**RESEARCH ARTICLE**

# PRGPred: A platform for prediction of domains of resistance gene analogue (RGA) in Arecaceae developed by using machine learning algorithms

**Mathodiyil S. Manjula** [1]
**Kaitheri E. Rachana** [1]
**Sudalaimuthu. Naganeeswaran** [1]
**Nambisan Hemalatha** [2]
**Anitha Karun** [1]
**Muliyar K. Rajesh** [1]

***Authors' addresses:***
[1] *ICAR - Central Plantation Crops Research Institute, Kasaragod 671124, Kerala, India.*
[2] *AIMIT, St. Aloysius College, Mangalore 575002, Karnataka, India*.

***Correspondence:***
*Nambisan Hemalatha*
*AIMIT, St. Aloysius College, Mangalore 575002, Karnataka, India*.
Tel.: +91-94955870533
e-mail: hemasreen71aimit@gmail.com

***Article info:***
*Received:* 30 June 2015
*Accepted:* 14 August 2015

## ABSTRACT

Plant disease resistance genes (*R*-genes) are responsible for initiation of defense mechanism against various phytopathogens. The majority of plant *R*-genes are members of very large multi-gene families, which encode structurally related proteins containing nucleotide binding site domains (NBS) and C-terminal leucine rich repeats (LRR). Other classes possess' an extracellular LRR domain, a transmembrane domain and sometimes, an intracellular serine/threonine kinase domain. R-proteins work in pathogen perception and/or the activation of conserved defense signaling networks. In the present study, sequences representing resistance gene analogues (RGAs) of coconut, arecanut, oil palm and date palm were collected from NCBI, sorted based on domains and assembled into a database. The sequences were analyzed in PRINTS database to find out the conserved domains and their motifs present in the RGAs. Based on these domains, we have also developed a tool to predict the domains of palm *R*-genes using various machine learning algorithms. The model files were selected based on the performance of the best classifier in training and testing. All these information is stored and made available in the online 'PRGpred' database and prediction tool.

**Key words:** RGA, PRINTS, SVM, WEKA, HMMER, BLAST

## Introduction

During the course of evolution, plants have developed numerous and diverse mechanisms to defend themselves against pathogenic microorganisms. The plant defense system encodes a wide array of pathogen receptors encoded by plant disease resistance (*R*) genes, which possess the ability to perceive a pathogen attack and facilitate a counter attack across the pathogen (Spoel & Dong, 2012). The resistance *R*-genes carry three key properties: pathogen recognition, signaling cascade activation, and a capacity for the rapid evolution of specificity (Koczyk & Chelkowski, 2003). *R*-genes have been cloned and characterized in a number of plants and based on the molecular structure of their encoded proteins, they have been classified into nucleotide binding leucine rich repeats proteins (NLR), receptor-like proteins (RLP), receptor-like kinases (RLK), including LRR-kinases and lectin receptor kinases, and intracellular kinase-like protein (PK) (Vossen et al., 2013).

The main classes of *R*-genes consist of a nucleotide binding site domain (NBS) and a leucine-rich repeat (LRR) domain and are often referred to as NBS-LRR *R*-genes. Normally, the NBS domain binds either ATP or GTP and the LRR domain is often involved in protein-protein interactions and ligand binding. NBS-LRR *R*-genes are subdivided into toll interleukin-1 receptor (TIR-NBS-LRR) and coiled-coil (CC-NBS-LRR) (Meyers et al., 2003). The LRR domain in R-proteins negotiates direct or indirect interaction with pathogen molecules (Dangl & Jones, 2001). Several highly conserved motifs are present in the NBS region of

**RESEARCH ARTICLE**

characterized *R*-genes and *R*-gene analogs (RGAs): P-loop, kinase-2, kinase-3a and GLPL (also called "hydrophobic domain"), a putative membrane spanning domain (Baldi et al., 2004). The conserved domains of *R*-genes provide opportunities for design of degenerate primers and isolating resistance gene analogues (RGAs) by the polymerase chain reaction (PCR) strategy from plant genomes. By this approach, NBS-LRR types RGAs have been isolated from very diverse species of plants (Vossen et al., 2013). RGA sequences have also been used as molecular markers for identifying the disease resistance loci in a number of plants (Vossen et al., 2013).

Proteins encoded by *R*-genes have been shown to possess modular domains and dynamic interaction between these modular domains is essential to achieve protein functionality (van Ooijen et al., 2007). Protein domains are structural, evolutionary and functional units of proteins and their successful prediction from the sequence information can assist prediction of protein tertiary structure, annotation of protein functions and determination of protein structure. With the rapid accumulation of protein sequences generated in the present genomics era, it is imperative to develop effective methods/tools to predict protein domains based solely on sequence information. However, predicting protein domain boundaries from a sequence is a challenging area of research and, therefore, faster computational techniques (statistical or machine learning) could be applied in protein domain prediction. Efforts in this direction can be classified into template-based method's, *ab initio* methods and a combination of both (Li et al., 2012) and a few software programs and web-server tools developed for predicting protein domains include FIEFDom (Bondugula et al., 2009), DoMpro (Cheng et al., 2006), DROP (Ebina et al., 2011), DomCut (Suyama & Ohara, 2003) and Globplot (Linding et al., 2003). Using a combination of techniques viz. random forest, maximum relevance minimum redundancy and incremental feature selection, and incorporating features of physicochemical and biochemical properties, sequence conservation, residual disorder, secondary structure, and solvent accessibility, a new method of protein domain prediction has also been developed recently (Li et al., 2012). Machine learning algorithms have also been utilized to predict sub-cellular localization of proteins (Chou & Cai, 2002).

