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

Pfeature Manual
Raghava’s Group

Input Dataset

- PDB IDs
- Protein/Peptide Sequences
- UniProt IDs

Pfeature
[Web Server, Python Library Standalone]

Feature Generation Modules
- Composition
- Binary Profiles
- Patterns
- Structure
- Evolutionary Information

Model Building

Calculated Descriptors as Input to build model

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

One of the major challenges in the field of bioinformatics is to identify structural and functional properties of a protein from its primary sequence. In the past, a number of methods have been developed to annotate a protein at the sequence level as well as at residue level. Protein level annotation includes identification of a protein having a specific function, such as 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 annotate a protein, one needs 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 needs to compute features of a 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 provide a wide range of options to compute a large number of protein features, there are still a number of features which are not included in any existing methods, particularly features required for annotating proteins at the residue level. Residue level annotation means that you want to predict the 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 computes 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 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

1. Overview .................................................................................................................. 1  
   1.1. What is Pfeature? ................................................................................................. 1  
   1.2. Who uses Pfeature? ......................................................................................... 2  
   1.3. Objective .......................................................................................................... 2  
   1.4. Overview of Features .................................................................................... 3  

2. Getting Started with Pfeature .................................................................................. 5  
   2.1. Installation of Library ..................................................................................... 5  
   2.2. Webserver Implementation ......................................................................... 12  
   2.3. Usage of Standalone ..................................................................................... 13  

3. Composition Based Features ............................................................................... 16  
   3.1. Simple ............................................................................................................ 16  
      3.1.1. Amino Acids ............................................................................................. 16  
      3.1.2. Dipeptide .................................................................................................. 17  
      3.1.3. Tripeptide ................................................................................................. 18  
      3.1.4. Atom & Bond .......................................................................................... 18  
   3.2. Physico-Chemical Properties ....................................................................... 19  
      3.2.1. Standard Physico-Chemical Properties ................................................ 20  
      3.2.2. Amino Acid Index (AAindex) .................................................................. 20  
      3.2.3. Advanced Properties ............................................................................. 21  
      3.2.4. Structural Properties ............................................................................. 21  
   3.3. Repeats & Distribution ..................................................................................... 21  
      3.3.1. Residue Repeats ...................................................................................... 21  
      3.3.2. Property Repeats ..................................................................................... 23  
      3.3.3. Distance distribution of Residues ............................................................ 24  
   3.4. Shannon Entropy .............................................................................................. 25  
      3.4.1. Protein level .............................................................................................. 25  
      3.4.2. Residue Level ........................................................................................... 25  
      3.4.3. Properties Level ....................................................................................... 26  
   3.5. Miscellaneous .................................................................................................. 26  
      3.5.1. Autocorrelation ......................................................................................... 26  
      3.5.2. Conjoint Triad Descriptors (CTD) ............................................................. 27
3.5.3. Composition enhanced Transition and Distribution (CeTD) ......................... 28
3.5.4. Pseudo Amino Acid Composition (PAAC) ................................................. 28
3.5.5. Amphiphilic Pseudo Amino Acid Composition (APAAC) ......................... 29
3.5.6. Quasi-Sequence Order (QSO) ................................................................. 29
3.5.7. Sequence Order Coupling Number (SOCN) .............................................. 30

4. Binary Profiles Based Features ................................................................. 31
   4.1. Amino Acids ....................................................................................... 31
   4.2. Dipeptides ......................................................................................... 32
   4.3. Atom & Bond ..................................................................................... 32
   4.4. Residue Properties ........................................................................... 33
   4.5. AA Index ........................................................................................... 33

5. Evolutionary Information Based Features ................................................. 34
   5.1. Generation of PSSM .......................................................................... 34
   5.2. Normalization of PSSM ..................................................................... 34
   5.3. Composition of PSSM ....................................................................... 35
   5.4. Profile of PSSM ................................................................................ 35

6. Structure Based Features ........................................................................... 36
   6.1. Fingerprints ......................................................................................... 36
   6.2. SMILES ............................................................................................... 36
   6.3. Surface Accessibility ......................................................................... 37
   6.4. Secondary Structure .......................................................................... 37

7. Features Using Patterns ............................................................................ 38
   7.1. Binary Profiles .................................................................................... 38
   7.2. PSSM Profile ...................................................................................... 38
   7.3. Standard Physicochemical Properties ............................................... 38
   7.4. AA Index ............................................................................................ 39
   7.5. Universal ............................................................................................. 39

8. Portion of a Sequence ............................................................................... 40
   8.1. Whole sequence .................................................................................. 40
   8.2. N-terminal .......................................................................................... 40
   8.3. C-terminal .......................................................................................... 40
   8.4. Splits .................................................................................................. 41
8.5. Rest ................................................................................................................................. 41

9. Complete List of Features ......................................................................................... 42

10. Feature Headers ...................................................................................................... 47

11. Download & Links .................................................................................................. 93
    11.1. Links .................................................................................................................... 93
    11.2. Documentation ................................................................................................... 93
    11.3. Descriptors ......................................................................................................... 93
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.
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.

<table>
<thead>
<tr>
<th>COMPOSITION BASED DESCRIPTORS</th>
<th>Type of Composition</th>
<th>Vector Size</th>
<th>Type of Composition</th>
<th>Vector Size</th>
<th>Type of Composition</th>
<th>Vector Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amino acid</td>
<td>20</td>
<td>Shannon Entropy</td>
<td>1</td>
<td>Composition enhanced Transition Distribution</td>
<td>189</td>
<td></td>
</tr>
<tr>
<td>Dipeptide</td>
<td>400</td>
<td>Atom &amp; Bond</td>
<td>9</td>
<td>Shannon Entropy at residue level</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>Tripeptide</td>
<td>8000</td>
<td>Physicochemical</td>
<td>30</td>
<td>Conjoint Triad Calculation</td>
<td>343</td>
<td></td>
</tr>
<tr>
<td>Atom</td>
<td>5</td>
<td>Pseudo Amino-acid</td>
<td>20 + λ</td>
<td>Shannon Entropy of Property</td>
<td>24</td>
<td></td>
</tr>
<tr>
<td>Bond</td>
<td>4</td>
<td>Distance Distribution</td>
<td>20</td>
<td>Amphiphilic Pseudo Amino-acid</td>
<td>18 + λ*3</td>
<td></td>
</tr>
<tr>
<td>Autocorrelation</td>
<td>3*γ</td>
<td>Quasi-Sequence Order</td>
<td>40 + (λ*2)</td>
<td>Repetitive Residue Information</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>AAIndex</td>
<td>γ</td>
<td>Sequence Order Coupling</td>
<td>λ*2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BINARY PROFILE BASED DESCRIPTORS</th>
<th>Type of Composition</th>
<th>Vector Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amino acid</td>
<td>20*L</td>
<td></td>
</tr>
<tr>
<td>Dipeptide</td>
<td>400*L</td>
<td></td>
</tr>
<tr>
<td>Atom</td>
<td>5*η</td>
<td></td>
</tr>
<tr>
<td>Bond</td>
<td>4*ε</td>
<td></td>
</tr>
<tr>
<td>AAIndex</td>
<td>553*L</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>EVOLUTIONARY INFORMATION BASED DESCRIPTORS (PSSM Profiles)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Generation</td>
<td>L X 21</td>
<td>Normalization</td>
</tr>
<tr>
<td>Profile</td>
<td>L X 21</td>
<td></td>
</tr>
</tbody>
</table>

