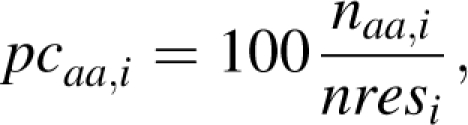
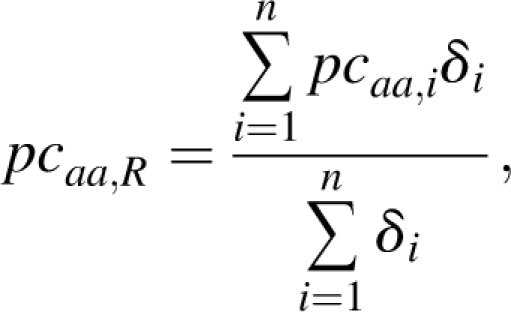
AAC: Amino acid composition was often used to describe protein sequences and to design predictive algorithms. The percentage of occurrence *pcaa,i* of the amino acid *aa* in the *i*th protein was computed for each of the 20 types of amino acids in each protein as



where *naa,i* and *nresi* are the number of residues of type *aa* observed in protein *i* and the total number of residues in protein *i*, respectively. Then, the *pcaa,i* values were averaged for all protein that contains the same number of residues *R*, by computing



where δ*i* = 1 if *nresi* = *R*, δ*i* = 0 if *nresi* ≠ *R*, and *R* is the number of residues. The resulting quantity *pcaa,R* is thus the percentage with which residue *aa* (*aa* = A, C, D, …, V, W, Y) is observed in proteins containing *R* residues.

|  |
| --- |
| **Dipeptide pseudo amino acid composition** |
| The option of dipeptide composition will generate 420 components for each protein sequence, the first 20 components are the conventional amino acid composition(AAC); the following 400 components are the fractions of 400 dipeptides, i.e. AA, AC, AD, http://www.csbio.sjtu.edu.cn/bioinf/PseAA/images/hellip.gif, YV, YW, YY; the 400 components are calculated using the following equation, |
|  |
| http://www.csbio.sjtu.edu.cn/bioinf/PseAA/images/dipep.gif |
| where dep(*i*) is the *ith* dipeptide of the 400 dipeptides, *i*=1,2,http://www.csbio.sjtu.edu.cn/bioinf/PseAA/images/hellip.gif,400. |
|  |
| **The format of the output for the 420 components are:** |
| ***1st line***: 20 components of amino acid composition (AAC); |
| ***2nd line***: 20 components of dipeptide composition beginning with amino acid A, i.e. AA, AC, AD, http://www.csbio.sjtu.edu.cn/bioinf/PseAA/images/hellip.gif, AY; |
| ***3rd line***: 20 components of dipeptide composition beginning with amino acid C, i.e. CA, CC, CD, http://www.csbio.sjtu.edu.cn/bioinf/PseAA/images/hellip.gif, CY; |
| http://www.csbio.sjtu.edu.cn/bioinf/PseAA/images/hellip.gifhttp://www.csbio.sjtu.edu.cn/bioinf/PseAA/images/hellip.gif |
| ***21st line***: 20 components of dipeptide composition beginning with amino acid Y, i.e. YA, YC, YD, http://www.csbio.sjtu.edu.cn/bioinf/PseAA/images/hellip.gif, YY |

