Chapter 2

THEORETICAL BACKGROUND

In this chapter, theoretical backgrounds related to protein function prediction by using sequence derived properties are presented. Section 2.1 presents literature review for the prediction of ion channels, enzymes, nuclear and G-protein coupled receptors and their subfamilies. Section 2.2 presents the extraction of sequence derived properties of protein such as amino acid composition, dipeptide composition, correlation factors, composition, transition, distribution, sequence order descriptor's and pseudo amino acid compositions. Section 2.3 presents feature selection techniques such as filter, wrapper and hybrid methods. Section 2.4 presents about basic concepts of various computational intelligence techniques used in protein function prediction. Section 2.5 presents basic concepts for measuring the performance of the classifiers. Finally conclusion is presented in section 2.6.

2.1. Literature review

This section presents a literature review of various computational intelligence techniques used in the prediction of ion channels, enzymes, nuclear and G-protein coupled receptors and their subfamilies by using sequence derived properties. The summary of the results obtained by many researchers are also presented to solve these problems by using computational intelligence techniques based approaches with appropriate datasets to improve the prediction performance.

2.1.1. Computational intelligence techniques in the prediction of ion channels and their types

Ion channels are membrane proteins that are responsible for electrical signaling by gating the flow of ions across the cell membrane. These are the prominent component of nervous systems. The dysfunction of ion channels play an important role in the development of various diseases such as hypertension, defective insulin secretion, cardiac arrhythmias, neurological diseases such as epilepsy and even developmental defects such as osteoporosis (Jentsch *et al.,* 2004). So it is necessary to know about the structure and function of the ion channels to develop a new drug for these diseases. Therefore, it is necessary to design a robust and efficient computational intelligence techniques based method to predict ion channels and their types. So for achieving this objective support vector machine (SVM) based techniques have been proposed in literatures. Here, an analysis of various computational intelligence techniques based methods available in literature is presented and examines the efficacy of each of these methods for the predictions of ion channels and their types which are as follows:

There are few papers which reported computational intelligence techniques based methods to predict ion channels and their types. Willett *et al.* (2007) and Pourbasheer *et al.* (2009) have proposed computational intelligence techniques based method to predict the activity of ion channel proteins. Liu *et al.* (2006) proposed a support vector machine based method to predict five types of voltage gated potassium channels and obtained accuracy of 98%. Saha *et al.* (2006) proposed a support vector machine based method to predict four types of voltage gated ion channels by using amino acid and dipeptide composition and obtained overall accuracy of 97.78%. Chen *et al.,* (2012) proposed a support vector machine based method to predict voltage gated potassium channel subfamilies by using amino acid and dipeptide composition and obtained overall accuracy of 93.09%. Lin *et al.* (2011) proposed a support vector machine based method to predict ion channels and their types by using dipeptide mode of pseudo amino acid composition and obtained overall accuracy of 86.6% to discriminate ion channels from non-ion channels, overall accuracy of 92.6% to classify voltage gated ion channels and ligand gated ion channels and an overall accuracy of 87.8% to predict four types of voltage gated ion channels.

Table 2.1: Summary of computational intelligence techniques in prediction of ion channels and their types

Author	CIT	Prediction	Performance	Datasets
Liu *et al.* (2006)	SVM	Voltage gated potassium channels	Overall accuracy: 98%	Dipeptide composition of amino acids
Saha *et al.* (2006)	SVM	Voltage gated ion channels	Overall accuracy: 97.78 %	Amino acid and dipeptide composition
Lin *et al.* (2011)	SVM	Types of ion channels	Overall accuracy: 86.6%, 92.6% and 87.8% for prediction of ion channels and non-ion channels, voltage gated and ligand gated ion channels and typed of voltage gated ion channels respectively	Pseudo amino acid composition
Chen *et al.* (2012)	SVM	Voltage gated potassium channel	Overall accuracy: 93.09%	Amino acid and dipeptide composition

2.1.2. Computational intelligence techniques in the prediction of enzyme functions

Enzymes are catalysts which speed up the rate of reaction without becoming the part of reaction. Enzyme proteins play an important role in metabolic pathways. Prediction of enzyme functional classes and subclasses play an important role into the research of the drugs design. So it is necessary to design a robust and efficient computational intelligence techniques based method to predict enzyme functional classes and subclasses. So for achieving this objective various computational intelligence techniques have been proposed in literatures. Some of the prominent computational intelligence techniques reported in literature for the application under consideration includes random forest, artificial neural network (ANN), support vector machine (SVM) and k-nearest neighbors (k-NN). Here, an analysis of various computational intelligence techniques based methods available in literature is presented and examines the efficacy of each of these methods for the predictions of enzyme functions which are as follows:

The Jensen *et al.* (2002) developed an artificial neural network based model for the classification of enzymes from amino acid sequences by using sequence similarity and other sequence derived features such as co-translational and post translational modification, secondary structure and physical and chemical properties. Cai *et al.* (2005) used the nearest neighbor method with the functional domain composition of a protein to predict enzyme family classes. Borro *et al.* (2006) proposed a Bayesian based approach with structure derived properties of a protein for the classification of enzymes. Lu *et al*. (2007) proposed the support vector machine (SVM) based methods by using feature vector from protein sequences such as functional domain composition for the classification of enzymes functional classes and sub-classes. Zhou *et al.* (2007) proposed SVM based method with amphiphilic pseudo amino acid composition for the classification of enzymes functional classes and sub-classes.

The k-nearest neighbor (k-NN) based method has been proposed by Huang *et al.* (2007) with amphiphilic pseudo-amino acid composition that includes the both features such as sequence order related features and the function related features for the classification of enzymes functional classes and sub-classes. Shen *et al.* (2007) used optimized evidence theoretic k-nearest neighbor classifier with functional domain composition and position specific scoring matrix for the classification of enzymes functional classes and sub-classes. They constructed a top-down three layer model where the top layer classifies a query protein sequence as an enzyme or non-enzyme, the second layer predicts the main function class and bottom layer further predicts the functional sub-classes. The k-nearest neighbor based method have been proposed by Cai *et al.* (2008) with a functional domain composition that includes the both features such as sequence order related features and the function related features and Nasibov *et al.* (2009) used amino acid composition for the classification of enzymes functional classes and sub-classes.