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STRUCTURE</strong></td>
</tr>
<tr>
<td>Fingerprints</td>
</tr>
<tr>
<td>Secondary Structure</td>
</tr>
</tbody>
</table>

# L: length of protein; \( \lambda \): The number depends upon the choice of maxlag; \( \gamma \): Number depends upon the choice of amino acid indices; \( \eta \): Number of atoms; \( \varepsilon \): Number of bonds
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**

```python
>>> 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")
```

![Example output]

```python
38.46, 0.00, 0.00, 3.85, 0.00, 3.85, 0.00, 0.00, 0.00, 0.00, 15.38, 7.69, 0.00, 7.69, 7.69, 3.85, 0.00, 0.00, 7.69, 3.85, 0.00
19.23, 3.85, 3.85, 0.00, 3.85, 7.69, 7.69, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 3.85, 11.54, 3.85, 0.00, 3.85, 0.00, 3.85, 3.85, 0.00
7.69, 3.85, 3.85, 15.38, 7.69, 7.69, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 3.85, 0.00, 0.00, 0.00, 3.85, 0.00, 11.54, 0.00, 0.00, 3.85, 0.00
7.69, 0.00, 0.00, 7.69, 0.00, 26.92, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 7.69, 11.54, 0.00, 7.69, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00
0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00
12.5, 0.00, 0.00, 3.85, 3.47, 0.17, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00
11.11, 0.00, 0.00, 0.00, 0.00, 7.41, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00
14.29, 7.14, 3.57, 0.00, 7.14, 10.71, 0.00, 0.00, 3.57, 10.71, 10.71, 7.14, 10.71, 0.00, 0.00, 7.14, 3.57, 3.57, 3.57, 0.00, 0.00, 0.00
3.45, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00
4.17, 0.16, 0.67, 8.33, 0.00, 8.33, 4.17, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00
```