In the present study, we have utilized machine learning algorithms for prediction of domains within RGA sequences in palms, in which genome information, previously scarce, are being rapidly added to the public domain since the last couple of years. Breeding for durable resistance is one of the major objectives of improvement programmes of cultivated palms and identification of and knowledge of RGA in palms can significantly aid palm breeding programmes. The database created in this study provides researchers a comprehensive collection of manually curated *R*-genes in four economically important palms *viz*. coconut, oil palm, date palm and arecanut. A tool for prediction of different domains has also been developed using machine learning algorithms. A freely available web interface has also been implemented and hosted.

## Materials and Methods

### *Assembly of the datasets and detection of conserved domains within them*

In the present study, the positive data set consisted of sequences representing resistance gene analogues (RGAs) of coconut, arecanut, oil palm and date palm, which were either generated from our lab (via RNA-Seq) or obtained from NCBI (http://www.ncbi.nlm.nih.gov/). The negative data set was developed from non-RGA protein sequences of palms. The positive and negative sequences were divided for training and testing. PRINTS (http://www.bioinf.manchester.ac.uk/dbbrowser/PRINTS/index.php) is a compendium of protein motif 'fingerprints' (Attwood et al., 1994), which is a collection of aligned motifs separate from conserved regions of a sequence alignment. The palm RGA sequences collected from NCBI were analyzed in PRINTS database to find out the conserved domains and their motifs present in the RGAs.

### *Support vector machine (SVM)*

Support vector machine is a computer algorithm for learning classification and regression rules from data, which was developed by from statistical learning theory for data classification (Vapnik, 1999). In the present study, we have used SVM$^{light}$ (Joachims, 1999), a freely downloadable package of SVM to predict the domains of RGA proteins. SVM is based on the supposition that the best separating hyperplanes between two classes is the one with the largest distance to instances of the different classes. It aims at maximizing the margin between two classes and hence is also called maximum margin classifier.

The separating hyperplane is defined by all possible vectors x that satisfy:

**RESEARCH ARTICLE**

$$f(x) = (w.x) + b = 0$$

where $w \epsilon R_k$ is a weight vector and $b \epsilon R$ is the offset of the hyperplane. The predicted class label ŷ is assigned with the result of the following decision function sgn(f(x)). The margin of the hyperplane is defined as the distance of an instance, which is closest to the hyperplane and this kind of optimization problem can be solved only if the training data is perfectly linearly separable. But this is not possible always and in many real world scenarios, classes overlap in the input space resulting in linearly inseparable case. To surmount this problem, a soft margin SVM, in contrast to the hard margin SVM presented above, was introduced, which allows for a certain degree of imperfect separation (Vapnik, 1999). Every instance is assigned a slack variable $\xi_i \geq 0$, which measures the distance of an instance which lies on the wrong side of the separating hyperplane to the borders of the margin. To find a good trade-off between a large margin and small loss, an optimization problem is defined as:

$$\min_{w\epsilon\mathbb{R}^k, b\mathcal{E}\mathbb{R}, \xi\in\mathbb{R}_0^{+n}} \frac{1}{2}||w||^2 + C\sum_{i=1}^{n}\xi_i$$

subject to $y_i\big((w, x_i) + b\big) \geq 1 \qquad \forall(x_i, y_i)\epsilon D$

where C is the cost parameter, also called as the tradeoff factor.

In addition to instances on the borders of the margin, instances with a slack variable ≥ 0 are called support vectors. Linear SVMs can be extended by introducing a dual representation of the classification problem. The separating hyperplane of an SVM can also be described as a linear combination of support vectors which are represented as:

$$w = \sum_{i=1}^{n} \propto_i y_i x_i$$

where $\alpha_i \geq 0$ are called Lagrange multipliers. Using this representation, margin maximization results in the dual optimization problem:

$$max \sum_{i=1}^{n} \propto_i - \frac{1}{2}\sum_{i=1}^{n} \propto_i \propto_j y_i y_J (x_i x_j)$$

subject to $\sum_{i=1}^{n} \propto_i y_i = 0, \propto_i \geq 0 \quad \forall i = 1, \ldots\ldots n.$

Using the dual representation of w, the predicted class label ŷ is determined by the following decision function:

$$f(x) = sgn(\sum_{i=1}^{n} \propto_i (x_i x_j)$$

An important advantage of the dual representation of an SVM is that the hyperplane which is only defined by inner products $(x_i, x_j)$ can be replaced by kernel function or kernel:

$$k(x_i x_j) = (\varphi(x_i), \varphi(x_i),$$

where function φ maps the input space into a higher dimensional space. As long as the kernel maps the data into an inner product space, an explicit mapping function φ is not very much necessary and this mapping is often referred to as the kernel trick. Therefore, if the data is not linear, we can completely map the data into a higher-dimensional space where a linearly separating hyperplane exist without actually transforming the data. Thus, the decision boundary in the original space is nonlinear instead of linear. The popular kernels used with SVM are the linear kernel, polynomial kernel and the Radial Basis Function (RBF) which has different properties. Various kernels based on strings, graphs, and other input types have also been developed to tackle problems in computational biology. The selection of a kernel is an important part of SVM training and classification since a properly designed kernel function can lessen generalization error, hasten convergence speed and enhance the prediction accuracy. In this study, the experimentation was conducted using various types of kernels such as polynomial, radial basis function (RBF) and sigmoid.