Lee *et al.* (2009) proposed support vector machine and random forest based methods by using sequence derived properties for the classification of enzymes functional classes and sub-classes. Later Wang *et al.* (2010) and Wang *et al.* (2011) used support vector machine based methods with pseudo amino acid composition and conjoint triad features to represent the protein sequences for the prediction of the families and functions of enzymes respectively. Yadav *et al.* (2012) proposed a support vector machine based approach using features extracted from the global structure based on fragment libraries for the classification of enzymes functional classes and sub-classes. Qiu *et al.* (2010) proposed an integrated method of support vector machine with discrete wavelet transform for the classification of the enzyme families by using hydrophobicity of amino acid from pseudo amino acid composition. Kumar *et al.* (2012) presented a random forest based method to predict the functional classes and sub-classes of enzymes based on sequence derived features. They constructed a top-down three layer model where the top layer classifies a query protein sequence as an enzyme or non-enzyme, the second layer predicts the six main functional classes and bottom layer further predicts the functional sub-classes. Volpato *et al.* (2013) proposed N-to-1 Neural Network for accurate prediction of enzyme by using amino acid sequences. Nagao *et al.* (2014) proposed a random forest based method for predicting enzyme functions with a set of specificity determining residues.

Table 2.2: Summary of computational intelligence techniques in prediction of enzyme functions

Author	**Computational Method**	**Performance**	**Datasets**
Cai *et al.* (2005)	kNN	Accuracy: 85%	Functional domain composition
Borro *et al.* (2006)	Bayesian Classifier	Accuracy: 45%.	Structural properties
Lu *et al.* (2007)	SVM	Accuracy :91.32%	Functional domain composition
Zhou. *et al.* (2007)	SVM	Accuracy: 80.87%.	Amphiphilic pseudo amino acid composition
Huang *et al.* (2007)	kNN	Accuracy : 76.6%,	Amphiphilic pseudo-amino acid composition
Shen *et al.* (2007)	OET-kNN	Overall accuracy: 91.3%, 93.7% and 98.3% for the 1st, 2nd and 3rd level	Functional domain composition and PSSM
Cai *et al.* (2008)	kNN	Accuracy: 85%.	Functional domain composition
Nasibov *et al.* (2009)	k-NN	Accuracy: 99%	Amino acid composition
Lee *et al.* (2009)	SVM and Random	Accuracy: 71.29- 99.53% by SVM and 94- 99.31%	Sequence derived properties

	Forest	by random forest	
Wang *et al.* (2010)	SVM	MCC: 0. 92 and Accuracy: 93%	Pseudo amino acid composition with (CTF)
Wang *et al.* (2011)	SVM	Accuracy: 81% to 98% and MCC: 0.82 to 0.98	Pseudo amino acid composition with (CTF
Yadav *et al.* (2012)	SVM	Accuracy: 95.25%	Structural features based on fragment libraries.
Qiu *et al.* (2010)	SVM with DWT	Accuracy: 91.9.	Pseudo amino acid composition
Kumar *et al.* (2012)	Random Forest	Overall accuracy: 94.87%, 87.7% and 84.25% for the 1st, 2nd and 3rd level.	Sequence-derived features
Volpato *et al.* (2013)	N-to-1 Neural Network	Overall accuracy: 96%, Specificity: 80% and FP rates: 7%.	Amino acid sequences
Nagao *et al.* (2014)	Random forest	Precision: 0.98 and Recall: 0.89	Set of specificity determining residues

Some of the observations related to the computational intelligence techniques in prediction of enzyme functions protein presented in Section 2.1.2 are as follows:

- The SVM, random forest and k-NN based methods are useful for the prediction of enzyme functions and families.

- The overall accuracy obtained by SVM ranges in between 69.1-99.53%, random forest ranges in between 71.29-99.31%, and k-NN ranges in between 56.9-99.0% for the various diverse datasets as reported in Table 2.2.

- All the SVM, random forest and k-NN based method obtained maximum accuracy by using the sequence derived properties.

- The variation of ANN is proposed as N-to-1 neural network and by using protein sequence and obtained 96% accuracy.

- So from the analysis it is observed that the sequence derived properties are useful to predict the enzyme functions and families.

2.1.3. Computational intelligence techniques in prediction of nuclear receptors and their subfamilies

Nuclear receptors (NRs) are key transcription factors that regulate a wide variety of biological processes such as homeostasis, reproduction, development, and metabolism. The nuclear receptors are involves in many physiological and pathological processes so prediction of different nuclear receptors and their subfamilies is a most challenging problem in bioinformatics. So for achieving this objective various computational intelligence techniques have been proposed in literatures. Some of the prominent computational intelligence techniques reported in literature for the application under consideration includes support vector machine (SVM) and k-nearest neighbors (k-NN). Here, an analysis of various computational intelligence techniques based methods available in literature is presented and examines the efficacy of each of these methods for the predictions of nuclear receptors and their subfamilies which are as follows:

Initially Bhasin *et al.* (2004) proposed a support vector machine based method by using amino acid composition and dipeptide of amino acids for the prediction of nuclear receptors and their sub families but they consider only 4 subfamilies in their datasets. Later Cai *et al.* (2005) proposed a SVM based method to classify the 19 subfamilies of nuclear receptors by using 4-tuple residue composition instead of dipeptide composition to encode the nuclear receptor sequences and after that to improve the prediction performance Gao *et al.* (2009) reconstruct the dataset used by Bhasin *et al.* (2004) and used SVM to predict nuclear receptors by using optimal pseudo amino acid composition based on physicochemical characters of amino acids and also determine the correlation factor and the weighting factor about pseudo amino acid composition to get the appropriate descriptor of proteins but they also consider only 4 subfamilies of nuclear receptors. Wang *et al.* (2011) used fuzzy k-nearest neighbor classifier based on the pseudo amino acid composition with physicochemical and statistical features derived from the protein sequences such as amino acid composition, dipeptide composition, complexity factor, and low-frequency fourier spectrum components and Xiao *et al.* (2012) used SVM by using pseudo amino acid composition whose components were derived from a physical-chemical matrix via a series of auto-covariance and cross-covariance transformations to predict the seven subfamilies of nuclear receptor. Recently Wang *et al.* (2014) proposed a SVM based method for the prediction of nuclear receptors by using amino acid composition, dipeptide composition and physicochemical properties to predict the eight subfamilies of nuclear receptors.