Word2Vec: **Word2vec** is a group of related models that are used to produce [word embeddings](https://en.wikipedia.org/wiki/Word_embedding). These models are shallow, two-layer [neural networks](https://en.wikipedia.org/wiki/Neural_network) that are trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large [corpus of text](https://en.wikipedia.org/wiki/Text_corpus) and produces a [vector space](https://en.wikipedia.org/wiki/Vector_space), typically of several hundred [dimensions](https://en.wikipedia.org/wiki/Dimensions), with each unique word in the [corpus](https://en.wikipedia.org/wiki/Corpus_linguistics) being assigned a corresponding vector in the space. [Word vectors](https://en.wikipedia.org/wiki/Word_vectors) are positioned in the vector space such that words that share common contexts in the corpus are located close to one another in the space.[[1]](https://en.wikipedia.org/wiki/Word2vec#cite_note-mikolov-1)An extension of word vectors for n-grams in [biological](https://en.wikipedia.org/wiki/Biological) sequences (e.g. [DNA](https://en.wikipedia.org/wiki/DNA), [RNA](https://en.wikipedia.org/wiki/RNA), and [Proteins](https://en.wikipedia.org/wiki/Protein)) for [bioinformatics](https://en.wikipedia.org/wiki/Bioinformatic) applications have been proposed by Asgari and Mofrad.[[13]](https://en.wikipedia.org/wiki/Word2vec#cite_note-:0-13) Named bio-vectors (BioVec) to refer to biological sequences in general with protein-vectors (ProtVec) for proteins (amino-acid sequences) and gene-vectors (GeneVec) for gene sequences, this representation can be widely used in applications of machine learning in proteomics and genomics. The results suggest that BioVectors can characterize biological sequences in terms of biochemical and biophysical interpretations of the underlying patterns. So far, we have seen deterministic methods to determine word vectors. But these methods proved to be limited in their word representations until Mitolov etc. el introduced word2vec to the NLP community. These methods were prediction based in the sense that they provided probabilities to the words and proved to be state of the art for tasks like word analogies and word similarities. They were also able to achieve tasks like King -man +woman = Queen, which was considered a result almost magical. So let us look at the word2vec model used as of today to generate word vectors.Word2vec is not a single algorithm but a combination of two techniques – CBOW(Continuous bag of words) and Skip-gram model. Both of these are shallow neural networks which map word(s) to the target variable which is also a word(s).

Both of these techniques learn weights which act as word vector representations. Let us discuss both these methods separately and gain intuition into their working. The purpose and usefulness of **Word2vec** is to group the vectors of similar words together in vectorspace. That is, it detects similarities mathematically. **Word2vec** creates vectors that are distributed numerical representations of word features, features such as the context of individual words.

AAindex: An **amino acid index** is a set of 20 numerical values representing various physico-chemical and biochemical properties of **amino acids**. An **amino acid** mutation matrix is generally 20 × 20 numerical values representing similarity of **amino acids**.

An amino acid index is a set of 20 numerical values representing any of the different physicochemical and biological properties of amino acids. The AAindex1 section of the Amino Acid Index Database is a collection of published indices together with the result of cluster analysis using the correlation coefficient as the distance between two indices. This section currently contains 566 indices.

Another important feature of amino acids that can be represented numerically is the similarity between amino acids. Thus, a similarity matrix, also called a mutation matrix, is a set of 210 numerical values, 20 diagonal and 20x19/2 off-diagonal elements, used for sequence alignments and similarity searches. The AAindex2 section of the Amino Acid Index Database is a collection of published amino acid mutation matrices together with the result of cluster analysis. This section currently contains 94 matrices.

In the release 9.0, we added a collection of published protein pairwise contact potentials to AAindex as AAindex3. This section currently contains 47 contact potential matrices.

 SVM: **SVM** is a supervised **machine** learning algorithm which can be **used for** classification or regression problems. It **uses** a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs. **SVM** or **Support Vector Machine** is a linear model for classification and regression problems. It **can** solve linear and non-linear problems and **work** well for many practical problems. The idea of **SVM** is simple: The algorithm creates a line or a hyperplane which separates the data into classes. **Support vectors** are data points that are closer to the hyperplane and influence the position and orientation of the hyperplane. Using these **support vectors**, we maximize the margin of the classifier. Deleting the **support vectors** will change the position of the hyperplane. These are the points that help us build our **SVM.** In [machine learning](https://en.wikipedia.org/wiki/Machine_learning), **support-vector machines** (**SVMs**, also **support-vector networks**[[1]](https://en.wikipedia.org/wiki/Support_vector_machine#cite_note-CorinnaCortes-1)) are [supervised learning](https://en.wikipedia.org/wiki/Supervised_learning) models with associated learning [algorithms](https://en.wikipedia.org/wiki/Algorithm) that analyze data used for [classification](https://en.wikipedia.org/wiki/Statistical_classification) and [regression analysis](https://en.wikipedia.org/wiki/Regression_analysis). Given a set of training examples, each marked as belonging to one or the other of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-[probabilistic](https://en.wikipedia.org/wiki/Probabilistic_classification) [binary](https://en.wikipedia.org/wiki/Binary_classifier) [linear classifier](https://en.wikipedia.org/wiki/Linear_classifier) (although methods such as [Platt scaling](https://en.wikipedia.org/wiki/Platt_scaling) exist to use SVM in a probabilistic classification setting). A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on the side of the gap on which they fall.