***WEKA***

WEKA (Waikato Environment for Knowledge Analysis) is a well-known suite of machine learning software written in Java (http://www.cs.waikato.ac.nz/ml/weka/). It is a cluster of machine learning algorithms for data mining tasks, which includes tools for data pre-processing, regression, clustering, association rules, classification and visualization. It is also applicable for developing new machine learning schemes. In this study, different types of model files were developed in WEKA by training and testing procedures. Training and testing were conducted in WEKA with the help of WEKA classifiers. We have selected 11 classifiers (*viz.*, Naïve Bayes, Logistic, Multilayer Perceptron, RBF Network, Simple Logistic, Voted Perceptron, IB1, IBk, KStar, J48 and Random Forest) for this work.

***Features of different methods***

Domains were encoded by different features, which are briefly explained below:

*Amino-acid composition*

Amino-acid composition is the fraction of each amino acid which are present in a given protein sequence.

**RESEARCH ARTICLE**

The fraction of all the natural 20 amino acids is calculated using the following equation:

$$\text{Fraction of amino acid (n)} = \frac{\text{Total number of amino acids n}}{\text{Total number of amino acids in the sequence}} \quad \text{(Equation 1)}$$

where n can be any amino acid.

*Dipeptide composition*

The total number of amino acids is 20 and therefore, the theoretical number of possible dipeptide is 400. The information about the fraction of amino acids as well as their local order can be identifying by using dipeptide composition. The dipeptide composition of each protein is calculated using the equation:

$$\text{Fraction of dipeptide (n+1)} = \frac{\text{Total number of dipeptides (n+1)}}{\text{Total number of all possible dipeptides}} \quad \text{(Equation 2)}$$

where dipeptide (n+1) is one of the 400 dipeptides.

*Hybrid composition*

The hybrid composition was developed by combining amino-acid composition and dipeptide composition features of a protein sequence and calculated by combining Equations (1) and (2). The WEKA input vector pattern of 420 was created *i.e.* 20 for amino acid and 400 for dipeptide composition.

***Assessment of accuracy of the classifiers using validation schemes***

Validation schemes are often used to evaluate and assess unbiased prediction accuracy of any machine learning classifier. In statistical prediction, two validation approaches are generally used to examine a predictor for its effectiveness in practical applications are independent dataset test and cross validation. In this study, both techniques have been used for SVM and WEKA algorithms. In cross-validation techniques, data is divided into two sets randomly. The first set ('training set') is used in training the learning method, whereas the second set ('test set') is used for subsequent evaluation of the accuracy of the trained method. This tests the ability of the method to generalize and make predictions on unknown data. In 10-fold cross-validation, the data set is randomly divided into ten subsets, each containing an equal number of proteins. In each iteration, one portion of the data is left out as the test data set. The remaining 9 portions (10-1) are used as the training data to construct a classifier. The classifier is then applied to the left-out test portion to assess the prediction accuracy. The procedure is repeated 10 times until all 10 portions are evaluated by the cross validation. Each sample is evaluated exactly once and the total prediction accuracy is then calculated. In the independent dataset test, none of the data to be tested occurs in the training dataset used for training the predictor and selection of data for testing could be quite arbitrary.

***Confusion Matrix and the measures of performance***

Typically, while considering a two-class problem, there are instances which are both negative and positive. To train a classifier $f(x|\phi)$ onto a training set, it is envisaged that x taken from a validation set will be positive when $f(x|\phi) \geq \theta$, for a threshold $\theta$. it is also envisaged that $f(x|\phi) \in [0,1]$ approximates the posterior probability that x will be a positive example, that is, $P(+|x) = f(x|\phi)$ and that x will be a negative one if $f(x|\phi) < \theta <$ and $P(-|x) = - f(x|\phi)$. Based on the exact label of x comprising the confusion matrix, there will be four cases and their numbers of occurrences are counted across the entire validation sets (Figure 1).

| | p' (Predicted) | n' (Predicted) |
|---|---|---|
| P (Actual) | True Positive | False Negative |
| n (Actual) | False Positive | True Negative |

**Figure 1.** *Confusion matrix*

True positive (tp): These include the number of instances when the class labels and the predicted classes both appear to be positive.

False negative (fn): These include the number of instances when the class labels are positive, but the predicted classes appear to be negative.

False positive (fp): These include the number of instances when the class labels are negative, but the predicted classes appear to be positive.

True negative (tn): These include the number of instances when the class labels and the predicted classes both appear to be negative.

**RESEARCH ARTICLE**

Different performance measures are calculated from these four values:

$$\text{Sensitivity} = \frac{TP}{TP+FN} \text{ X } 100 \quad \text{(Equation 3)}$$

$$\text{Specificity} = \frac{TN}{FP+FN} \text{ X } 100 \quad \text{(Equation 4)}$$

$$\text{Precision} = \frac{TP}{TP+FP} \text{ X } 100 \quad \text{(Equation 5)}$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+FN} \text{ X } 100 \quad \text{(Equation 6)}$$

$$\text{MCC} = \frac{(TP \text{ X } TN)-(FP \text{ X } FN)}{\sqrt{(TP + FP)(TP + FN)(TN+FP)(TN+FN)}} \text{ X } 100 \quad \text{(Equation 7)}$$

We have adopted five frequently considered measurements: Accuracy (Ac), Sensitivity (Sn), Specificity (Sp), Precision (Pr) and Mathew's Correlation Coefficient (MCC). Sensitivity and specificity are measures suitable for assessing the performances of a novel classifier. The sensitivity (Sn) and specificity (Sp) exhibit accurate prediction ratios of positive (+) and negative data (-) sets respectively (Eq. 3 and 4). Precision constitutes the proportion of the predicted positive occurrences which are accurate (Eq. 5). Accuracy (Ac) illustrates the exact ratio involving the positive (+) and negative (-) data sets (Eq. 6). Mathew's Correlation Coefficient's (MCC) are incorporated to calculate the prediction performance of the developed algorithm (Eq.7). The MCC contributes a balanced measure between specificity and sensitivity for each of the classes. The MCC value ranges from -1 to 1, and a larger MCC value occupies for better prediction performance. A best classification technique should possess specificity and sensitivity values approximately 100% and also a MCC value which is equal to 1.

