Development of machine-learning based prediction methods for inhibitors of HTRA1

MAYANK GUPTA (131581)

under supervision of

DR. JAYASHREE RAMANA

April, 2017

Submitted in partial fulfilment of the Degree of

Bachelor of Technology

in

Bioinformatics

DEPARTMENT OF BIOTECHNOLOGY AND BIOINFORMATICS

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY

WAKNAGHAT (SOLAN)

CONTENTS

Chapter No	Topics	Page No.
	Certificate from the Supervisor	II
	Acknowledgement	III
	Summary	IV-V
Chapter 1	Introduction	
Chapter2	Materials and Methods	
2.1	Tools	
2.2	Methodology	
Chapter 3	Results and Discussion	
Chapter 4	Conclusions	
	References	

CERTIFICATE

This is to certify that the work titled **Development of machine-learning based prediction methods for inhibitors of HTRA1,** submitted by **Mayank Gupta** in partial fulfillment for the award of degree of Bachelors in Technology of Jaypee University of Information Technology, Waknaghat has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor

Name of Supervisor Dr.Jayashree Ramana

Designation Assistant Professor (Senior Grade)

Date 24th April, 2017

ACKNOWLEDGEMENT

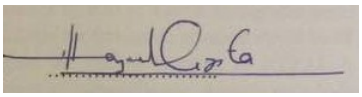
With the completion of this project for my Bachelors in Technology programme, I express earnest gratitude towards my guide Dr. Jayashree Ramana, for her guidance, support and her unscathed patience, which I have challenged more than once during the project cycle.

Her knowledge of the subject, inspires me, while her motivation along with her incessant resilience, grounds me.

It is an honest realisation, with correct guidance, we can always find our ways even when the final goal is nowhere in sight.

A hat-tip to Dr. Ragothaman Yenamalli, Dr. Tirathraj Singh for their support, guidance and motivation, both in subject and morale.

A special token of thanks to my dearest friend Deepkshitiz Sood, for picking up calls, at early mornings and late nights.



Signature of the Student

Name of Student Mayank Gupta

Date 24th April, 2017

SUMMARY

Introduction:

A wide array of activities in bioinformatics involves prediction of patterns and classification in biological data. Biological databanks, with their behemoth size, necessitate computer intervention and automation in this classification process. Currently, support vector machines (SVMs) are the computer programs with best prediction performance. SVMs optimise the margin separating two classes for better generalisation on unseen data.

HTRA1, a 50 kDa secreted protein, a member of a family of serine proteases called “High Temperature Requirement A”. The family includes other members namely: HTRA2, HTRA3, and HTRA4. All these proteins show a nonspecific protease activity while the exact role of these HTRAs is yet unknown. HTRA1 comprises a signalling peptide, a Kazal-like protease inhibitor domain, an IGF (Insulin like Growth Factor) binding domain, a PDZ domain, and a conserved serine protease domain.

The protein has shown a role in osteoarthritis, Alzheimer's disease and age-related macular degeneration, to suggest a few studies. Changes in expression of the HTRA1 gene or changes in activity of the enzyme are usually responsible for such conditions. The protein has also shown a role in chemotherapy-induced cytotoxicity in gastric, ovarian and other similar cancers.

These studies are suggestive of HTRA1 as a novel therapeutic target for multiple diseases and conditions. A specific inhibitor for this serine protease would be of paramount importance in further studies to elucidate the normal function of HTRA1 & its deregulation in the development and progression of human disease. It could potentially lead to the development of novel and effective clinical interventions.

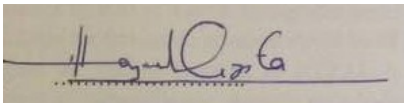
Materials and Methods:

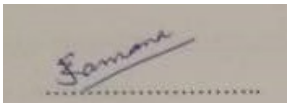
This project involves designing of an SVM Model for prediction of HTRA1 inhibitors with the help of tools – PaDEL Descriptors & WEKA.

We have also made use of Perl & Python Programming Languages, MS Excel, Notepad++ for file handling, file conversions, data manipulation, visualisation and format conversions.

Results, Discussion and Conclusions:

The models generated have prime accuracy of **77.39%** using a training set of 792 Active Compounds and 264 Inactive Compounds. The model has been tested with 10 Fold Cross Validation.





Signature of Student	Signature of Supervisor
Mayank Gupta	Dr. Jayashree Ramana
24th April, 2017	24th April, 2017

1.Introduction

1.1. HTRA1

A serine proteases family called “High Temperature Requirement A” comprises four member proteins: HTRA-1, 2, 3, and 4. While all of these proteins exhibit a nonspecific protease activity, their specific roles and functions are yet unknown [1].