Table 2.1: Usage of functions involved in Pfeature python library

<table>
<thead>
<tr>
<th>Name of the Function</th>
<th>Function call</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Composition Based Features</strong></td>
<td></td>
</tr>
<tr>
<td>Amino-acid composition</td>
<td>aac_wp(inputfile, outputfile)</td>
</tr>
<tr>
<td>Amino-acid Composition of N-Terminal</td>
<td>aac_nt(inputfile, outputfile, number)</td>
</tr>
<tr>
<td>Amino-acid Composition of C-Terminal</td>
<td>aac_ct(inputfile, outputfile, number)</td>
</tr>
<tr>
<td>Amino-acid Composition of Rest</td>
<td>aac_rt(inputfile, outputfile, number1, number2)</td>
</tr>
<tr>
<td>Amino-acid Composition of N- and C-Terminal</td>
<td>aac_nct(inputfile, outputfile, number)</td>
</tr>
<tr>
<td>Function</td>
<td>Command</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>Amino-acid Composition of split</td>
<td>aac_st(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Dipeptide Composition</td>
<td>dpc_wp(inputfile,outputfile,order)</td>
</tr>
<tr>
<td>Dipeptide Composition of N-Terminal</td>
<td>dpc_nt(inputfile,outputfile,number,order)</td>
</tr>
<tr>
<td>Dipeptide Composition of C-Terminal</td>
<td>dpc_ct(inputfile,outputfile,number,order)</td>
</tr>
<tr>
<td>Dipeptide Composition of Rest</td>
<td>dpc_rt(inputfile,outputfile,number1,number2,order)</td>
</tr>
<tr>
<td>Dipeptide Composition of N- and C-Terminal</td>
<td>dpc_nct(inputfile,outputfile,number,order)</td>
</tr>
<tr>
<td>Dipeptide Composition of split</td>
<td>dpc_st(inputfile,outputfile,lambda,number_of_splits)</td>
</tr>
<tr>
<td>Tripeptide Composition</td>
<td>tpc_wp(inputfile,outputfile)</td>
</tr>
<tr>
<td>Tripeptide Composition of N-Terminal</td>
<td>tpc_nt(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Tripeptide Composition of C-Terminal</td>
<td>tpc_ct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Tripeptide Composition of Rest</td>
<td>tpc Rt(inputfile,outputfile,number1,number2)</td>
</tr>
<tr>
<td>Tripeptide Composition of N- and C-Terminal</td>
<td>tpc_nct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Tripeptide Composition of split</td>
<td>tpc_st(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Atom Composition</td>
<td>atc_wp(inputfile,outputfile)</td>
</tr>
<tr>
<td>Atom Composition of N-Terminal</td>
<td>atc_nt(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Atom Composition of C-Terminal</td>
<td>atc_ct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Atom Composition of Rest</td>
<td>atc_rt(inputfile,outputfile,number1,number2)</td>
</tr>
<tr>
<td>Atom Composition of N- and C-Terminal</td>
<td>atc_nct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Atom Composition of split</td>
<td>atc_st(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Bond Composition</td>
<td>btc_wp(inputfile,outputfile)</td>
</tr>
<tr>
<td>Bond Composition of N-Terminal</td>
<td>btc_nt(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Bond Composition of C-Terminal</td>
<td>btc_ct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Bond Composition of Rest</td>
<td>btc_rt(inputfile,outputfile,number1,number2)</td>
</tr>
<tr>
<td>Bond Composition of N- and C-Terminal</td>
<td>btc_nct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Bond Composition of split</td>
<td>btc_st(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Physico-Chemical Properties Composition</td>
<td>pcp_wp(inputfile,outputfile)</td>
</tr>
<tr>
<td>Physico-Chemical Properties Composition of N-Terminal</td>
<td>pcp_nt(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Physico-Chemical Properties Composition of C-Terminal</td>
<td>pcp_ct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Physico-Chemical Properties Composition of Rest</td>
<td>pcp_rt(inputfile,outputfile,number1,number2)</td>
</tr>
<tr>
<td>Physico-Chemical Properties Composition of N- and C-Terminal</td>
<td>pcp_nct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Physico-Chemical Properties Composition of split</td>
<td>pcp_st(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Amino-acid index Composition</td>
<td>aai_wp(inputfile,outputfile)</td>
</tr>
<tr>
<td>Amino-acid index Composition of N-Terminal</td>
<td>aai_nt(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Amino-acid index Composition of C-Terminal</td>
<td>aai_ct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Amino-acid index Composition of Rest</td>
<td>aai_rt(inputfile,outputfile,number1,number2)</td>
</tr>
<tr>
<td>Function Description</td>
<td>Command</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------</td>
</tr>
<tr>
<td>Amino-acid index Composition of N- and C-Terminal</td>
<td>aai_nct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Amino-acid index Composition of split</td>
<td>aai_st(inputfile,outputfile ,number)</td>
</tr>
<tr>
<td>Repetitive Residue Information</td>
<td>rri_wp(inputfile,outputfile)</td>
</tr>
<tr>
<td>Repetitive Residue Information of N-Terminal</td>
<td>rri_nt(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Repetitive Residue Information of C-Terminal</td>
<td>rri_ct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Repetitive Residue Information of Rest</td>
<td>rri_rt(inputfile,outputfile,number1,number2)</td>
</tr>
<tr>
<td>Repetitive Residue Information of N- and C-Terminal</td>
<td>rri_nct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Repetitive Residue Information of split</td>
<td>rri_st(inputfile,outputfile ,number)</td>
</tr>
<tr>
<td>Distance Distribution of Residues</td>
<td>ddr_wp(inputfile,outputfile)</td>
</tr>
<tr>
<td>Distance Distribution of Residues for N-Terminal</td>
<td>ddr_nt(inputfile,outputfile)</td>
</tr>
<tr>
<td>Distance Distribution of Residues for C-Terminal</td>
<td>ddr_ct(inputfile,outputfile)</td>
</tr>
<tr>
<td>Distance Distribution of Residues for Rest</td>
<td>ddr_rt(inputfile,outputfile)</td>
</tr>
<tr>
<td>Distance Distribution of Residues for N- and C-Terminal</td>
<td>ddr_nct(inputfile,outputfile)</td>
</tr>
<tr>
<td>Distance Distribution of Residues for split</td>
<td>ddr_st(inputfile,outputfile, number of splits)</td>
</tr>
<tr>
<td>Physico-chemical properties repeat Composition</td>
<td>pri_wp(inputfile,outputfile)</td>
</tr>
<tr>
<td>Physico-chemical properties repeat Composition of N-Terminal</td>
<td>pri_nt(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Physico-chemical properties repeat Composition of C-Terminal</td>
<td>pri_ct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Physico-chemical properties repeat Composition of Rest</td>
<td>pri_rt(inputfile,outputfile,number1,number2)</td>
</tr>
<tr>
<td>Physico-chemical properties repeat Composition of N- and C-Terminal</td>
<td>pri_nct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Physico-chemical properties repeat Composition of Split</td>
<td>pri_st(inputfile,outputfile ,number)</td>
</tr>
<tr>
<td>Shannon Entropy</td>
<td>sep_wp(inputfile,outputfile)</td>
</tr>
<tr>
<td>Shannon Entropy of N-Terminal</td>
<td>sep_nt(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Shannon Entropy of C-Terminal</td>
<td>sep_ct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Shannon Entropy of Rest</td>
<td>sep_rt(inputfile,outputfile,number1,number2)</td>
</tr>
<tr>
<td>Shannon Entropy of N- and C-Terminal</td>
<td>sep_nct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Shannon Entropy of Split</td>
<td>sep_st(inputfile,outputfile ,number)</td>
</tr>
<tr>
<td>Shannon Entropy of Residue Level</td>
<td>ser_wp(inputfile,outputfile)</td>
</tr>
<tr>
<td>Shannon Entropy of Residue Level for N-Terminal</td>
<td>ser_nt(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Shannon Entropy of Residue Level for C-Terminal</td>
<td>ser_ct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Shannon Entropy of Residue Level for Rest</td>
<td>ser_rt(inputfile,outputfile,number1,number2)</td>
</tr>
<tr>
<td>Function</td>
<td>Command</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Shannon Entropy of Residue Level for N-Terminal</td>
<td>ser_nct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Shannon Entropy of Residue Level for Split</td>
<td>ser_st(inputfile,outputfile ,number)</td>
</tr>
<tr>
<td>Shannon Entropy of Physicochemical Property</td>
<td>spc_wp(inputfile,outputfile)</td>
</tr>
<tr>
<td>Shannon Entropy of Physicochemical Property for N-Terminal</td>
<td>spc_nt(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Shannon Entropy of Physicochemical Property for C-Terminal</td>
<td>spc_ct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Shannon Entropy of Physicochemical Property for Rest</td>
<td>spc_rt(inputfile,outputfile,number1,number2)</td>
</tr>
<tr>
<td>Shannon Entropy of Physicochemical Property for N- and C-Terminal</td>
<td>spc_nct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Shannon Entropy of Physicochemical Property for Split</td>
<td>spc_st(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Autocorrelation</td>
<td>acr_wp(inputfile,outputfile,lag)</td>
</tr>
<tr>
<td>Autocorrelation of N-Terminal</td>
<td>act_nt(inputfile,outputfile,number,lag)</td>
</tr>
<tr>
<td>Autocorrelation of C-Terminal</td>
<td>acr_ct(inputfile,outputfile,number,lag)</td>
</tr>
<tr>
<td>Autocorrelation of Rest</td>
<td>acr_rt(inputfile,outputfile,number1,number2,lag)</td>
</tr>
<tr>
<td>Autocorrelation of N- and C-Terminal</td>
<td>acr_nct(inputfile,outputfile,number,lag)</td>
</tr>
<tr>
<td>Autocorrelation of Split</td>
<td>acr_st(inputfile,outputfile,number,lag)</td>
</tr>
<tr>
<td>Conjoint Triad Calculation</td>
<td>ctc_wp(inputfile,outputfile)</td>
</tr>
<tr>
<td>Conjoint Triad Calculation for N-Terminal</td>
<td>ctc_nt(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Conjoint Triad Calculation for C-Terminal</td>
<td>ctc_ct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Conjoint Triad Calculation for Rest</td>
<td>ctc_rt(inputfile,outputfile,number1,number2)</td>
</tr>
<tr>
<td>Conjoint Triad Calculation for N- and C-Terminal</td>
<td>ctc_nct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Conjoint Triad Calculation for Split</td>
<td>ctc_st(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Composition enhanced Transition Distribution</td>
<td>ctd_wp(inputfile,outputfile)</td>
</tr>
<tr>
<td>Composition enhanced Transition Distribution of N-Terminal</td>
<td>ctd_nt(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Composition enhanced Transition Distribution of C-Terminal</td>
<td>ctc_ct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Composition enhanced Transition Distribution of Rest</td>
<td>ctd_rt(inputfile,outputfile,number1,number2)</td>
</tr>
<tr>
<td>Composition enhanced Transition Distribution of N- and C-Terminal</td>
<td>ctd_nct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Composition enhanced Transition Distribution</td>
<td>ctd_st(inputfile,outputfile ,number)</td>
</tr>
<tr>
<td>Pseudo Amino-acid Composition</td>
<td>paac_wp(inputfile,outputfile,lamda,weight )</td>
</tr>
<tr>
<td>Pseudo Amino-acid Composition of N-Terminal</td>
<td>paac_nt(inputfile,outputfile,number,lamda,weight)</td>
</tr>
<tr>
<td>Pseudo Amino-acid Composition of C-Terminal</td>
<td>paac_ct(inputfile,outputfile,number,lamda,weight)</td>
</tr>
<tr>
<td>Function</td>
<td>Command</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Pseudo Amino-acid Composition of Rest</td>
<td>paac_rt(inputfile,outputfile,number1,number2,lamda,weight)</td>
</tr>
<tr>
<td>Pseudo Amino-acid Composition of N- and C-Terminal</td>
<td>paac_nct(inputfile,outputfile,number,lamda,weight)</td>
</tr>
<tr>
<td>Pseudo Amino-acid Composition of split</td>
<td>paac_st(inputfile,outputfile,lamda,number)</td>
</tr>
<tr>
<td>Amphiphilic Pseudo Amino-acid Composition</td>
<td>apaac_wp(inputfile,outputfile,lamda,weight)</td>
</tr>
<tr>
<td>Amphiphilic Pseudo Amino-acid Composition of N- Terminal</td>
<td>apaac_nt(inputfile,outputfile,number,lamda,weight)</td>
</tr>
<tr>
<td>Amphiphilic Pseudo Amino-acid Composition of C- Terminal</td>
<td>apaac_ct(inputfile,outputfile,number,lamda,weight)</td>
</tr>
<tr>
<td>Amphiphilic Pseudo Amino-acid Composition of Rest</td>
<td>apaac_rt(inputfile,outputfile,number1,number2,lamda,weight)</td>
</tr>
<tr>
<td>Amphiphilic Pseudo Amino-acid Composition of N- and C-Terminal</td>
<td>apaac_nct(inputfile,outputfile,number,lamda,weight)</td>
</tr>
<tr>
<td>Amphiphilic Pseudo Amino-acid Composition of split</td>
<td>apaac_st(inputfile,outputfile,number1,number2,lamda,weight)</td>
</tr>
<tr>
<td>Quasi-Sequence Order</td>
<td>qos_wp(inputfile,outputfile,gap,weight)</td>
</tr>
<tr>
<td>Quasi-Sequence Order of N-Terminal</td>
<td>qos_nt(inputfile,outputfile,number,gap,weight)</td>
</tr>
<tr>
<td>Quasi-Sequence Order of C-Terminal</td>
<td>qos_ct(inputfile,outputfile,number,gap,weight)</td>
</tr>
<tr>
<td>Quasi-Sequence Order of Rest</td>
<td>qos_rt(inputfile,outputfile,number1,number2,gap,weight)</td>
</tr>
<tr>
<td>Quasi-Sequence Order of N- and C-Terminal</td>
<td>qos_nct(inputfile,outputfile,number,gap,weight)</td>
</tr>
<tr>
<td>Quasi-Sequence Order of split</td>
<td>qos_st(inputfile,outputfile,Number_of_splits,gap,weight)</td>
</tr>
<tr>
<td>Sequence Order Coupling</td>
<td>soc_wp(inputfile,outputfile,gap)</td>
</tr>
<tr>
<td>Sequence Order Coupling for N-Terminal</td>
<td>soc_nt(inputfile,outputfile,number,gap)</td>
</tr>
<tr>
<td>Sequence Order Coupling for C-Terminal</td>
<td>soc_ct(inputfile,outputfile,number,gap)</td>
</tr>
<tr>
<td>Sequence Order Coupling for Rest</td>
<td>soc_rt(inputfile,outputfile,number1,number2,gap)</td>
</tr>
<tr>
<td>Sequence Order Coupling for N- and C-Terminal</td>
<td>soc_nct(inputfile,outputfile,number,gap)</td>
</tr>
<tr>
<td>Sequence Order Coupling of split</td>
<td>soc_split(inputfile,outputfile,Number_of_splits,gap)</td>
</tr>
</tbody>
</table>