Table 2.3: Summary of computational intelligence techniques in prediction of nuclear receptors and their subfamilies

Author	Computational Method	Performance	Datasets
Bhasin *et al.* (2004)	SVM	Overall accuracy: 82.6% by AAC and 97.5% by dipeptide	Amino acid composition and dipeptide composition
Cai *et al.* (2005)	SVM	Overall accuracy: 96%	4-tuple residue composition
Gao *et al.* (2009)	SVM	Overall accuracy: 99.6%	Pseudo amino acid composition
Wang *et al.* (2011)	Fuzzy kNN	Overall accuracy: 93%	Pseudo amino acid composition with physicochemical and statistical features
Xiao *et al.* (2012)	SVM	Accuracy: 98%.	Pseudo amino acid composition
Wang *et al.* (2014)	SVM	Accuracy: 97%.	Amino acid composition, dipeptide composition and physicochemical property

Some of the observations related to the computational intelligence techniques in prediction of nuclear receptors and their subfamilies of protein presented in Section 2.1.3 are as follows:

- The SVM and fuzzy k-NN based methods are useful for the prediction of nuclear receptors and their subfamilies.
- The overall accuracy obtained by SVM ranges in between 82.6-99.6% and by fuzzy k-NN is 93% (See Table 2.3).
- The SVM based method obtained maximum 99.6% accuracy by using the pseudo amino acid composition of protein sequences.

2.1.4. Computational intelligence techniques in prediction of G-protein coupled receptors and their subfamilies

G-protein coupled receptors (GPCRs) are seven-transmembrane domain receptors that sense molecules outside the cell and activate inside signal transduction pathways for cellular responses. These are called seven transmembrane receptors because they pass through the cell membrane seven times. There are a larger number of G-protein coupled receptors are available in human in these some have been identified their function like growth factors, light, hormones, amines, neurotransmitters, and lipids etc. However, a large number of the GPCRs found in the human genome have unknown functions and so it is necessary to design an efficient approach to predict families and subfamilies of G-protein coupled receptors for the new drug discovery. So for achieving this objective various computational intelligence techniques have been proposed in literatures. Some of the prominent computational intelligence techniques reported in literature for the application under consideration includes support vector machine (SVM) and k-nearest neighbors (k-NN). Here, an analysis of various computational intelligence techniques based method for the prediction of G-protein coupled receptors available in literature is presented and examines the efficacy of each of these methods for the predictions of G-protein coupled receptors and their subfamilies which are as follows:

Initially Bhasin *et al.* (2004) proposed a SVM based method by using amino acid composition and dipeptide of amino acids for the prediction of G-protein coupled receptor. Later Bhasin *et al.* (2005) proposed a SVM based method for the classification of amine type of G-protein-coupled receptors by using of amino acid composition and dipeptide composition of proteins. Gao *et al.* (2006) proposed a nearest neighbor method to discriminate GPCRs from non-GPCRs and subsequently classify GPCRs at four levels on the basis of amino acid composition and dipeptide composition of proteins. Gu *et al.* (2010) proposed an Adaboost classifier to predict G-protein coupled receptors and their subfamilies by pseudo amino acid composition with approximate entropy and hydrophobicity patterns. Peng *et al.* (2010) proposed a principal component analysis based method for the prediction of G-protein coupled receptors and their subfamilies by using sequence derived features.

Table 2.4: Summary of computational intelligence techniques in prediction of G-protein coupled receptors and their subfamilies

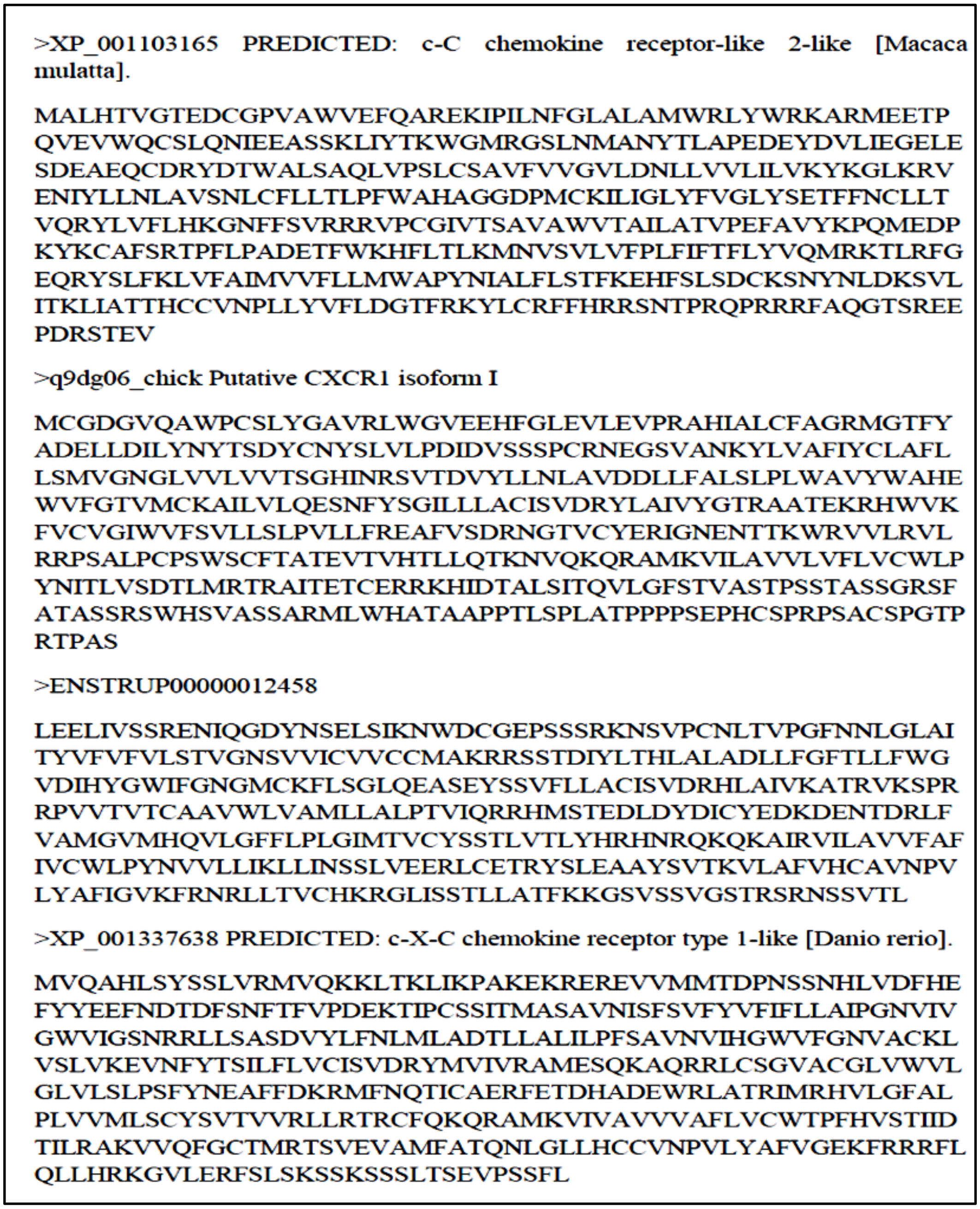
Author	Computational Method	Performance	Datasets
Bhasin *et al.* (2004)	SVM	Overall accuracy: 99.5%	Dipeptide composition of amino acids
Bhasin and Raghava (2005)	SVM	Overall accuracy: 89.8 % by AAC and 96.4% by dipeptide	Amino acid composition and dipeptide composition
Gao *et al.* (2006)	kNN	Overall accuracy: 96.4% MCC: 0.930	Amino acid composition and dipeptide composition
Gu *et al.* (2010)	Adaboost	Overall accuracy: 91.2%	Pseudo amino acid composition with approximate entropy and hydrophobicity patterns
Peng *et al.* (2010)	PCA	Overall accuracies: from first to the fifth level 99.5%, 88.8%, 80.47%, 80.3%, and 92.34%,	Sequence derived features