***Performance curves (ROC) and the Area under Curves (AUC)***

The The comparative weight of a false negative and a false positive decides the values of threshold θ and θ = 0.5 is used when both of them possess equal weights and θ is greater whenever a false positive comprises a elevated weight t than a false negative. If the exact weights are not known, if it required to vary the value of θ to observe the variations in performance measures as θ is varied. Afterwards, the performance are plotted as a function of θ to decipher the total behavior. A ROC curve or receiver operating characteristics curve can be said to be a plot representing tp-rate (hit rate) and fp-rate (false alarm rate). ROC takes into consideration the performances of two-class classifiers and verifies for superior performances on both positives and negative instances.

If the curves are complicated, it will be difficult to make a comparison of these two curves. One of the techniques to sum up a curve is to calculate the area under the curve (AUC), which is approximated by accounting for the trapezoidal areas created by consecutive points on the ROC (Tom, 2006).

***HMMER***

HMMER is sequence alignment software based on a statistical framework using profile hidden Markov models (HMM) (Durbin et al., 1998). Profile-HMMs are designed from statistical models of multiple sequence alignments in the HMMER package using the 'hmmbuild' program. The profile-HMM implementation (Krogh, 1994) was used in the study and 'hmmbuild' and 'hmmsearch' programs were used for the prediction of RGAs.

***Stand Alone BLAST***

BLAST is an algorithm used for the comparison of protein or nucleotide sequences from the same or different organisms. Standalone BLAST executable may be identified on the NCBI anonymous FTP server (ftp://ftp.ncbi.nih.gov) under/blast/executable/. In this study, we have used makeblastdb program for building the database and blastp program was used for search the protein sequences against our database.

***Web-server***

We have implemented a web server that allows the user to recognize domains of palm resistance gene analogues from primary amino acid sequences. The tool was developed in Perl and web interface in PHP and HTML to asses user queries.

## Results and Discussion

Among the 338 palm RGA sequences, retrieved from NCBI and analyzed, four domains and 428 motifs were detected. The sequences were sorted based on the four domains *viz.*, DISEASERESIST, TYRKINASE, LEURICHRPT and INTERLEUKIN and a database was created. This represents the largest and the only manually curated palm RGA resource. These sequences constituted the

**RESEARCH ARTICLE**

positive dataset. An equal number of sequences representing non-RGA proteins from palms, retrieved from NCBI, constituted the negative dataset. The positive and negative sequences were divided for training and testing. Machine learning algorithms were then utilized to predict the different domains in the palm RGA sequences and we have utilized independent data test validation and 10-fold cross-validation to evaluate the performance of these machine learning algorithms used for the prediction. A comparison of results derived from cross-validation and independent data test revealed results obtained from cross-validation results to be better compared to independent data test. From Tables 1 and 2, it is clear that the module developed in SVM by amino acid composition method achieved 100% accuracy in all four domains by using RBF kernel. Earlier reports also indicate the suitability of RBF kernels in delivering better performances (Shao & Chang, 2005) and to provide good estimated error rates (Witten & Frank, 2000).

**Table 1.** *Validation of independent data test results for four domains of RGA proteins with SVM*

| Composition | Domains | Algorithm | Sn (%) | Sp (%) | Ac (%) | Pr (%) | MCC |
|---|---|---|---|---|---|---|---|
| Amino acid | DISEASERSIST | Polynomial | 100 | 100 | 100 | 100 | 1 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |
| Amino acid | LEURICHRPT | Polynomial | 100 | 100 | 100 | 100 | 1 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |
| Amino acid | INTERLEUKIN | Polynomial | 100 | 100 | 100 | 100 | 1 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |
| Amino acid | TYRKINASE | Polynomial | 100 | 100 | 100 | 100 | 1 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |
| Dipeptide | DISEASERSIST | Polynomial | 72 | 100 | 86 | 100 | 0.75 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |
| Dipeptide | LEURICHRPT | Polynomial | 50 | 100 | 75 | 100 | 0.58 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |
| Dipeptide | INTERLEUKIN | Polynomial | 56 | 100 | 78 | 100 | 0.62 |
| | | RBF | 75 | 100 | 88 | 100 | 0.78 |
| | | Sigmoid | 63 | 100 | 81 | 100 | 0.67 |
| Dipeptide | TYRKINASE | Polynomial | 91 | 100 | 95 | 100 | 0.91 |
| | | RBF | 78 | 100 | 89 | 100 | 0.8 |
| | | Sigmoid | 88 | 100 | 94 | 100 | 0.88 |
| HYBRID | DISEASERSIST | Polynomial | 96 | 100 | 98 | 100 | 0.96 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |
| HYBRID | LEURICHRPT | Polynomial | 91 | 100 | 95 | 100 | 0.91 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |
| HYBRID | INTERLEUKIN | Polynomial | 75 | 100 | 87 | 100 | 0.78 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |
| HYBRID | TYRKINASE | Polynomial | 100 | 100 | 100 | 100 | 1 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |

**RESEARCH ARTICLE**

**Table 2.** *Validation of 10-fold cross validation results for four domains of RGA proteins with SVM*

| Composition | Domains | Algorithm | Sn (%) | Sp (%) | Ac (%) | Pr (%) | MCC |
|---|---|---|---|---|---|---|---|
| Amino acid | DISEASERSIST | Polynomial | 100 | 100 | 100 | 100 | 1 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |
| Amino acid | LEURICHRPT | Polynomial | 100 | 100 | 100 | 100 | 1 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |
| Amino acid | INTERLEUKIN | Polynomial | 100 | 100 | 100 | 100 | 1 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |
| Amino acid | TYRKINASE | Polynomial | 100 | 100 | 100 | 100 | 1 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |
| Dipeptide | DISEASERSIST | Polynomial | 100 | 100 | 100 | 100 | 1 |
| | | RBF | 83 | 79 | 81 | 80 | 0.63 |
| | | Sigmoid | 98 | 100 | 99 | 100 | 0.98 |
| Dipeptide | LEURICHRPT | Polynomial | 100 | 100 | 100 | 100 | 1 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 80 | 90 | 83 | 0.82 |
| Dipeptide | INTERLEUKIN | Polynomial | 97 | 100 | 98 | 100 | 0.97 |
| | | RBF | 97 | 100 | 98 | 100 | 0.97 |
| | | Sigmoid | 97 | 100 | 98 | 100 | 0.97 |
| Dipeptide | TYRKINASE | Polynomial | 100 | 100 | 100 | 100 | 1 |
| | | RBF | 91 | 48 | 69 | 64 | 0.43 |
| | | Sigmoid | 95 | 100 | 97 | 100 | 0.95 |
| HYBRID | DISEASERSIST | Polynomial | 100 | 100 | 100 | 100 | 1 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |
| HYBRID | LEURICHRPT | Polynomial | 100 | 100 | 100 | 100 | 1 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |
| HYBRID | INTERLEUKIN | Polynomial | 97 | 100 | 99 | 100 | 0.97 |
| | | RBF | 97 | 100 | 99 | 100 | 0.97 |
| | | Sigmoid | 97 | 100 | 99 | 100 | 0.97 |
| HYBRID | TYRKINASE | Polynomial | 100 | 100 | 100 | 100 | 1 |
| | | RBF | 100 | 100 | 100 | 100 | 1 |
| | | Sigmoid | 100 | 100 | 100 | 100 | 1 |

The performance of the dipeptide composition-based module and hybrid method were compared with that of an amino acid composition-based module developed on the same dataset by using WEKA. The performances of the different types of composition-based modules of independent data set and cross validation from four domains, implemented using WEKA work bench, are shown in Tables 3 and 4. In the 10-fold cross-validation test from WEKA, the best overall sensitivity was achieved from amino acid composition-based module which had 100% accuracy over the other composition methods. Among the 11 classifiers chosen after an initial screening, Naïve-Bayes, Simple Logistic and IB1 produced the best results and only these three classifiers were chosen for further analysis. Naïve-Bayes achieved an overall prediction accuracy of 100 % with a high-confidence MCC of 1 for all the domains. Naïve-Bayes is a simple but effective classifier. Although its conditional independence assumption is often violated, it performs surprisingly well in classification (Domingos & Pazzani, 1997).

**RESEARCH ARTICLE**

**Table 3.** *Validation of independent data test results for four domains of RGA proteins with modules in WEKA*

| Composition | Domains | Algorithm | Sn (%) | Sp (%) | Ac (%) | Pr (%) | MCC |
|---|---|---|---|---|---|---|---|
| Amino acid | DISEASERSIST | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 98 | 100 | 99 | 100 | 0.97 |
| | | IB1 | 100 | 100 | 100 | 100 | 1 |
| Amino acid | LEURICHRPT | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 100 | 100 | 100 | 100 | 1 |
| | | IB1 | 100 | 100 | 100 | 100 | 1 |
| Amino acid | TYRKINASE | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 99 | 100 | 99 | 100 | 0.99 |
| | | IB1 | 100 | 100 | 100 | 100 | 1 |
| Amino acid | INTERLEUKIN | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 63 | 100 | 81 | 100 | 0.67 |
| | | IB1 | 94 | 100 | 97 | 100 | 0.94 |
| Dipeptide | DISEASERSIST | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 87 | 100 | 93 | 100 | 0.88 |
| | | IB1 | 76 | 100 | 88 | 100 | 0.78 |
| Dipeptide | LEURICHRPT | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 75 | 100 | 87 | 100 | 0.77 |
| | | IB1 | 37 | 100 | 69 | 100 | 0.48 |
| Dipeptide | TYRKINASE | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 89 | 100 | 95 | 100 | 0.9 |
| | | IB1 | 93 | 100 | 96 | 100 | 0.93 |
| Dipeptide | INTERLEUKIN | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 37 | 100 | 69 | 100 | 0.48 |
| | | IB1 | 25 | 100 | 62 | 100 | 0.37 |
| HYBRID | DISEASERSIST | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 98 | 100 | 99 | 100 | 0.98 |
| | | IB1 | 80 | 100 | 90 | 100 | 0.82 |
| HYBRID | LEURICHRPT | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 100 | 100 | 100 | 100 | 1 |
| | | IB1 | 50 | 100 | 75 | 100 | 0.58 |
| HYBRID | TYRKINASE | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 99 | 100 | 99 | 100 | 0.99 |
| | | IB1 | 95 | 100 | 97 | 100 | 0.95 |
| HYBRID | INTERLEUKIN | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 62 | 100 | 81 | 100 | 0.67 |
| | | IB1 | 25 | 100 | 62 | 100 | 0.38 |