### Binary Profile Based Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Profile of amino acid</td>
<td>aab_wp(inputfile,outputfile)</td>
</tr>
<tr>
<td>Binary Profile of N-Terminal amino acid</td>
<td>aab_nt(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Binary Profile of C-Terminal of amino acid</td>
<td>aab_ct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Binary Profile of rest of amino acid</td>
<td>aab_rt(inputfile,outputfile,number1,number2)</td>
</tr>
<tr>
<td>Binary Profile of N- and C-Terminal amino acid</td>
<td>aab_nct(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Binary Profile of split of amino acid</td>
<td>aab_st(inputfile,outputfile,number)</td>
</tr>
<tr>
<td>Binary Profile of Dipeptide</td>
<td>dpb_wp(inputfile,outputfile,order)</td>
</tr>
<tr>
<td>Dipeptide Binary Profile of N-Terminal</td>
<td>dpb_nt(inputfile,outputfile,number,order)</td>
</tr>
<tr>
<td>Dipeptide Binary Profile of C-Terminal</td>
<td>dpb_ct(inputfile,outputfile,number,order)</td>
</tr>
<tr>
<td>Feature</td>
<td>Function</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Dipeptide Binary Profile of Rest</td>
<td>dpb_rt(inputfile, outputfile, number1, number2, order)</td>
</tr>
<tr>
<td>Dipeptide Binary Profile of N- and C-Terminal</td>
<td>dbp_nct(inputfile, outputfile, number, order)</td>
</tr>
<tr>
<td>Dipeptide Binary Profile of Split</td>
<td>dpb_st(inputfile, outputfile, lag, Number_of_splits)</td>
</tr>
<tr>
<td>Atomic Binary Profile</td>
<td>atb_wp(inputfile, outputfile)</td>
</tr>
<tr>
<td>Atomic Binary Profile of N-Terminal</td>
<td>atb_nt(inputfile, outputfile, number)</td>
</tr>
<tr>
<td>Atomic Binary Profile of C-Terminal</td>
<td>atb_ct(inputfile, outputfile, number)</td>
</tr>
<tr>
<td>Atomic Binary Profile of rest</td>
<td>atb_rrt(inputfile, outputfile, number1, number2)</td>
</tr>
<tr>
<td>Atomic Binary Profile of N- and C-Terminal</td>
<td>atb_nct(inputfile, outputfile, number)</td>
</tr>
<tr>
<td>Atomic Binary Profile of split</td>
<td>atb_st(inputfile, outputfile, Number_of_splits)</td>
</tr>
<tr>
<td>Binary Profile of bond</td>
<td>btb_wp(inputfile, outputfile)</td>
</tr>
<tr>
<td>Bonds Binary Profile of N-Terminal</td>
<td>btb_nt(inputfile, outputfile, number)</td>
</tr>
<tr>
<td>Bonds Binary Profile of C-Terminal</td>
<td>btb_ct(inputfile, outputfile, number)</td>
</tr>
<tr>
<td>Bonds Binary Profile of rest</td>
<td>btb_rrt(inputfile, outputfile, number1, number2)</td>
</tr>
<tr>
<td>Bonds Binary Profile of N- and C-Terminal</td>
<td>btb_nct(inputfile, outputfile, number)</td>
</tr>
<tr>
<td>Bonds Binary Profile of split</td>
<td>btb_st(inputfile, outputfile, Number_of_splits)</td>
</tr>
<tr>
<td>Binary Profile of Physicochemical Property</td>
<td>pcb_wp(inputfile, outputfile)</td>
</tr>
<tr>
<td>Binary Profile of Physicochemical Property of N-Terminal</td>
<td>pcb_nt(inputfile, outputfile, number)</td>
</tr>
<tr>
<td>Binary Profile of Physicochemical Property of C-Terminal</td>
<td>pcb_ct(inputfile, outputfile, number)</td>
</tr>
<tr>
<td>Binary Profile of Physicochemical Property of Rest</td>
<td>pcb_rrt(inputfile, outputfile, number1, number2)</td>
</tr>
<tr>
<td>Binary Profile of Physicochemical Property of N- and C-Terminal</td>
<td>pcb_nct(inputfile, outputfile, number)</td>
</tr>
<tr>
<td>Binary Profile of Physicochemical Property of Split</td>
<td>pcb_st(inputfile, outputfile, Number_of_splits)</td>
</tr>
<tr>
<td>Binary Profile of AAIndex</td>
<td>aib_wp(inputfile, outputfile)</td>
</tr>
<tr>
<td>Binary Profile of AAIndex of N-Terminal</td>
<td>aib_nt(inputfile, outputfile, number)</td>
</tr>
<tr>
<td>Binary Profile of AAIndex of C-Terminal</td>
<td>aib_ct(inputfile, outputfile, number)</td>
</tr>
<tr>
<td>Binary Profile of AAIndex of Rest</td>
<td>aib_rrt(inputfile, outputfile, number1, number2)</td>
</tr>
<tr>
<td>Binary Profile of AAIndex of N- and C-Terminal</td>
<td>aib_nct(inputfile, outputfile, number)</td>
</tr>
<tr>
<td>Binary Profile of AAIndex of split</td>
<td>aib_st(inputfile, outputfile, Number_of_splits)</td>
</tr>
<tr>
<td><strong>PSSM Based Features</strong></td>
<td></td>
</tr>
<tr>
<td>PSSM composition</td>
<td>pssm_comp(inputfile, outputfile)</td>
</tr>
<tr>
<td>Normalization of PSSM profile using method 1</td>
<td>pssm_n1(inputfile, outputfile)</td>
</tr>
<tr>
<td>Normalization of PSSM profile using method 2</td>
<td>pssm_n2(inputfile, outputfile)</td>
</tr>
<tr>
<td>Normalization of PSSM profile using method 3</td>
<td>pssm_n3(inputfile, outputfile)</td>
</tr>
<tr>
<td>Normalization of PSSM profile using method 4</td>
<td>pssm_n4(inputfile, outputfile)</td>
</tr>
</tbody>
</table>
Pattern Based Features