In addition to performing [linear classification](https://en.wikipedia.org/wiki/Linear_classifier), SVMs can efficiently perform a non-linear classification using what is called the [kernel trick](https://en.wikipedia.org/wiki/Kernel_method#Mathematics:_the_kernel_trick), implicitly mapping their inputs into high-dimensional feature spaces.

When data are unlabelled, supervised learning is not possible, and an [unsupervised learning](https://en.wikipedia.org/wiki/Unsupervised_learning) approach is required, which attempts to find natural [clustering of the data](https://en.wikipedia.org/wiki/Cluster_analysis) to groups, and then map new data to these formed groups. The **support-vector clustering**[[2]](https://en.wikipedia.org/wiki/Support_vector_machine#cite_note-HavaSiegelmann-2) algorithm, created by [Hava Siegelmann](https://en.wikipedia.org/wiki/Hava_Siegelmann" \o "Hava Siegelmann) and [Vladimir Vapnik](https://en.wikipedia.org/wiki/Vladimir_Vapnik), applies the statistics of support vectors, developed in the support vector machines algorithm, to categorize unlabeled data, and is one of the most widely used clustering algorithms in industrial applications

KNN: K nearest neighbors is a simple **algorithm** that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). **KNN** has been used in statistical estimation and pattern recognition already in the beginning of 1970's as a non-parametric technique. **KNN works** by finding the distances between a query and all the examples in the data, selecting the specified number examples (**K**) **closest** to the query, then votes for the most frequent label (in the case of classification) or averages the labels (in the case of regression).

CNN: A convolutional neural network (**CNN**) is a specific type of artificial neural network that uses perceptrons, a **machine learning** unit algorithm, for supervised **learning**, to analyze data. CNNs apply to image processing, natural language processing and other kinds of cognitive tasks. One of the main parts of Neural Networks is Convolutional neural networks (**CNN**). CNNs use image recognition and classification in order to detect objects, recognize faces, etc. ... CNNs are primarily used to classify images, cluster them by similarities, and then perform object recognition. Generally most of the Neural Networks are used in **supervised** methods. Either to predict (regression) something or in classification. Convolutional Neural Networks (**CNN**) are widely employed in Image Classifications and other object recognition applications. ... However, mostly **supervised.** hey have applications in [image and video recognition](https://en.wikipedia.org/wiki/Computer_vision), [recommender systems](https://en.wikipedia.org/wiki/Recommender_system),[[4]](https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-4) [image classification](https://en.wikipedia.org/wiki/Image_classification), [medical image analysis](https://en.wikipedia.org/wiki/Medical_image_computing), [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing),[[5]](https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-5) and financial [time series](https://en.wikipedia.org/wiki/Time_series).[[6]](https://en.wikipedia.org/wiki/Convolutional_neural_network#cite_note-Tsantekidis_7%E2%80%9312-6)

**RNN:** A **recurrent neural network** (**RNN**) is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. ... Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. Why is an **RNN** (**Recurrent Neural Network**) **used for machine translation**, say **translating** English to French? (Check all that apply.) ... It is strictly more powerful than a Convolutional Neural Network (CNN). It is applicable when the input/output is a sequence (e.g., a sequence of words). As per Wikipedia, a recurrent neural network (**RNN**) is a class of artificial neural network where connections between units form a directed graph along a sequence. This allows it to exhibit dynamic temporal behavior for a time sequence. ... In other neural networks, all the inputs are independent of each other. Recurrent neural networks were traditionally difficult to train. The Long Short-Term Memory, or **LSTM**, network is perhaps the most successful **RNN** because it overcomes the problems of training a recurrent network and in turn has been used on a wide range of applications. Long short-term memory (**LSTM**) is an artificial recurrent **neural network** (RNN) architecture used in the field of **deep learning**. ... **LSTM** networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series