**RESEARCH ARTICLE**

**Table 4.** *Comparison of the prediction performance of modules in WEKA from four domains using 10-fold cross validation*

| Composition | Domains | Algorithm | Sn (%) | Sp (%) | Ac (%) | Pr (%) | MCC |
|---|---|---|---|---|---|---|---|
| Amino acid | DISEASERSIST | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 100 | 100 | 100 | 100 | 1 |
| | | IB1 | 100 | 100 | 100 | 100 | 1 |
| Amino acid | LEURICHRPT | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 98 | 100 | 99 | 100 | 0.97 |
| | | IB1 | 100 | 100 | 100 | 100 | 1 |
| Amino acid | TYRKINASE | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 100 | 100 | 100 | 100 | 1 |
| | | IB1 | 100 | 100 | 100 | 100 | 1 |
| Amino acid | INTERLEUKIN | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 95 | 100 | 98 | 100 | 0.95 |
| | | IB1 | 100 | 100 | 100 | 100 | 1 |
| Dipeptide | DISEASERSIST | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 94 | 100 | 97 | 100 | 0.94 |
| | | IB1 | 63 | 100 | 82 | 100 | 0.68 |
| Dipeptide | LEURICHRPT | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 89 | 100 | 95 | 100 | 0.9 |
| | | IB1 | 25 | 100 | 62 | 100 | 0.38 |
| Dipeptide | TYRKINASE | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 98 | 100 | 99 | 100 | 0.98 |
| | | IB1 | 55 | 100 | 73 | 100 | 0.57 |
| Dipeptide | INTERLEUKIN | Naïve Bayes | 90 | 100 | 95 | 100 | 0.9 |
| | | Simple Logistic | 45 | 100 | 72 | 100 | 0.54 |
| | | IB1 | 85 | 100 | 92 | 100 | 0.86 |
| HYBRID | DISEASERSIST | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 100 | 100 | 100 | 100 | 1 |
| | | IB1 | 69 | 100 | 85 | 100 | 0.72 |
| HYBRID | LEURICHRPT | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 98 | 100 | 99 | 100 | 0.98 |
| | | IB1 | 46 | 100 | 73 | 100 | 0.54 |
| HYBRID | TYRKINASE | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 100 | 100 | 100 | 100 | 1 |
| | | IB1 | 92 | 100 | 96 | 100 | 0.93 |
| HYBRID | INTERLEUKIN | Naïve Bayes | 100 | 100 | 100 | 100 | 1 |
| | | Simple Logistic | 100 | 100 | 100 | 100 | 1 |
| | | IB1 | 20 | 100 | 60 | 100 | 0.33 |

**RESEARCH ARTICLE**

In this study we used ROC plot analysis to differentiate the prediction performance for individual locations (Swets, 1988; Zweig & Campbell, 1993). ROC curve shows the tradeoff between sensitivity and specificity. An appropriate predictor will produce a curve along the top and left boundary of the square and will get a score of one. Performance comparisons of overall accuracies achieved by amino acid composition method using SVM in four different domains are represented graphically in Figure 2. The ROC curves of each location for our best classifier using WEKA are presented in Figure 3. The Figures 2 and 3 also depict "excellent classification" area under the curve (AUC=1) value for SVM and WEKA algorithms and high confidence AUCs for all other compositions (Hosmer & Lemeshow, 2000). ROC graphs have long been used in signal detection theory to illustrate the relationship between hit rates and false alarm rates of classifiers (Egan, 1975; Swets et al., 2000).

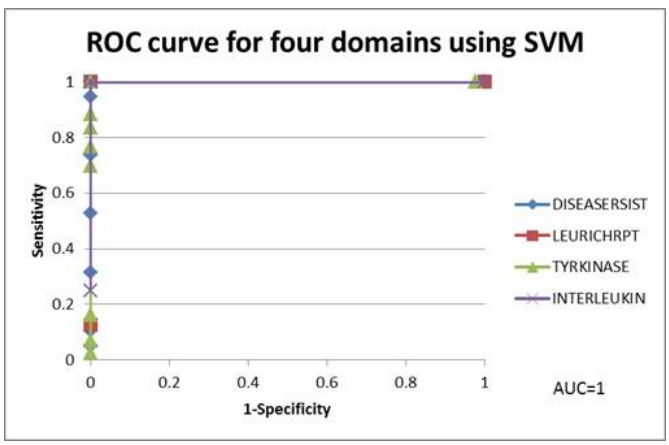


**Figure 2.** *ROC curve for four domains in SVM using independent test results.*

We have also used BLAST and HMMER methods to predict the RGA proteins of coconut, arecanut, oil palm and date palm and compare them with SVM and WEKA-based modules. The accuracy and MCC of the BLAST-based module were 95% and 0.9 for DISEASERSIST, LEURICHRPT achieved maximum accuracy of 85% with MCC of 0.73, TYRKINASE achieved maximum accuracy of 90% with MCC of 0.82 and INTERLEUKIN achieved maximum accuracy of 80% with MCC of 0.66, which were significantly lower than the SVM-based and WEKA-based modules. In the case of HMMER, DISEAESRSIST achieved 85% accuracy with MCC of 0.73, LEURICHRPT achieved 75% accuracy with MCC of 0.58, TYRKINASE achieved 80% accuracy with MCC of 0.65 and INTERLEUKIN achieved 70% accuracy with MCC of 0.33 respectively.