<table>
<thead>
<tr>
<th>Feature</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Profile of generated pattern</td>
<td>pat_bin(inputfile,outputfile,window_length)</td>
</tr>
<tr>
<td>Physico-chemical properties composition of generated pattern</td>
<td>pat_pcp(inputfile,outputfile,window_length)</td>
</tr>
<tr>
<td>Pattern generated from sequences</td>
<td>pat_str(inputfile,outputfile,window_length)</td>
</tr>
<tr>
<td>Pattern generated from CSV file</td>
<td>pat_csv(inputfile,outputfile,window_length)</td>
</tr>
<tr>
<td>Amino acid index composition of generated pattern</td>
<td>pat_aai(inputfile,outputfile,window_length)</td>
</tr>
</tbody>
</table>

2.2 Webserver Implementation

Pfeature is available as webserver at URL: [https://webs.iiitd.edu.in/raghava/pfeature/](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](image)
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
```

![Figure 2.2: Complete usage of script pfeature_comp.py](image-url)
ii) python3 pfeature_bin.py -h

Figure 2.3: Complete usage of script pfeature_bin.py

```python
usage: pfeature_bin.py [-h] [-i INPUT] [-o OUTPUT] [-c C_TERMINAL] [-n N_TERMINAL]
[-nct NC_TERMINAL] [-rm REST_N] [-rc REST_C] [-s SPLIT] [-d LAG]

Please provide following arguments

optional arguments:
  -h, --help            show this help message and exit
  -i INPUT, -I INPUT    input INPUT
  -o OUTPUT, -O OUTPUT  output OUTPUT

Input: protein or peptide sequence in FASTA format or single sequence per line in single letter code

Output: File for saving results by default pfeature_result.csv

Job Type:
- AAB: Amino acid based binary profile
- DPP: Dipeptide based binary profile
- ATB: Atom based binary profile
- BTP: Bond based binary profile
- PCC: Physico-chemical properties based binary profile
- AIB: Amino-acid indices based binary profile
- ALLBIN: All binary profiles together except ATB and BTP

-n N_TERMINAL, -N N_TERMINAL    Length from N-terminal by default 0
-c C_TERMINAL, -C C_TERMINAL    Length from C-terminal by default 0
-nct NC_TERMINAL, -ncnt NC_TERMINAL    Length from N- and C-terminal, by default 0
-rm REST_N, -RM REST_N    Number of residues removed from N-terminal, by default 0
-rc REST_C, -RC REST_C    Number of residues removed from C-terminal, by default 0
-s SPLIT, -S SPLIT    Number of splits a sequence divided into, by default 0
-d LAG, -D LAG, -L LAG    This represents the order of gap, lag or dipeptide, by default 1
```

iii) python3 pfeature_pssm.py -h

Figure 2.4: Complete usage of script pfeature_pssm.py

```python
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
  -o OUTPUT, -O OUTPUT  output OUTPUT

Input: protein or peptide sequence in FASTA format or single sequence per line in single letter code

Output: File for saving results by default pssm_profile.csv

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 \(((x-min)/(max-min)) \times 100\) formula
N4: It normalizes pssm profile based on \((1/(1+e^{-x/100})\) formula