Some of the observations related to the computational intelligence techniques in prediction of G-protein coupled receptors and their subfamilies presented in Section 2.1.4 are as follows:

- The SVM, and k-NN based methods are useful for the prediction of G-protein coupled receptors and their subfamilies.
- The overall accuracy obtained by SVM ranges in between 89.8-99.5 and k-NN based classifier obtained overall accuracy 96.4% (See Table2.4).
- The ensemble based Adaboost classifier obtained maximum accuracy 91.2% by using the pseudo amino acid composition with approximate entropy.

2.2. Features extraction of protein sequences

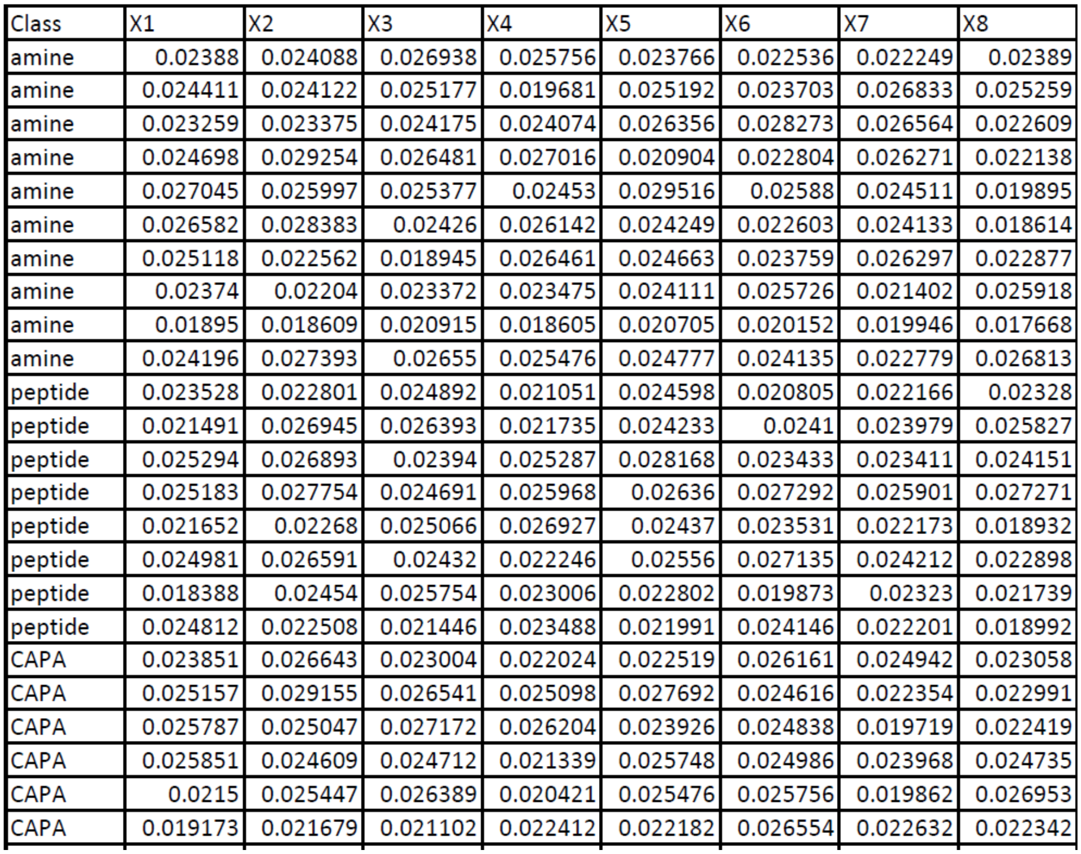
Proteins are a chain of 20 amino acids in specific orders. For the experimental purposes the sequence of protein sequences are extracted from standard repository such as SWISS-PROT (Boeckmann *et al.*, 2003), universal protein resource (UniProt) (Bairoch *et al.,* 2005), national center for biotechnology information (NCBI) (Wheeler *et al.*, 2007) and protein data bank (PDB) (Bernstein *et al.*, 1997) etc. in a FASTA format as shown in Figure 2.1.



```
>XP_001103165 PREDICTED: c-C chemokine receptor-like 2-like [Macaca
mulatta].

MALHTVGTEDCGPVAWVEFQAREKIPILNFGLALAMWRLYWRKARMEETP
QVEVWQCSLQNIEEASSKLIYTKWGMRGSLNMANYTLAPEDEYDVLIEGELE
SDEAEQCDRYDTWALSAQLVPSLCSAVFVVGVLDNLLVVLILVKYKGLKRV
ENIYLLNLAVSNLCFLLTLPFWAHAGGDPMCKILIGLYFVGLYSETFFNCLLT
VQRYLVFLHKGNFFSVRRRVPCGIVTSAVAWVTAILATVPEFAVYKPQMEDP
KYKCAFSRTPFLPADETFWKHFLTLKMNVSVLVFPLFIFTFLYVQMRKTLRFG
EQRYSLFKLVFAIMVVFLLMWAPYNIALFLSTFKEHFSLSDCKSNYNLDKSVL
ITKLIATTHCCVNPLLYVFLDGTFRKYLCRFFHRRSNTPRQPRRRFAQGTSREE
PDRSTEV

>q9dg06_chick Putative CXCR1 isoform I

MCGDGVQAWPCSLYGAVRLWGVEEHFGLEVLEVPRAHIALCFAGRMGTFY
ADELLDILYNYTSDYCNYSLVLPDIDVSSSPCRNEGSVANKYLVAFIYCLAFL
LSMVGNGLVVLVVTSGHINRSVTDVYLLNLAVDDLLFALSLPLWAVYWAHE
WVFGTVMCKAILVLQESNFYSGILLLACISVDRYLAIVYGTRAATEKRHWVK
FVCVGIWVFSVLLSLPVLLFREAFVSDRNGTVCYERIGNENTTKWRVVLRVL
RRPSALPCPSWSCFTATEVTVHTLLQTKNVQKQRAMKVILAVVLVFLVCWLP
YNITLVSDTLMRTRAITETCERRKHIDTALSITQVLGFSTVASTPSSTASSGRSF
ATASSRSWHSVASSARMLWHATAAPPTLSPLATPPPPSEPHCSPRPSACSPGTP
RTPAS

>ENSTRUP00000012458

LEELIVSSRENIQGDYNSELSIKNWDCGEPSSSRKNSVPCNLTVPGFNNLGLAI
TYVFVFVLSTVGNSVVICVVCCMAKRRSSTDIYLTHLALADLLFGFTLLFWG
VDIHYGWIFGNGMCKFLSGLQEASEYSSVFLLACISVDRHLAIVKATRVKSPR
RPVVTVTCAAVWLVAMLLALPTVIQRRHMSTEDLDYDICYEDKDENTDRLF
VAMGVMHQVLGFFLPLGIMTVCYSSTLVTLYHRHNRQKQKAIRVILAVVFAF
IVCWLPYNVVLLIKLLINSSLVEERLCETRYSLEAAYSVTKVLAFVHCAVNPV
LYAFIGVKFRNRLLTVCHKRGLISSTLLATFKKGSVSSVGSTRSRNSSVTL

>XP_001337638 PREDICTED: c-X-C chemokine receptor type 1-like [Danio rerio].

MVQAHLSYSSLVRMVQKKLTKLIKPAKEKREREVVMMTDPNSSNHLVDFHE
FYYEEFNDTDFSNFTFVPDEKTIPCSSITMASAVNISFSVFYVFIFLLAIPGNVIV
GWVIGSNRRLLSASDVYLFNLMLADTLLALILPFSAVNVIHGWVFGNVACKL
VSLVKEVNFYTSILFLVCISVDRYMVIVRAMESQKAQRRLCSGVACGLVWVL
GLVLSLPSFYNEAFFDKRMFNQTICAERFETDHADEWRLATRIMRHVLGFAL
PLVVMLSCYSVTVVRLLRTRCFQKQRAMKVIVAVVVAFLVCWTPFHVSTIID
TILRAKVVQFGCTMRTSVEVAMFATQNLGLLHCCVNPVLYAFVGEKFRRRFL
QLLHRKGVLERFSLSKSSKSSSLTSEVPSSFL
```