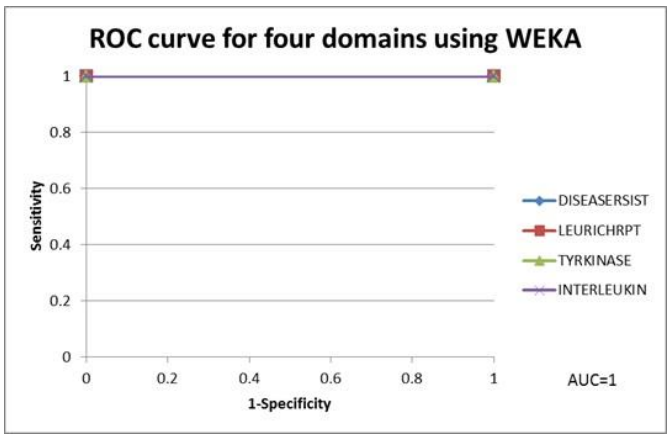


**Figure 3.** *ROC curve for four domains in WEKA using independent test results.*

The results show that the BLAST and HMM-based methods are able to discriminate between all the domains only with around 70% accuracy. Apparently, the performance of SVM- based classifier should be more appropriate than HMM based classifier since the former one is based on the principle of structural risk minimization (Justino et al., 2005). These results suggest that similarity-based search tools alone cannot be efficient and reliable as compared to machine learning algorithms. The performance comparisons of four methods for independent data set are shown in the Figure 4 and Table 5.

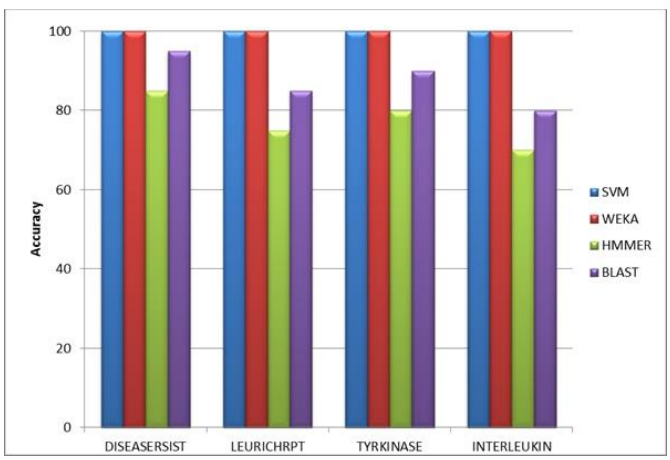


**Figure 4.** *Comparison of the prediction performances of the four palm RGA domains using different algorithms.*

**RESEARCH ARTICLE**

**Table 5.** *Comparison of the prediction performance of four domains using different methods*

| Algorithms | Domains | Sn (%) | Sp (%) | Ac (%) | Pr (%) | MCC |
|---|---|---|---|---|---|---|
| SVM | DISEASERSIST | 100 | 100 | 100 | 100 | 1 |
| | LEURICHRPT | 100 | 100 | 100 | 100 | 1 |
| | TYRKINASE | 100 | 100 | 100 | 100 | 1 |
| | INTERLEUKIN | 100 | 100 | 100 | 100 | 1 |
| WEKA | DISEASERSIST | 100 | 100 | 100 | 100 | 1 |
| | LEURICHRPT | 100 | 100 | 100 | 100 | 1 |
| | TYRKINASE | 100 | 100 | 100 | 100 | 1 |
| | INTERLEUKIN | 100 | 100 | 100 | 100 | 1 |
| BLAST | DISEASERSIST | 100 | 90 | 95 | 91 | 0.90 |
| | LEURICHRPT | 70 | 100 | 85 | 100 | 0.73 |
| | TYRKINASE | 80 | 100 | 90 | 100 | 0.82 |
| | INTERLEUKIN | 100 | 60 | 80 | 71 | 0.66 |
| HMMER | DISEASERSIST | 70 | 100 | 85 | 100 | 0.73 |
| | LEURICHRPT | 50 | 100 | 75 | 100 | 0.58 |
| | TYRKINASE | 60 | 100 | 80 | 100 | 0.65 |
| | INTERLEUKIN | 40 | 100 | 70 | 100 | 0.33 |

Based on our study, we have developed a web server on the World Wide Web as a dynamic web server 'PRGpred'. The overall architecture of the PRGpred web server is shown in the Figure 5. The home page of the tool is given in Figure 6. Users can submit sequence for prediction using cut-and-paste options or file uploading. The prediction result will be displayed in a user friendly format on the screen within few seconds. The output returns the sequence ID and number of motifs present in the given sequence. PRGpred is freely available at http://prgpred.cpcribioinformatics.in/prgpred/.

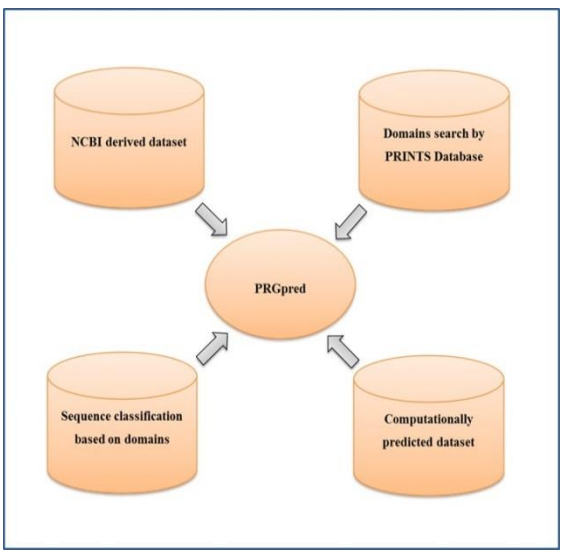


**Figure 5.** *A schematic representation of data mining flow of the PRGpred web server.*


**Figure 6.** *An overview of the home page of PRGpred.*

## Conclusion

We have designed efficient and powerful modules for RGA protein domain prediction in palms using various machine learning algorithms. The best performing modules have been implemented into a web-server, with a database and RGA-domain prediction tool, which has an user-friendly interface. We hope that the tool will aid researchers in the area of resistance breeding in palms.