HTRA1 is a 50-kDa, extracellularly secreted protein. The carboxyl terminus comprises a serine protease domain (which is highly conserved) and a PDZ protein-protein interaction motif. The amino terminal region contains a predicted signal peptide, an IGF (Insulin Growth Factor) binding domain, and a Kazal-like protease inhibitor domain.

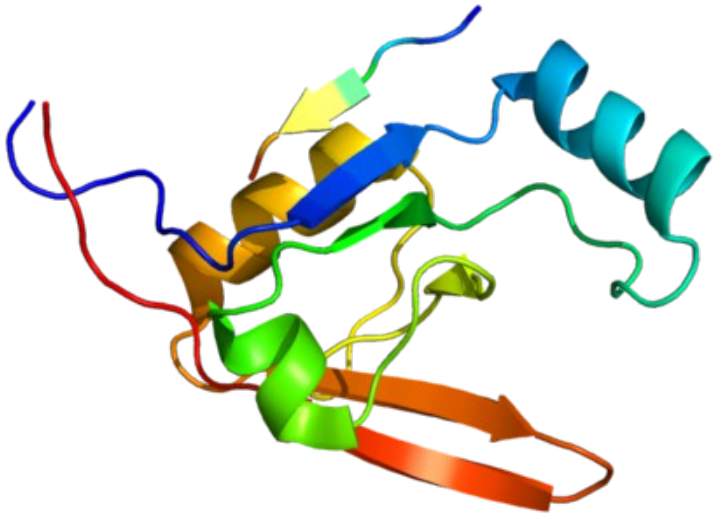


Fig1 : Structure of HTRA1, visualised in PyMol, PDB ID 2JOA

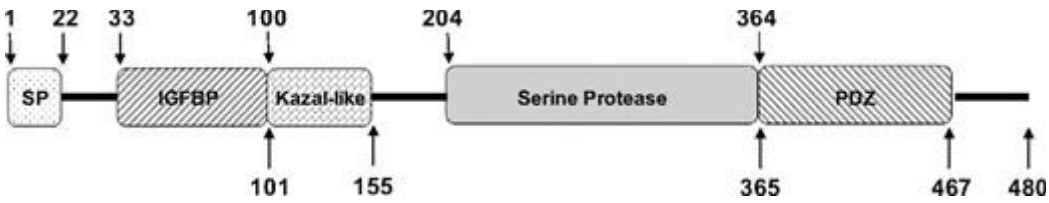


Fig2: The Structural Domains in HTRA1

The normal function of HTRA1 is unclear [2] but is expected to be involved in cleavage of extracellular matrix proteins. A number of these target substrates have been identified including C-polypeptides of fibril-forming types I, II and III procollagen, fibronectin and proteoglycans. HTRA1 has shown a role in osteoarthritis [3], Alzheimer's disease [4] and age-

related macular degeneration [5]. Overexpression of the HTRA1 gene or changes in enzyme activity is usually associated with these conditions. HTRA1 has shown a role in chemotherapy-induced cytotoxicity in mesotheliomas [6], ovarian & gastric cancers [7]. It is also shown regulation of TGF-beta signalling [8].

The gene expression when induced by an oxidative stress, promotes premature cell senescence through p38 MAPK in a protease activity-dependent manner [9]. The protein is down-regulated since early stages of bladder urothelial carcinomas development [10]. If successfully validated, it is a potential biomarker with high sensitivity and specificity for early detection of neoplasia [11]. Role of the molecule is also evident in inflammatory immune responses, which mediates control of periodontal infections as evident by immunostaining studies [12].

A frameshift mutation in the HTRA1 gene results in reduced HtrA1 protein and increased TGF-β1 expression, which may cause severe CARASIL and peripheral small arterial disease [13]. HtrA1 regulates mineralization by inhibiting TGF-β /BMP signalling and/or by cleaving specific matrix proteins, including decorin and MGP (matrix Gla protein) [14].

These studies suggest HtrA1 to be a novel therapeutic target for several diseases. A specific inhibitor for this serine protease would be invaluable in elucidation studies of normal functioning HTRA1 and its deregulation in the development and progression of human disease, potentially leading to the developments of new and effective clinical interventions.

1.2. SVM

A support vector machine (SVM) is a machine learning computer algorithm which uses previously described examples to assign labels to new objects [15].

These learning algorithms create supervised learning models to analyze datasets used in regression analysis and for classification. An SVM training algorithm builds a model assigning new datasets to either category to create a non-probabilistic, binary, linear classifier when provided with a set of training dataset, each marked as belonging to one or the other of two categories.

The SVMs showcase a data-driven method for solving classification tasks. When large numbers of features are considered for sample description, SVMs show lower prediction error compared to classifiers based on other methods like artificial neural networks.

