Techniques Used in Literatures

- Amino Acid Composition (AAC)
- Dipeptide Composition
- Composition of k-spaced amino acid and pairs (CKSAAP)
- One-hot encoding

Amino Acid Composition: In protein sequence there are 20 types of amino acids. If we want to describe a fraction of amino acid from the protein sequence, we have to use amino acid composition technique.

Formula: $f(r) = N_r / N$; $r = 1,2,3,\ldots\ldots,20$

Here, N_r is the number of amino acid type. R and N is the length of the sequence.

- ACWYKAYWX window size: 9

A	C	D	E	………..	W	X	Y
$\frac{2}{9}$	$\frac{1}{9}$	$\frac{0}{9}$	$\frac{0}{9}$	………..	$\frac{2}{9}$	$\frac{1}{9}$	$\frac{2}{9}$

We can use this technique. But it has few problems. For example, If I switch 2nd “W” with 3rd last “Y” It will give the same result. Another problem is it does not ensure the position of the amino acids.

Dipeptide Composition: Dipeptide composition will give 400-dimensional descriptor. If we count “X” the 441 dimensional (21*21).

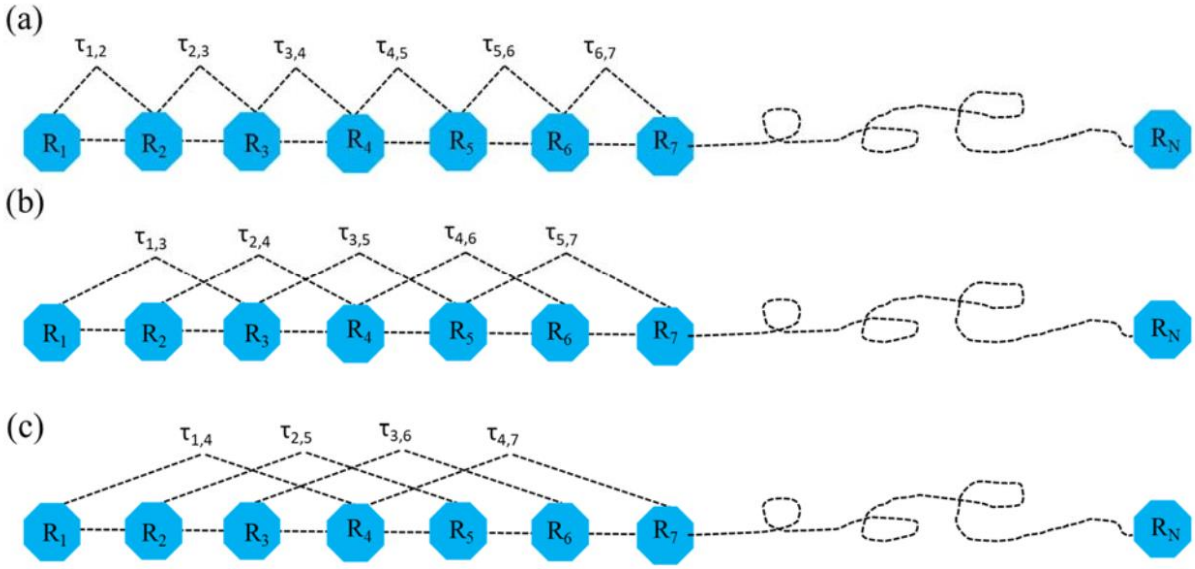
ACWYKAYWX

AA	**AC**	**AD**	**AE**	**……….**	**YW**	**YX**	**YY**
$\frac{0}{8}$	$\frac{\mathbf{1}}{8}$	$\frac{\mathbf{0}}{8}$	$\frac{\mathbf{0}}{8}$	**……….**	$\frac{1}{8}$	$\frac{0}{8}$	$\frac{0}{8}$

Now, even if I switch the amino acids it will not give the same results.

Composition of k-spaced amino acid and pairs (CKSAAP): If we want to calculate the amino acid pair frequency separated by k residues(any) then 'Amino acid composition or, Dipeptide composition' won't be much of a help. To help CKSAAP comes handy.

Here, (K = 3) 0, 1 , 2



One Hot Vector: With one hot vector encoding we can convert the categorical variables into numerical values. For each amino acid 21 unique pattern is needed.

A = {1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
K = {0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
D = {0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
.
.
.
.
Rest……….

One might ask why don’t we use {1,1,0,………….} instead of {1,0,0,…………}. For that to answer we have to think about the algorithm. If we use {1,1,0,………….}, then it will put more weight than {1,0,0,…………}. We don’t want that.

KNN: K-nearest neighbor is a very simple Machine Learning algorithm. It can be used in both classification and regression problems. By using Euclidean distance, it calculates distance between points and selects specified number of examples (“K”) closest to that point. If the problem is classification problem it votes for frequent results. If it is regression problem;, then it average the results.