By default it is N0
```
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:
\[ AAC_i = \frac{R_i}{L} \]  \hspace{1cm} (1)

where \( AAC_i \) is amino acid composition of residue type \( i \); \( R_i \) and \( L \) number of residues of type \( i \) 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.](image)

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

\[ DPC^j_i = \frac{d^j_i}{L - j} \]  \hspace{1cm} (2)
Where $DPC_i^j$ is the fraction or composition of dipeptide of type $i$ for jth order. $D_i^j$ and $L$ are the number of dipeptides of type $i$ and length of a protein sequence, respectively. Here higher order dipeptide $D_i^j$ 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.

\[
TPC_i = T_i L - 2
\]

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.

\[
ATC_i = \frac{A_i}{N}
\]
\[ BTC_i = \frac{B_i}{N} \]  

(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.

<table>
<thead>
<tr>
<th>Atomic Composition</th>
<th>Bond Composition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Carbon Atom</td>
<td>Total Bonds</td>
</tr>
<tr>
<td>Hydrogen Atom</td>
<td>Hydrogen Bond</td>
</tr>
<tr>
<td>Nitrogen Atom</td>
<td>Single Bond</td>
</tr>
<tr>
<td>Oxygen Atom</td>
<td>Double Bond</td>
</tr>
<tr>
<td>Sulphur Atom</td>
<td></td>
</tr>
</tbody>
</table>

3.2 Physico-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

<table>
<thead>
<tr>
<th>Positively Charged</th>
<th>Aromaticity</th>
<th>Hydroxylic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negatively Charged</td>
<td>Acidity</td>
<td>Sulphur Content</td>
</tr>
<tr>
<td>Neutral Charge</td>
<td>Basicity</td>
<td>Tiny</td>
</tr>
<tr>
<td>Polarity in residues</td>
<td>Neutral (pH)</td>
<td>Small</td>
</tr>
<tr>
<td>Non-polarity in residues</td>
<td>Hydrophobicity</td>
<td>Large</td>
</tr>
<tr>
<td>Aliphaticity</td>
<td>Hydrophilicity</td>
<td></td>
</tr>
<tr>
<td>Cyclicity</td>
<td>Neutral towards water</td>
<td></td>
</tr>
</tbody>
</table>

We used the following formula to calculate these features

\[ PCP_i = \frac{P_i}{L} \]  \hspace{1cm} (6)

Where, \( PCP_i \) is Physico-chemical properties composition of type \( i \); \( P_i \) and \( L \) are sum of property of type \( i \), 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:

\[ AAIC_i = \frac{AAI_i}{L} \]  \hspace{1cm} (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.

3.3 Repeats & 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 “GARAGRARDEARTAG”; 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 “GAARGRGAARDEARDERTG” 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 “GRGRGAAAAARDERTG” 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.

\[
PRI_i = \frac{\sum_{j=1}^{N}(P_j)^2}{\sum_{j=1}^{N} P_j}
\]  

(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 functions, 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.

\[ \text{DDOR}_i = \frac{(R_{NT})^2 + \sum_{j=1}^{N}(R_j)^2 + (R_{CT})^2}{(L - F_i) + 1} \]  

(10)

Where, \( \text{DDOR}_i \) is distance distribution of residue type \( i \), \( N \) is total number of inter-residue distances for type \( i \).

- \( R_{NT} \) = 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.
3.4 Shannon Entropy

3.4.1 Protein Level

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

\[ H(X) = -\sum_{i=1}^{20} p_i \log_2 p_i \]  

(11)

Where, \( i \) is the amino acid in the sequence (i=A, C, D, \ldots , 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:

\[ p_i = \frac{c_i}{L} \]  

(12)

\[ H_i = -p_i \log_2 p_i \]  

(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:

\[
H_i = -p_i \log(p_i) - (1 - p_i) \log(1 - p_i)
\]

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.

### 3.5 Miscellaneous

#### 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**: \( dval \leq \min(L-1, 30) \) where \( L \) is the length of the sequence/ subsequence 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:-

\[ f_{\text{norm}} = \frac{f_i - \min(f_1, f_2, f_3 \ldots \ldots f_{343})}{\max(f_1, f_2, f_3 \ldots \ldots f_{343})} \]

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:

\[
\text{Composition} = \frac{\text{Frequency of same Residue} \times 100}{\text{Length of peptide sequence}} \quad (14)
\]

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

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Group 1</th>
<th>Group 2</th>
<th>Group 3</th>
</tr>
</thead>
</table>

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 H1(i) be hydrophobicity values for i = 1,2,3,......20, H2(i) be the hydrophilicity values for i = 1,2,3,......20, and M(i) be the side chain masses of the 20 natural amino acids. They are converted to the following quantities by a standard conversion:

\[
H_1(i) = \frac{H_1^0(i) - \frac{1}{20} \sum_{i=1}^{20} H_1^0}{\sqrt{\frac{1}{20} \sum_{i=1}^{20} [H_1^0(i) - \frac{1}{20} \sum_{i=1}^{20} H_1^0(i)]^2}} \quad (15)
\]
Where, \( H_1^o (i) \) and \( M^o(i) \) are normalized as \( H_1(i) \) 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, webservice 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:

\[
P_c = \frac{f_c}{\sum_{r=1}^{20} f_r + w \sum_{j=1}^{20} \tau_j} \quad (1 < c < 20) \quad (16(i))
\]

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

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:

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

where \( f_r \) 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:
\[ X_d = \frac{w \tau_d - 20}{\sum_{r=1}^{20} f_r + w \sum_{d=1}^{nlag} \tau_d} \quad d = 21, 22, \ldots, 30 + nlag \quad (18) \]

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:

\[ \tau_d = \sum_{i=1}^{N-d} (d_{i,i+d})^2 \quad d = 1, 2, 3, \ldots, nlag \quad (19) \]

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 \(nlag\).
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.

<table>
<thead>
<tr>
<th>Residue</th>
<th>Binary Profile</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
<tr>
<td>C</td>
<td>0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
<tr>
<td>D</td>
<td>0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
<tr>
<td>E</td>
<td>0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
<tr>
<td>F</td>
<td>0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
<tr>
<td>G</td>
<td>0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
<tr>
<td>H</td>
<td>0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
<tr>
<td>I</td>
<td>0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
<tr>
<td>K</td>
<td>0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
<tr>
<td>L</td>
<td>0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
<tr>
<td>M</td>
<td>0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
<tr>
<td>N</td>
<td>0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
<tr>
<td>P</td>
<td>0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
<tr>
<td>Q</td>
<td>0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0</td>
</tr>
<tr>
<td>R</td>
<td>0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0</td>
</tr>
<tr>
<td>S</td>
<td>0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0</td>
</tr>
<tr>
<td>T</td>
<td>0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0</td>
</tr>
<tr>
<td>V</td>
<td>0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0</td>
</tr>
<tr>
<td>W</td>
<td>0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0</td>
</tr>
<tr>
<td>Y</td>
<td>0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1</td>
</tr>
<tr>
<td>X</td>
<td>0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0</td>
</tr>
</tbody>
</table>

**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.

### 4.2 Dipeptides

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

![Figure 4.2: Binary profiles for few possible dipeptides](image)

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.

### 4.3 Atom & 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.
4.4 Residue 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.

4.5 AA 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.
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 $\frac{(num - min)}{(max - min)}$.
- **pssm_n3**: This is the third technique to normalize the matrix using the formula $\frac{(num - min)}{(max - min)} \times 100$.
- **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.

![Diagram of sub-modules for Structure]