Figure 2.1: Protein sequences in FASTA format

to fully characterize protein sequence eight feature vectors are extracted from PROFEAT server (Rao *et al.,* 2011) to represent the protein sample, including amino acid composition, dipeptide composition, correlation, composition, transition, distribution of physiochemical properties, sequence order descriptors, and pseudo amino acid composition with total of 1497 features being calculated for the prediction of functional classes and sub-classes of ion channels, enzymes, nuclear and G-protein coupled receptors as shown in Figure 2.2.



Class	X1	X2	X3	X4	X5	X6	X7	X8
amine	0.02388	0.024088	0.026938	0.025756	0.023766	0.022536	0.022249	0.02389
amine	0.024411	0.024122	0.025177	0.019681	0.025192	0.023703	0.026833	0.025259
amine	0.023259	0.023375	0.024175	0.024074	0.026356	0.028273	0.026564	0.022609
amine	0.024698	0.029254	0.026481	0.027016	0.020904	0.022804	0.026271	0.022138
amine	0.027045	0.025997	0.025377	0.02453	0.029516	0.02588	0.024511	0.019895
amine	0.026582	0.028383	0.02426	0.026142	0.024249	0.022603	0.024133	0.018614
amine	0.025118	0.022562	0.018945	0.026461	0.024663	0.023759	0.026297	0.022877
amine	0.02374	0.02204	0.023372	0.023475	0.024111	0.025726	0.021402	0.025918
amine	0.01895	0.018609	0.020915	0.018605	0.020705	0.020152	0.019946	0.017668
amine	0.024196	0.027393	0.02655	0.025476	0.024777	0.024135	0.022779	0.026813
peptide	0.023528	0.022801	0.024892	0.021051	0.024598	0.020805	0.022166	0.02328
peptide	0.021491	0.026945	0.026393	0.021735	0.024233	0.0241	0.023979	0.025827
peptide	0.025294	0.026893	0.02394	0.025287	0.028168	0.023433	0.023411	0.024151
peptide	0.025183	0.027754	0.024691	0.025968	0.02636	0.027292	0.025901	0.027271
peptide	0.021652	0.02268	0.025066	0.026927	0.02437	0.023531	0.022173	0.018932
peptide	0.024981	0.026591	0.02432	0.022246	0.02556	0.027135	0.024212	0.022898
peptide	0.018388	0.02454	0.025754	0.023006	0.022802	0.019873	0.02323	0.021739
peptide	0.024812	0.022508	0.021446	0.023488	0.021991	0.024146	0.022201	0.018992
CAPA	0.023851	0.026643	0.023004	0.022024	0.022519	0.026161	0.024942	0.023058
CAPA	0.025157	0.029155	0.026541	0.025098	0.027692	0.024616	0.022354	0.022991
CAPA	0.025787	0.025047	0.027172	0.026204	0.023926	0.024838	0.019719	0.022419
CAPA	0.025851	0.024609	0.024712	0.021339	0.025748	0.024986	0.023968	0.024735
CAPA	0.0215	0.025447	0.026389	0.020421	0.025476	0.025756	0.019862	0.026953
CAPA	0.019173	0.021679	0.021102	0.022412	0.022182	0.026554	0.022632	0.022342

Figure 2.2: Sequence derived properties of protein sequences

Here, the total 1497 number of sequence derived features are represented as $X_{1,}$ X_2,...........,X_{1497} where first 20 features from X_1 to X_{20} represents amino acid composition, 400 number of features from X_{21} to X_{420} represents dipeptide composition, 720 number of features from X_{421} to X_{1140} represents correlation factors, 21 number of features from X_{1141} to X_{1161} represents composition, 21 number of features from X_{1162} to X_{1182} represents transition, 105 number of features from X_{1183} to X_{1287} represents distribution of physiochemical properties, 160 number of features from X_{1288} to X_{1447} represents sequence order descriptors, and 50 number of features from X_{1448} to X_{1497} represents pseudo amino acid composition of a protein sequences.