**RESEARCH ARTICLE**

## Acknowledgement

This work was supported by a grant from Department of Biotechnology (BTISnet), Government of India.

## References

Attwood TK, Beck ME, Bleasby AJ, Parry-Smith DJ. 1994. PRINTS a database of protein motif fingerprints. Nucleic Acids Res., 22: 3590 -3596.

Baldi P, Patocchi A, Zini E, Toller C, Velasco R, Komjanc M. 2004. Cloning and linkage mapping of resistance gene homologues in apple. Theor. Appl. Genet., 109: 231-239.

Bondugula R, Lee MS, Wallqvist A. 2009. FIEFDom: a transparent domain boundary recognition system using a fuzzy mean operator. Nucleic Acids Res., 37: 452–462.

Cheng J, Sweredoski M, Baldi P. 2006. DOMpro: Protein domain prediction using profiles, secondary structure, relative solvent accessibility and recursive neural networks. J. Data Min. Knowl. Discov., 13: 1–10.

Chou K, Cai Y. 2002. Using functional domain composition and support vector machines for prediction of protein sub-cellular location. J. Biol. Chem., 277: 45765-45769.

Dangl JL, Jones JD. 2001. Plant pathogens and integrated defense responses to infection. Nature, 411: 826–833.

Domingos P, Pazzani M. 1997. Beyond independence: Conditions for the optimality of the simple Bayesian classifier. Machine Learning, 29: 103-130.

Durbin R, Eddy S, Krogh A, Mitchison G. 1998. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids, Cambridge University Press, USA.

Ebina T, Toh H, Kuroda Y. 2011. DROP: an SVM domain linker predictor rained with optimal features selected by random forest. Bioinformatics, 27: 487–494.

Egan JP. 1975. Signal Detection Theory and ROC Analysis: Series in Cognition And Perception, Academic Press, New York, USA.

Hosmer DW, Lemeshow S. 2000. Applied Logistic Regression. 2nd Edn. John Wiley and Sons, New York.

Joachims T. 1999. Making large-scale SVM learning practical. In: Scholkopf, B., Burges, C. and A. Smola, editors. Advances in Kernel Methods: Support Vector Learning. Cambridge, MA, MIT Press, pp. 41–56.

Justino ER, Bortolozzi F, Sabourin R. 2005. A comparison of SVM and HMM classifiers in the off-line signature verification. Patt. Recog. Lett., 26: 1377-1385.

Koczyk G, Chelkowski J. 2003. An assessment of the resistance gene analogues of *Oryza sativa* ssp. *japonica*: their presence and structure. Cell. Mol. Biol. Lett., 8: 963-972.

Krogh A, Brown M, Mian IS, Sjölander K, Haussler D. 1994. Hidden markov models in computational biology: Applications to protein modeling. J. Mol. Biol., 235: 1501–1531.

Li B-Q, Hu L-L, Chen L, Feng K-Y, Cai Y-D, Chou K-C. 2012. Prediction of protein domain with mRMR feature selection and analysis. PLoS ONE 7(6): e39308.

Linding R, Russell RB, Neduva V, Gibson TJ. 2003. GlobPlot: Exploring protein sequences for globularity and disorder. Nucleic Acids Res., 31: 3701–3708.

Meyers BC, Kozik A, Griego A, Kuang H, Michelmore RW. 2003. Genome-wide analysis of NBS-LRR-encoding genes in *Arabidopsis*. Plant Cell, 15: 809–834.

Shao Y,. Chang C-H. 2005. Wavelet transform to hybrid support vector machine and hidden markov model for speech recognition. In: IEEE International Symposium on Circuits and Systems, International Conference Centre, Kobe, Japan, 23-26 May, 2005, pp. 3833-3836.

Spoel S, Dong X. 2012. How do plants achieve immunity? Defence without specialized immune cells. Nat. Rev. Immunol., 12: 89–100.

Suyama, M, Ohara O. 2003. DomCut: prediction of inter-domain linker regions in amino acid sequences. Bioinformatics, 19: 673–674.

Swets JA. 1988. Measuring the accuracy of diagnostic systems. Science, 240: 1285-1293

Swets JA, Dawes RM, Monahan J. 2000. Better decisions through science. Sci. Am., 283: 82–87.

Tom F. 2006. An introduction to ROC analysis. Patt.Recog. Lett., 27: 861-874.

van Ooijen G, van den Burg HA, Cornelissen BJ, Takken FL. 2007. Structure and function of resistance proteins in solanaceous plants. Annu. Rev. Phytopathol., 45:43–72.

Vapnik V. 1999. The Nature of Statistical Learning Theory. 2nd Ed. Springer-Verlag, New York. 314 p.

Vossen JH, Dezhsetan S, Esselink D, Arens M, Sanz MJ, Verweij W, Verzaux E, van der Linden CG. 2013. Novel applications of motif-directed profiling to identify disease resistance genes in plants. Plant Methods, 9: 37.

Witten I, Frank E. 2000. Data Mining. Morgan Kaufmann, Academic Press, USA. 560 p.

Zweig MH, Campbell G. 1993. Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine. Clin. Chem., 39: 561–577.