**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](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.
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.

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.
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

<table>
<thead>
<tr>
<th>Type of Features</th>
<th>Description</th>
<th>Features</th>
<th>Dimension Vectors</th>
<th>Supported By</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>COMPOSITION: SIMPLE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AAC</td>
<td>Amino acid Composition</td>
<td>Whole</td>
<td>20</td>
<td>[a,b,c,d,e]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N-Terminal</td>
<td>20</td>
<td>[a]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C-Terminal</td>
<td>20</td>
<td>[a]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rest</td>
<td>20</td>
<td>[a]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Split</td>
<td>20*N</td>
<td>[a]</td>
</tr>
<tr>
<td>DPC</td>
<td>Dipeptide Composition</td>
<td>Whole</td>
<td>400</td>
<td>[a,b,c,d,e]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N-Terminal</td>
<td>400</td>
<td>[a]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C-Terminal</td>
<td>400</td>
<td>[a]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rest</td>
<td>400</td>
<td>[a]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Split</td>
<td>400*N</td>
<td>[a]</td>
</tr>
<tr>
<td>TPC</td>
<td>Tripeptide Composition</td>
<td>Whole</td>
<td>8000</td>
<td>[a,b,c,d]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N-Terminal</td>
<td>8000</td>
<td>[a]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C-Terminal</td>
<td>8000</td>
<td>[a]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rest</td>
<td>8000</td>
<td>[a]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Split</td>
<td>8000*N</td>
<td>[a]</td>
</tr>
<tr>
<td>ABC</td>
<td>Atom and Bond Composition</td>
<td>Whole</td>
<td>9</td>
<td>[a]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N-Terminal</td>
<td>9</td>
<td>[a]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>C-Terminal</td>
<td>9</td>
<td>[a]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rest</td>
<td>9</td>
<td>[a]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Split</td>
<td>9*N</td>
<td>[a]</td>
</tr>
<tr>
<td><strong>COMPOSITION: PHYSICO-CHEMICAL PROPERTIES</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCP</td>
<td>Physico-Chemical properties composition</td>
<td>Whole</td>
<td>19</td>
<td>[a,b,c,d,e]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>N-Terminal</td>
<td>19</td>
<td>[a]</td>
</tr>
<tr>
<td></td>
<td>Amino Acid Index Composition</td>
<td>Advanced Physico-Chemical properties composition</td>
<td>Structural Physico-Chemical properties composition</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------</td>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>AAI</td>
<td>C-Terminal 19 [a]</td>
<td>Whole 5 [a,b,c,d,e]</td>
<td>Whole 6 [a,b,c,d,e]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rest 19 [a]</td>
<td>N-Terminal 5 [a]</td>
<td>N-Terminal 6 [a]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Split 19*N [a]</td>
<td>C-Terminal 5 [a]</td>
<td>C-Terminal 6 [a]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Rest 5 [a]</td>
<td>Rest 6 [a]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Split 5*N [a]</td>
<td>Split 6 [a]</td>
<td></td>
</tr>
<tr>
<td>PCP_adv</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PCP_str</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**COMPOSITION: REPEATS & DISTRIBUTION**

<table>
<thead>
<tr>
<th></th>
<th>Repetitive Residue Information</th>
<th>Repeat of Physico-chemical Properties</th>
<th>Distance Distribution of Residues</th>
</tr>
</thead>
<tbody>
<tr>
<td>RRI</td>
<td>Whole 20 [a]</td>
<td>Whole 19 [a]</td>
<td>Whole 20 [a]</td>
</tr>
<tr>
<td></td>
<td>N-Terminal 20 [a]</td>
<td>N-Terminal 19 [a]</td>
<td>N-Terminal 20 [a]</td>
</tr>
<tr>
<td></td>
<td>C-Terminal 20 [a]</td>
<td>C-Terminal 19 [a]</td>
<td>C-Terminal 20 [a]</td>
</tr>
<tr>
<td></td>
<td>Rest 20 [a]</td>
<td>Rest 19 [a]</td>
<td>Rest 20 [a]</td>
</tr>
<tr>
<td></td>
<td>Split 20*N [a]</td>
<td>Split 19*N [a]</td>
<td>Split 20*N [a]</td>
</tr>
<tr>
<td>COMPOSITION: SHANNON ENTROPY</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SEP</strong></td>
<td>Shannon Entropy at Protein Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Whole</td>
<td>1 [a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-Terminal</td>
<td>1 [a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-Terminal</td>
<td>1 [a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rest</td>
<td>1 [a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Split</td>
<td>1*N [a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SER</strong></td>
<td>Shannon Entropy at Residue Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Whole</td>
<td>20 [a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-Terminal</td>
<td>20 [a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-Terminal</td>
<td>20 [a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rest</td>
<td>20 [a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Split</td>
<td>20*N [a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>SPC</strong></td>
<td>Shannon Entropy at Property Level</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Whole</td>
<td>19 [a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N-Terminal</td>
<td>19 [a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-Terminal</td>
<td>19 [a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rest</td>
<td>19 [a]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Split</td>
<td>19*N [a]</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>COMPOSITION: MISCELLANEOUS</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ACR</strong></td>
<td>Autocorrelation Descriptors</td>
</tr>
<tr>
<td>Whole</td>
<td>1659 [a,b,c,d,e]</td>
</tr>
<tr>
<td>N-Terminal</td>
<td>1659 [a]</td>
</tr>
<tr>
<td>C-Terminal</td>
<td>1659 [a]</td>
</tr>
<tr>
<td>Rest</td>
<td>1659 [a]</td>
</tr>
<tr>
<td>Split</td>
<td>1659*N [a]</td>
</tr>
<tr>
<td><strong>CTC</strong></td>
<td>Conjoint Triad Descriptors</td>
</tr>
<tr>
<td>Whole</td>
<td>343 [a,b,c,d,e]</td>
</tr>
<tr>
<td>N-Terminal</td>
<td>343 [a]</td>
</tr>
<tr>
<td>C-Terminal</td>
<td>343 [a]</td>
</tr>
<tr>
<td>Rest</td>
<td>343 [a]</td>
</tr>
<tr>
<td>Split</td>
<td>343*N [a]</td>
</tr>
<tr>
<td><strong>CcTD</strong></td>
<td>Composition enhanced Transition Distribution</td>
</tr>
<tr>
<td>Whole</td>
<td>189 [a,b,c,d,e]</td>
</tr>
<tr>
<td>N-Terminal</td>
<td>189 [a]</td>
</tr>
<tr>
<td>C-Terminal</td>
<td>189 [a]</td>
</tr>
<tr>
<td>Rest</td>
<td>189 [a]</td>
</tr>
<tr>
<td>Split</td>
<td>189*N [a]</td>
</tr>
<tr>
<td><strong>PAAC</strong></td>
<td>Pseudo Amino Acid Composition</td>
</tr>
<tr>
<td>Whole</td>
<td>20 + λ [a,b,c,d,e]</td>
</tr>
<tr>
<td>N-Terminal</td>
<td>20 + λ [a]</td>
</tr>
<tr>
<td>C-Terminal</td>
<td>20 + λ [a]</td>
</tr>
<tr>
<td>Rest</td>
<td>20 + λ [a]</td>
</tr>
<tr>
<td>Split</td>
<td>N*(20 + λ) [a]</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>APAAC</td>
<td>Amphiphilic Pseudo Amino Acid Composition</td>
</tr>
<tr>
<td>QSO</td>
<td>Quasi-Sequence Order</td>
</tr>
<tr>
<td>(SOCN)</td>
<td>Sequence Order Coupling Number</td>
</tr>
<tr>
<td>BINARY PROFILES</td>
<td></td>
</tr>
<tr>
<td>AAB</td>
<td>Amino Acid Binary Profile</td>
</tr>
<tr>
<td>DPB</td>
<td>Dipeptide Binary Profile</td>
</tr>
<tr>
<td>ABB</td>
<td>Atom and Bond Binary Profile</td>
</tr>
<tr>
<td>PCB</td>
<td>Physico-Chemical Properties Binary Profile</td>
</tr>
<tr>
<td>AIB</td>
<td>Amino Acid Index Binary Profile</td>
</tr>
</tbody>
</table>
### EVOLUTIONARY INFORMATION

<table>
<thead>
<tr>
<th>Split</th>
<th>N<em>553</em>L</th>
<th>[a]</th>
</tr>
</thead>
<tbody>
<tr>
<td>G_PSSM</td>
<td>Generation of PSSM</td>
<td>Whole</td>
</tr>
<tr>
<td>N_PSSM</td>
<td>Normalization of PSSM</td>
<td>Whole</td>
</tr>
<tr>
<td>C_PSSM</td>
<td>Composition of PSSM</td>
<td>Whole</td>
</tr>
<tr>
<td>P_PSSM</td>
<td>Profile of PSSM</td>
<td>Whole</td>
</tr>
<tr>
<td>N-Terminal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C-Terminal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Rest</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### STRUCTURE