Table 2.5: Description of sequence derived properties

S. No.	Features of protein sequences	Total No. of features	Description
1	X_1 to X_{20}	20	Amino acid composition
2	X_{21} to X_{420}	400	Dipeptide composition
3	X_{421} to X_{1140}	720	Correlation factors
4	X_{1141} to X_{1161}	21	Composition
5	X_{1162} to X_{1182}	21	Transition
6	X_{1183} to X_{1287}	105	Distribution of physiochemical properties
7	X_{1288} to X_{1447}	160	Sequence order descriptors
8	X_{1448} to X_{1497}	50	Pseudo amino acid composition

The brief descriptions of these prominent features are given as follows:

2.2.1. Amino acid composition (AAC)

The amino acid composition (AAC) is the fraction of each amino acid in the protein sequence. It is defined as

$$AAC\ (n) = \frac{Number\ of\ amino\ acid\ of\ type\ n}{Length\ of\ amino\ acid\ sequence} \qquad Where\ n = 1\ to\ 20 \qquad (2.1)$$

Total 20 features are calculated corresponding to each amino acid.

2.2.2. Dipeptide composition (DC)

An amino acid composition provides only sequence information but ignore the sequence order information so dipeptide composition (DC) of a protein is used to transform a variable length protein sequence to a fixed 400 feature vectors. It is calculated as

$$\mathrm{DC\ (n)} = \frac{\text{Total number of the n}^{\text{th}}\text{ dipeptide}}{\text{Total number of all possible dipeptide}} \qquad \text{Where n} = 1 \text{ to } 400 \qquad (2.2)$$

2.2.3. Correlation factors (CF)

For a given protein sequence an autocorrelation factors are defined by using the distribution of amino acid properties along the sequence. The normalized value of eight sequence properties such as hydrophobicity, average flexibility, free energy of solution in water, polarizability, residue accessible surface area in tripeptide, residue volume, steric parameter and relative mutability are used to calculate these features.

Let $P_1, P_{2\ldots}$ *and* P_N are the physiochemical property values of 1st residue, 2nd residue… and *n*th residue respectively. So by using these values a protein sequence can be converted as $[P_1, P_2 \ldots P_N]$. The three types of autocorrelation features are computed as follows.

Normalized Moreau- Broto autocorrelation features can be calculated by using

$$\text{MB autocorrelation feature (l)} = \frac{1}{\mathrm{N}-\mathrm{l}}\sum_{\mathrm{i}=1}^{\mathrm{N}-\mathrm{l}} \mathrm{P_i\ P_{i+l}} \qquad \text{where} \quad \mathrm{l} = 1 \text{ to } 30 \qquad (2.3)$$

Moran autocorrelation features are calculated as

$$MACF(l) = \frac{\frac{1}{N-l}\sum_{l=1}^{N-l}(P_i - \bar{P})(P_{i+l} + \bar{P})}{\frac{1}{N}\sum_{i=1}^{N}(P_i - \bar{P})^2} \qquad \text{where} \quad \mathrm{l} = 1 \text{ to } 30 \qquad (2.4)$$

and $\qquad \bar{P} = \frac{\sum_{i=1}^{N} P_i}{N}$

Geary autocorrelation features are calculated as

$$GACF(l) = \frac{\frac{1}{2(N-l)}\sum_{l=1}^{N-l}(P_i - P_{i+l})^2}{\frac{1}{N}\sum_{i=1}^{N}(P_i - \bar{P})^2} \qquad \text{where} \quad \mathrm{l} = 1 \text{ to } 30 \qquad (2.5)$$

and $\qquad \bar{P} = \frac{\sum_{i=1}^{N} P_i}{N}$

So here, 8*30=240 of each correlation features with total of 240*3=720 features will be calculated.

2.2.4. Composition, transition and distribution features (CTD)

For calculating composition, transition and distribution the 20 amino acid of a protein sequence is divided into three groups: polar, neutral and hydrophobicity by using the seven physiochemical properties: hydrophobicity, normal Vander Waals volume, polarity, polarizability, charge secondary structure and solvent accessibility. So for each property value every amino acid is represented by three indexes 1, 2 and 3 according to one of three groups. The composition, transition and distribution features are calculated by using

$$\boldsymbol{Composition} = \frac{number\ of\ i\ in\ the\ encoded\ sequence}{length\ of\ the\ sequence} \qquad i = 1, 2, 3 \qquad (2.6)$$

$$\boldsymbol{Transition} = \frac{number\ of\ dipeptide\ encoded\ as"\ ij"and\ "ji"}{length\ of\ the\ sequence - 1} \qquad ij =' \ 12', '13', '23' \qquad (2.7)$$

Distribution: These features are the distribution of each property for 1st residue, 25% residue, 50% residue, 75% residue and 100% residue respectively for each group in the amino acid sequence.

So here, 7*3 = 21 composition features, 7*3 = 21 transition features and 7*3*5 = 105 distribution features are calculated.

2.2.5. Sequence order descriptors

Sequence order descriptors are calculated from the physicochemical distance matrix between each pair of the 20 amino acids.

Sequence order coupling numbers

The dth rank sequence order coupling number is defined as

$$\tau_d = \sum_{i=1}^{N-d} (d_{i,i+d})^2 \qquad \text{where d = 1 to 30} \qquad (2.8)$$

where $d_{i,i+d}$ are the physicochemical distance between the two amino acids at position i and i+d. N is the length of the protein sequence.

Quasi sequence order descriptors

For each amino acid type, a quasi-sequence order descriptor can be defined as

$$X_r = \frac{f_r}{\sum_{r=1}^{20} f_r + w\sum_{d=1}^{30} \tau_d} \qquad 1 \le r \le 20 \;\; and\, w = 0.1 \qquad (2.9)$$

$$X_d = \frac{w.\tau_{d-20}}{\sum_{r=1}^{20} f_r + w\sum_{d=1}^{30} \tau_d} \qquad 21 \le d \le 50 \quad and\, w = 0.1 \qquad (2.10)$$

where f_r is the normalized occurrence for amino acid type i and w is a weighting factor. Here, 60 numbers of features for sequence order coupling numbers and 100 features of quasi-sequence order features are extracted.