SVMs enhance and optimise the margin separating two classes whereas other computer programs implement a classifier through the minimisation of error occurred in training. Thus, these trained models apply optimally on datasets, making SVMs ideal for protease functional site recognition, gene expression data classification, protein function prediction and transcription initiation site prediction [16].

2. Materials and Methods

2.1 Tools

2.1.1. PaDEL Descriptors

Padel is a widely used bio-chemistry tool for obtaining descriptors and fingerprints. This software uses the principles and approaches from The Chemistry Development Kit and at presently are able to calculate 797 descriptors (of which 663 are 1D & 2D while 134 are 3D) and to characterize 881 type of fingerprints.

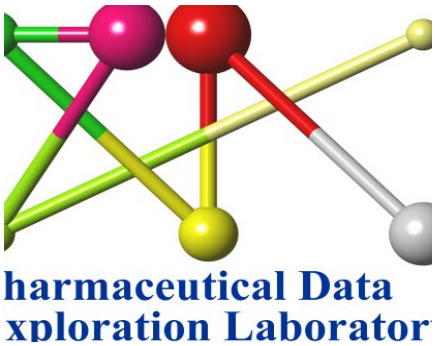


Fig3: PaDEL Descriptors Logo

Padel was constructed using Java and as a result provides you with user friendly interface and an added advantage of library component. The library component facilitates the inclusion of padel with other quantitative structure activity relationship software, so as to enable the descriptor calculation feature and to promote its usability as standalone software as well. It is a smart software that follows the Master/Worker pattern, speeding up calculations utilizing multicore CPUs.

Usage of this software offers a myriad of advantages, like- its being free and open source software makes it accessible and editable by developer's community, offers not only GUI but also command line interface, multi OS compatibility, reorganization of about 90 molecular file formats and multithreaded operability.

PaDEL is a valuable addition in currently available dynamic of descriptor calculating software. This software is available for download at- http://padel.nus.edu.sg/software/padeldescriptor.

2.1.2. Weka

Fig4 : WEKA Official Logo

Weka offers a diverse collection of machine learning algorithms for varied data mining tasks. It encompasses series of in-built features enabling the pre-processing of data, data classification, regression, clustering, association rules and even allows for visualization techniques. These algorithms can either be applied on your dataset directly or can be called using Java code. It further provides quick start for designing novel innovative machine learning algorithms.

Weka is endowed with algorithms for transformation of datasets like- discretisation and sampling algorithms, pre processing the dataset, feeding the processed data into learning models, allowing analysis of the resultant classifiers and their performances, without any necessary programming. The input required by the algorithms is usually a relational table derived directly or by executing a database query from a file.

Weka can be used in following ways-

- by applying learning method on the input dataset and further gather significant insights
- also learned models can be used for generating predictions on new occurrences
- lastly, several different learners can be applied to understand their various performance measure like- specificity, sensitivity etc.

Weka GUI, called the Explorer, gives access to all its features using menu options and form fills. The other ways include- Knowledge flow interface and the Experimenter. The Knowledge Flow interface enables dragging learning algorithms and data sources boxes to finally join them together into desired configurations, the Experimenter answers certain practical questions when using classification and regression techniques. Last and the fourth interface is Workbench is the most efficient form, offering unified interface that combines the other three into one application.

2.1.3. Accessory Tools

The tools which are used for file handling, file conversions, data manipulation, visualisation and format conversions.

2.1.3.1. Perl Programming Language

2. 1.3.2. MS Excel

2. 1.3.3. Notepad ++

2.1.3.4. Python Programming Language

2.2 Methodology

The methodology can be summarised as in Fig5, below:

Fig5: Work Flow

2.2.1. Retrieving the Data Set

Dataset was retrieved from PubChem, a renowned database for chemicals, molecules and compounds, their activities at various biological/ biochemical assays. NCBI (National Center for Biotechnology Information) a part of the National Library of Medicine is responsible for maintaining PubChem.