<table>
<thead>
<tr>
<th>Split</th>
<th>Whole</th>
<th>a: N<em>553</em>L</th>
<th>[a]</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIN</td>
<td>Fingerprints</td>
<td></td>
<td>14532</td>
</tr>
<tr>
<td>SMI</td>
<td>SMILES</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>SA</td>
<td>Surface Accessibility</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>SS</td>
<td>Secondary Structure</td>
<td></td>
<td>3</td>
</tr>
</tbody>
</table>

### PATTERN

<table>
<thead>
<tr>
<th>Split</th>
<th>Whole</th>
<th>L X (21*o)</th>
<th>[a]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Profile</td>
<td>Binary Profile generated using patterns of window length (o)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PSSM Profile</td>
<td>PSSM Profile generated using patterns of window length (o)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physico-Chemical Properties</td>
<td>Physico-Chemical Properties, calculated using patterns of window length (o)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AA Index</td>
<td>Amino acid index composition, calculated using patterns of window length (o)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Universal</td>
<td>Generation of patterns of window length (o)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### MODEL BUILDING

<table>
<thead>
<tr>
<th>Split</th>
<th>Whole</th>
<th>R X M</th>
<th>[a]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Merging Features</td>
<td>Merge the two files into single file</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Feature Relevance</td>
<td>Mean based method to get the relevance of each feature</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- a: Pfeature, b: ifeature, 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_YPY → 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 \rightarrow \text{Residue repeat index for small residues}
PRI_LR \rightarrow \text{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.

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*)

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Group 1</th>
<th>Group 2</th>
<th>Group 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>volume</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **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.
AP AAC1_SC_lam1 → Sequence correlation factor for side chain mass with lambda 1

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

AP AAC2_A → Amphiphilic pseudo amino acid composition of Alanine
AP AAC2_C → Amphiphilic pseudo amino acid composition of Cysteine
AP AAC2_D → Amphiphilic pseudo amino acid composition of Aspartic acid
AP AAC2_E → Amphiphilic pseudo amino acid composition of Glutamic acid
AP AAC2_F → Amphiphilic pseudo amino acid composition of Phenylalanine
AP AAC2_G → Amphiphilic pseudo amino acid composition of Glycine
AP AAC2_H → Amphiphilic pseudo amino acid composition of Histidine
AP AAC2_I → Amphiphilic pseudo amino acid composition of Isoleucine
AP AAC2_K → Amphiphilic pseudo amino acid composition of Lysine
AP AAC2_L → Amphiphilic pseudo amino acid composition of Leucine
AP AAC2_M → Amphiphilic pseudo amino acid composition of Methionine
AP AAC2_N → Amphiphilic pseudo amino acid composition of Asparagine
AP AAC2_P → Amphiphilic pseudo amino acid composition of Proline
AP AAC2_Q → Amphiphilic pseudo amino acid composition of Glutamine
AP AAC2_R → Amphiphilic pseudo amino acid composition of Arginine
AP AAC2_S → Amphiphilic pseudo amino acid composition of Serine
AP AAC2_T → Amphiphilic pseudo amino acid composition of Threonine
AP AAC2_V → Amphiphilic pseudo amino acid composition of Valine
AP AAC2_W → Amphiphilic pseudo amino acid composition of Tryptophan
AP AAC2_Y → Amphiphilic pseudo amino acid composition of Tyrosine

AP AAC2_HB_lam1 → Sequence correlation factor for hydrophobicity with lambda 1
AP AAC2_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
QSO3SCP → 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
SOC3_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 $20 \times$ window/protein length (N)

$A_1 \rightarrow$ Presence/Absence (1 or 0) for Alanine at position 1

$C_1 \rightarrow$ Presence/Absence (1 or 0) for Cysteine at position 1

$D_1 \rightarrow$ Presence/Absence (1 or 0) for Aspartic acid at position 1

$E_1 \rightarrow$ Presence/Absence (1 or 0) for Glutamic acid at position 1

$F_1 \rightarrow$ Presence/Absence (1 or 0) for Phenylalanine at position 1

-----

$A_2 \rightarrow$ Presence/Absence (1 or 0) for Alanine at position 2

$C_2 \rightarrow$ Presence/Absence (1 or 0) for Cysteine at position 2

$D_2 \rightarrow$ Presence/Absence (1 or 0) for Aspartic acid at position 2

$E_2 \rightarrow$ Presence/Absence (1 or 0) for Glutamic acid at position 2

$F_2 \rightarrow$ Presence/Absence (1 or 0) for Phenylalanine at position 2

-----

$A_n \rightarrow$ Presence/Absence (1 or 0) for Alanine at position n

$C_n \rightarrow$ Presence/Absence (1 or 0) for Cysteine at position n

$D_n \rightarrow$ Presence/Absence (1 or 0) for Aspartic acid at position n

$E_n \rightarrow$ Presence/Absence (1 or 0) for Glutamic acid at position n

$F_n \rightarrow$ 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

---

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

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

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

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

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

S11 → 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\textsuperscript{th} position
DO_m → Presence/Absence (1 or 0) for double bond at m\textsuperscript{th} position
CY_m → Presence/Absence (1 or 0) for cyclic ring at m\textsuperscript{th} position
BE_m → Presence/Absence (1 or 0) for benzene ring at m\textsuperscript{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 $25 \times \text{window/protein length}(n)$

PC1 $\rightarrow$ Presence/Absence (1 or 0) for positively charged residues at position 1

NC1 $\rightarrow$ Presence/Absence (1 or 0) for positively charged residues at position 1

NE1 $\rightarrow$ Presence/Absence (1 or 0) for neutral charged residues at position 1

PO1 $\rightarrow$ Presence/Absence (1 or 0) for polar residues at position 1

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

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

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

AR1 $\rightarrow$ Presence/Absence (1 or 0) for aromatic residues at position 1

AC1 $\rightarrow$ Presence/Absence (1 or 0) for acidic residues at position 1

BS1 $\rightarrow$ Presence/Absence (1 or 0) for basic residues at position 1

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

HB1 $\rightarrow$ Presence/Absence (1 or 0) for hydrophobic residues at position 1

HL1 $\rightarrow$ Presence/Absence (1 or 0) for hydrophilic residues at position 1

NT1 $\rightarrow$ Presence/Absence (1 or 0) for neutral residues at position 1

HX1 $\rightarrow$ Presence/Absence (1 or 0) for hydroxylic residues at position 1

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

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

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

SS_CO1 $\rightarrow$ 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/](https://webs.iiitd.edu.in/raghava/pfeature/)
- **Python Library**: [https://github.com/raghavagps/Pfeature/tree/master/PyLib](https://github.com/raghavagps/Pfeature/tree/master/PyLib)
- **Standalone Version**: [https://github.com/raghavagps/Pfeature/tree/master/Standalone](https://github.com/raghavagps/Pfeature/tree/master/Standalone)
- **Python Scripts**: [https://github.com/raghavagps/Pfeature/tree/master/scripts](https://github.com/raghavagps/Pfeature/tree/master/scripts)
- **GitHub**: [https://github.com/raghavagps/Pfeature](https://github.com/raghavagps/Pfeature)

11.2 Documentation

- **Pfeature Manual**

11.3 Descriptors

- **Function Table**
  - [https://github.com/raghavagps/Pfeature/blob/master/PyLib/Functions_Tables.pdf](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](https://github.com/raghavagps/Pfeature/blob/master/Feature_Description.pdf)

- **Descriptor Headers**
  - [https://github.com/raghavagps/Pfeature/blob/master/Descriptor_headers.pdf](https://github.com/raghavagps/Pfeature/blob/master/Descriptor_headers.pdf)