure (Strands) residues at position 2

SS_CO2 → Presence/Absence (1 or 0) for secondary structure (Coil) residues at position 2

SA_BU2→ Presence/Absence (1 or 0) for solvent accessibility (Buried) residues at position 2

SA_EX2 → Presence/Absence (1 or 0) for solvent accessibility (Exposed) residues at position 2

SA_IN2 → Presence/Absence (1 or 0) for solvent accessibility (Intermediate) residues at position 2

TN2 → Presence/Absence (1 or 0) for tiny residues at position 2

SM2 → Presence/Absence (1 or 0) for small residues at position 2

LR2 → Presence/Absence (1 or 0) for large residues at position 2

TNn → Presence/Absence (1 or 0) for tiny residues at position n

SMn → Presence/Absence (1 or 0) for small residues at position n

LRn → Presence/Absence (1 or 0) for large residues at position n

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

AA Index profile: Total features 553*window/protein length(n)

ANDN920101_1 → Presence/Absence (1 or 0) for ANDN920101 at position 1

KARS160122_1 → Presence/Absence (1 or 0) for KARS160122 at position 1

ANDN920101_2 → Presence/Absence (1 or 0) for ANDN920101 at position 2

KARS160122_2 → Presence/Absence (1 or 0) for KARS160122 at position 2

ANDN920101_n → Presence/Absence (1 or 0) for ANDN920101 at position n

---

KARS160122_2n→ Presence/Absence (1 or 0) for KARS160122 at position n

Note: 'N' prefix is added to the header for choosing N-terminal residues, 'C' prefix is added to the header for choosing C-terminal residues, 'R' prefix is added to the header for choosing rest method, 'NC' prefix is added to the header if NC-terminal has chosen, 'RNC' prefix is added to the header if Rest after removing NC-terminal has chosen, and the suffix of 'sn,' where n is the number of splits, is added on choosing the split option.

Chapter 11.0
Download & Links

This section provides the links for all the files and repositories linked to Pfeature.

11.1 Links

Pfeature Webserver : https://webs.iiitd.edu.in/raghava/pfeature/
Python Library : https://github.com/raghavagps/Pfeature/tree/master/PyLib
Standalone Version : https://github.com/raghavagps/Pfeature/tree/master/Standalone
Python Scripts : https://github.com/raghavagps/Pfeature/tree/master/scripts
GitHub : https://github.com/raghavagps/Pfeature

11.2 Documentation

- **Pfeature Manual**
  - https://webs.iiitd.edu.in/raghava/pfeature/Pfeature_Manual.pdf

11.3 Descriptors

- **Function Table**
  - https://github.com/raghavagps/Pfeature/blob/master/PyLib/Functions_Tables.pdf
- **Dimension of each feature**
  - https://github.com/raghavagps/Pfeature/blob/master/Feature_Description.pdf
- **Descriptor Headers**
  - https://github.com/raghavagps/Pfeature/blob/master/Descriptor_headers.pdf