2.2.6. Pseudo amino acid composition (PAAC)

The pseudo amino acid composition is calculated by using the three properties: hydrophobicity (*H1*), hydrophilicity (*H2*) and side chain mass (*M*) of each 20 amino acid to represent the sequence order correlation between all of the most 30 contiguous residues. The correlation between these three properties is calculated as:

$$\theta(R_i, R_j) = \frac{1}{3}\left\{\left[H_1(R_i) - H_1(R_j)\right]^2 + \left[H_2(R_i) - H_2(R_j)\right]^2 + \left[M(R_i) - M(R_j)\right]^2\right\} \qquad (2.11)$$

By using these correlation values a set of sequence order correlated features are calculated as

$$\Theta_\lambda = \frac{1}{\mathrm{N} - \lambda}\sum_{\mathrm{i}=1}^{\mathrm{N}-\lambda} \Theta(\mathrm{R_i}\,, R_{\mathrm{i}+\lambda}) \qquad \text{where} \quad \lambda\; = 1 \text{ to } 30 \qquad (2.12)$$

Let f_n be the normalized occurrence frequency of the 20 amino acid in the protein sequence, a set of 20+ λ pseudo amino acid composition features can be calculated as

$$PAAC_n = \frac{f_n}{\sum_{r=1}^{20} f_r + w\sum_{j=1}^{30} \Theta_j} \qquad 1 \le n \le 20 \;\; and\, w = 0.1 \qquad (2.13)$$

$$PAAC_n = \frac{\Theta_{n-20}}{\sum_{r=1}^{20} f_r + w\sum_{j=1}^{30} \Theta_j} \qquad 21 \le n \le 50 \quad and\, w = 0.1 \qquad (2.14)$$

So, total 50 pseudo amino acid composition features are calculated.

2.3. Feature selection techniques

Feature selection is the process of selecting a best subset of features, among all the features that are useful for the learning algorithms. The goals of feature selection are:

- To provide faster and more cost effective models by reducing the size of the problem and hence reducing computational time and space required to run classifiers.
- To improve the performance of the classifiers, firstly by removing noisy or irrelevant features secondly by reducing the likelihood of overfitting to noisy data. So the basic objective of feature selection algorithms to improve the performance of the classifier, i.e. prediction performance in the case of classification and better cluster detection in the case of clustering.

2.3.1. Filter method

Filter methods assess the relevance of features by looking only at the intrinsic properties of the data. Filter method calculates the relevance score of the features by using the essential properties of the data and low scoring features are removed. It evaluates features in isolation so not consider the correlation between features. Afterwards, this subset of features is presented as input to the classification algorithm. Advantages of filter techniques are that they easily scale to very high-dimensional datasets, they are computationally simple and fast and they are independent of the classification algorithm. As a result, feature selection needs to be performed only once and then different classifiers can be evaluated.

A common disadvantage of filter methods is that they ignore the interaction with the classifier and that most proposed techniques are univariate. This means that each feature is considered separately thereby ignoring feature dependencies which may lead to worse classification performance when compared to other types of feature selection techniques. In order to overcome the problem of ignoring feature dependencies, a number of multivariate filter techniques were introduced, aiming at the incorporation of feature dependencies to some degree.

2.3.2. Wrapper method

The wrapper method uses the classifier for searching the subset of features. It uses the backward elimination process to remove the irrelevant features from the subset of features. In wrapper method the rank of the features is calculated recursively and low rank features are removed from the result. Advantages of wrapper approaches include the interaction between feature subset search and model selection, and the ability to take into account feature dependencies. A common drawback of these techniques is that they have a higher risk of over-fitting than filter techniques and are very computationally intensive, especially if building the classifier has a high computational cost.

2.3.3. Hybrid method

In the hybrid feature selection the search for an optimal subset of features is built into the classifier construction and can be seen as a search in the combined space of feature subsets and hypotheses. Just like wrapper approaches, hybrid methods are thus specific to a given learning algorithm. Hybrid methods have the advantage that they include the interaction with the classification model, while at the same time being far less computationally intensive than wrapper methods. So Instead of choosing one particular feature selection method, and accepting its outcome as the final subset, different feature selection methods can be combined using ensemble feature selection approaches.

2.4. Computational intelligence techniques

This section presents an overview of various computational intelligence techniques used in protein function prediction such as artificial neural network, Naive Bayes classifier, support vector machine, k-nearest-neighbor, decision trees, bagging, boosting, random subspace method and random forests.

2.4.1. Artificial neural network (ANN)

An artificial neural networks (Hagan *et al.,* 1996; Schalkoff, 1997) is inspired by the concept of biological nervous system. ANNs are the collection of computing elements (neurons) that may be connected in various ways. In ANNs the effect of the synapses is represented by the connection weight, which modulates the input signal. The architecture of artificial neural networks is a fully connected, three layered (input layer, hidden layer and output layer) structure of nodes in which information flows from the input layer to the output layer through the hidden layer. ANNs are capable of linear and nonlinear classification. An ANN learns by adjusting the weights in accordance with the learning algorithms. It is capable to process and analyze large complex datasets, containing non-linear relationships. There are various types of artificial neural network architecture that are used in protein function prediction such as perceptron, multi-layer perceptron (MLP), radial basis function networks and kohonen self-organizing maps.

2.4.2. Naive Bayes classifier

Naive Bayes classifier (Keller, 2002) is a statistical method based on Bayes theorem. It calculates the probability of each training data for each class. The class of test data assigns by using the inverse probability. It assumes the entire variables are independent, so only mean and variance are required to predict the class. So the main advantage of the naive Bayes classifier is that it requires a small amount of training data to estimate the mean and variances that are used to predict the class.

2.4.3. Support vector machine (SVM)

Support vector machine (Cortes and Vapnik, 1995) is based on the statistical learning theory. The SVM is capable of resolving linear and non-linear classification problems. The principal idea of classification by support vector is to separate examples with a linear decision surface and maximize the margin of separation between the classes to be classified. SVM works by mapping data with a high-dimensional feature space so that data points can be categorized, even when the data are not otherwise linearly separable. A separator between the categories is found, and then the data are transformed in such a way that the separator could be drawn as a hyperplane. Following this, characteristics of new data can be used to predict the group to which a new record should belong. After the transformation, the boundary between the two categories can be defined by a hyperplane. The mathematical function used for the transformation is known as the kernel function. SVM supports the linear, polynomial, radial basis function (RBF) and sigmoid kernel types. When there is a straightforward linear separation then linear function is used otherwise we used polynomial, radial basis function (RBF) and sigmoid kernel function. Besides the separating line between the categories, a SVM also finds marginal lines that define the space between the two categories. The data points that lie on the margins are known as the support vectors.

2.4.4. k-nearest neighbor (k-NN)

The k-Nearest Neighbors algorithm (Cover and Hart, 1967) is a non-parametric method used for classification and regression. In both cases, the input consists of the k closest training examples in the feature space. The output depends on whether k-NN is used for classification or regression. In k-NN classification, the output is a class membership. The k-NN classifiers are based on finding the k nearest neighbor and taking a majority vote among the classes of these k neighbors to assign a class for the given query. The k-NN is a type of instance based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The k-NN is more efficient for large datasets and robustness when processing noisy data but high computation cost reduces its speed.

2.4.5. Decision trees

The decision trees are a branch test-based classifiers such as such as ID3 (Iterative Dichotomiser 3) (Quinlan, 1996), C 5.0 (Quinlan, 2004), Classification And Regression Tree (CART) (Breiman *et al.,* 1984) and CHi-squared Automatic Interaction Detector (CHAID) (Kass, 1980) etc. These classifiers use the knowledge of training data it creates a decision trees that is used to classify test data. In the decision tree every branch represents a set of classes and a leaf represent a particular class. A decision node identifies a test on a single attribute value with one branch and its subsequent classes represent as class outcomes. To maximize interpretability these classifiers are expressed as decision trees or rule sets (IF-THEN), forms that are generally easier to understand than neural networks. Decision tree based classifiers are easy to use and does not presume any special knowledge of statistics or machine learning.

2.4.6. Bagging

Bagging (Breiman, 1996) is ensemble classifiers. In bagging *'n'* random instances are selected using a uniform distribution (with replacement) from a training dataset of size '*n*'. The learning process starts using these '*n*' randomly selected instances and this process can be repeated several times. Since the selection is with replacement, usually the selected instances will contain some duplicates and some omissions as compared to the original training dataset. Each cycle through the process results in one classifier. After the construction of several classifiers, taking a vote of the predictions of each classifier performs the final prediction.

2.4.7. Boosting

Boosting (Freund and Schapire, 1996) is similar to bagging except that one keeps track of the performance of the learning algorithm and forces it to concentrate its efforts on instances that have not been correctly learned. Instead of selecting the *'n*' training instances randomly using a uniform distribution, one chooses the training instances in such a manner as to favors the instances that have not been accurately learned. After several cycles the prediction is performed by taking a weighted vote of the predictions of each classifier with the weights being proportional to each classifier's accuracy on its training set.

2.4.8. Random subspace method

Random subspace (Ho, 1998) method or attribute bagging (Bryll , 2003) is an ensemble classifier that consists of several classifiers and outputs the class based on the outputs of these individual classifiers. Random subspace method has been used for linear classifiers, support vector machines, nearest neighbors and other types of classifiers. This method is also applicable to one-class classifiers. It is an attractive choice for classification problems where the number of features is much larger than the number of training objects. The ensemble classifier is constructed using the following algorithm:

- Let the number of training objects be *N* and the number of features in the training data be *D.*
- Choose *L* to be the number of individual classifiers in the ensemble.

- For each individual classifier *C*, choose d_c (d_c < D) to be the number of input variables for *C.* It is common to have only one value of d_c for all the individual classifiers

- For each individual classifier *C,* create a training set by choosing d_c features from *D* without replacement and train the classifier.

- For classifying a new object, combine the outputs of the *L* individual classifiers by majority voting or by combining the posterior probabilities.

2.4.9. Random forests

Random forest classifier (Breiman, 2001) used an ensemble of random trees. Each of the random trees is generated by using a bootstrap sample data. At each node of the tree a subset of feature with highest information gain is selected from a random subset of entire features. Thus random forest used bagging as well as feature selection to generate the trees. Once a forest is generated every tree participates in classification by voting to a class. The final classification is based on the majority voting of a particular class. It performs better in comparison with single tree classifiers such as CART and C 5.0 etc.

2.5. Performance measures

The performance of the classifiers is measured by using 10-fold cross validation. In *K*-fold cross validation the dataset of all proteins is partitioned into *K* subsets where one subset is used for validation and remaining *K-1* subsets is used for training. This process is repeated for *K* times so that every subset is used once as a test data. In this thesis, accuracy (*ACC*), receiver operating characteristics (ROC), precision, sensitivity, specificity and Matthew’s correlation coefficient (*MCC*) are used to measure the performance.

Table 2.6: Confusion matrix to measure the performance of the classifiers

	Predicted Class		
	Positive	Negative	Total
Positive	TP	FN	P
Negative	FP	TN	N

The performance of the classifiers is measured by the quantity of True positive (*TP*), True Negative (*TN*), False Positive (*FP*), False Negative (*FN*). Where *TP* (True Positive) is the number of positive instances that are classified as positive, *FP* (False Positive) is the number of negative instances that are classified as positive, *TN* (True Negative) is the number of negative instances that are classified as negative and *FN* (False Negative) is the number of positive instances that are classified as negative. By using these quantities standard accuracy, sensitivity, specificity, precision, *MCC* and *ROC* area performance measures are defined as:

Accuracy: Accuracy is defined as the proportion of instances that are correctly classified.

$$Accuracy = \frac{(TP + TN)}{(P + N)} \qquad (2.15)$$

Sensitivity: Sensitivity is defined as the proportion of positive instances that are correctly classified as positive.

$$Senstivity = \frac{(TP)}{(P)} \qquad (2.16)$$

Specificity: Specificity is defined as the the proportion of negative instances that are correctly classified as negative.

$$Specificity = \frac{(TN)}{(N)} \qquad (2.17)$$

Precision: Precision is defined as the proportion of instances classified as positive that are really positive.

$$Precision = \frac{(TP)}{(TP + FP)} \tag{2.18}$$

Matthew's correlation coefficient (*MCC*): The *MCC* is a balanced measure that considers both true and false positives and negatives. The *MCC* can be obtained as

$$\boldsymbol{MCC} = \frac{(\mathrm{TP})(\mathrm{TN}) - (\mathrm{FP})(\mathrm{FN})}{\sqrt{[\mathrm{TP} + \mathrm{FP}][\mathrm{TP} + \mathrm{FN}][\mathrm{TN} + \mathrm{FP}][\mathrm{TN} + \mathrm{FN}]}} \tag{2.19}$$

Receiver operating characteristics (*ROC*): The *ROC* (Hanley and McNeil, 1982; Worster *et al.*, 2006) is a graph that shows the performance of a classifier by plotting *TP* rate versus FP rate at various threshold settings. Area under *ROC* curve (*AUC*) of a classifier is the probability that the classifier ranks a randomly chosen positive instance higher than a randomly chosen negative instance.

2.6. Conclusion

In this chapter, the theoretical backgrounds related to protein function prediction were presented. It presented the literature reviews for the computational intelligence techniques used in prediction of ion channels, enzymes, nuclear and G-protein coupled receptors. The features extracted from protein sequences that were used in the prediction of protein function were described in this chapter. The basic concepts related to feature selection techniques such as filter, wrapper and hybrid methods and various computational intelligence techniques such as artificial neural network, Naive Bayes classifier ,support vector machine, k-nearest-neighbor, decision trees, bagging, boosting, random subspace method and random forests were presented. In the last section the performance measure of the classifier was presented.