Dataset consists of multiple structure files in .sdf fromat, divided as Active, Inactive molecules where Actives are confirmed for a particular biochemical activity whereas Inactive are proven to be not.

```
844645
  -OEChem-11211623132D

 26 29  0     0  0  0  0  0  0999 V2000
    7.8295  -11.9617    0.0000 O   0  0  0  0  0  0  0  0  0  0  0  0
    5.2214  -10.4789    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
    0.0102   -7.5017    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
    1.3002   -3.7488    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
    1.3026   -5.2488    0.0000 N   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000   -3.0008    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000   -1.5000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0048   -6.0009    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    3.9180   -9.7351    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.3123   -8.2482    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.2990   -0.7500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.2978   -3.7529    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.2955   -5.2529    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    2.6227  -10.4916    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    3.9104   -8.2351    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.3199   -9.7482    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    2.6076   -7.4917    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.2990   -0.7500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    5.2315  -11.9789    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    6.5156   -9.7203    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    1.2990    0.7500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
   -1.2990    0.7500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    0.0000    1.5000    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    6.5355  -12.7203    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    7.8195  -10.4617    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
    2.3383   -1.3500    0.0000 C   0  0  0  0  0  0  0  0  0  0  0  0
  1 24  1  0  0  0  0
  1 25  1  0  0  0  0
  2  9  1  0  0  0  0
  2 19  1  0  0  0  0
  2 20  1  0  0  0  0
  3  8  1  0  0  0  0
  3 10  1  0  0  0  0
  4  5  2  0  0  0  0
  4  6  1  0  0  0  0
  5  8  1  0  0  0  0
  6  7  1  0  0  0  0
  6 12  2  0  0  0  0
  7 11  1  0  0  0  0
  7 18  2  0  0  0  0
  8 13  2  0  0  0  0
  9 14  2  0  0  0  0
  9 15  1  0  0  0  0
 10 16  2  0  0  0  0
 10 17  1  0  0  0  0
 11 21  2  0  0  0  0
 11 26  1  0  0  0  0
 12 13  1  0  0  0  0
 14 16  1  0  0  0  0
 15 17  2  0  0  0  0
 18 22  1  0  0  0  0
 19 24  1  0  0  0  0
 20 25  1  0  0  0  0
 21 23  1  0  0  0  0
 22 23  2  0  0  0  0
M  END
> <PUBCHEM_COMPOUND_ID_TYPE>
0

> <PUBCHEM_TOTAL_CHARGE>
0

> <PUBCHEM_SUBSTANCE_ID>
844645

> <PUBCHEM_SUBSTANCE_VERSION>
5

> <PUBCHEM_EXT_DATASOURCE_NAME>
MLSMR

> <PUBCHEM_EXT_DATASOURCE_REGID>
MLS000076246

> <PUBCHEM_SUBSTANCE_COMMENT>
http://mlsmr.evotec.com/MLSMR_HomePage/pdf/pubchem_reference.pdf
MLSMR_SAMPLE_SUPPLIER: Asinex Ltd.
MLSMR_SUPPLIER_STRUCTURE_ID: ASN 06751231
MLSMR_COMPOUND_CLASS: DC

> <PUBCHEM_SUBSTANCE_SYNONYM>
(4-Morpholin-4-yl-phenyl)-(6-o-tolyl-pyridazin-3-yl)-amine
MLS000076246
SMR000007231

> <PUBCHEM_XREF_EXT_ID>
MLS000076246

> <PUBCHEM_EXT_DATASOURCE_URL>
http://mlsmr.evotec.com/MLSMR_HomePage/

> <PUBCHEM_CID_ASSOCIATIONS>
646977  1

> <PUBCHEM_COORDINATE_TYPE>
1
3

$$$$
```

Fig6: An SDF File

The specifications for the dataset is given as following:-

PubChem AID: 540248

Fluorescence polarization- based biochemical high throughput confirmation assay for inhibitors of the HTRA serine peptidase 1 (HTRA1)

Protein Target: HTRA1 protein

Total tested substances: 1596

Active compounds: 1056

Inactive compounds: 540

Data retrieved is in form of two files : An “Active” File and another file containing “All” molecules.

To separate inactive molecules from “All” File, a python script is used.

```python
# the output file
output_file = open('file3.txt', 'w')

# contains names
names = []
try:
    # file1 contains the NAMES
    with open('file1.txt', 'r') as file1:
        for each_name in file1:
            each_name = each_name.strip()
            names.append(each_name)
    k = 0
    # file2 contains the NAMES, DATA
    with open('file2.txt', 'r') as file2:
        for each_line in file2:
            if k == len(names): break
            if each_line.strip() == names[k]:
                # add name
                print(each_line, file=output_file, end='')
                while True:
                    # add that data till $$$$ to file3.txt
                    line = file2.readline().strip()
                    print(line, file=output_file)
                    # if delimiter stop
                    if line == '%$$$$':
                        break
                k += 1

except IOError as error:
    print("File error: " + str(error))
```

Fig7: Python Script to separate inactive from all compounds.

From complete dataset, active and inactive compounds are divided into test and training data sets manually. The ratio used is:-

Training : Test :: 3 : 1

Active compounds: 792 (training set), 264(test set)

Inactive compounds: 402(training set), 134(test set)

2.2.2. Descriptor Calculation

Molecular fingerprints are an encoding methodology for structure of a molecule. Fingerprint consists of a binary digits (bits) pattern which denotes presence/ absence of particular substructures in the molecule.

Comparing fingerprints provide for applications like query matching with a given substructure, determining similarity between two molecules, etc [20].

In contrast to numerical descriptors where the quantitative plot is used for values which fall in specific value ranges, binary encoding is simple, as only the lack (0) or occurrence (1) of a specific substructure is detected.

Thus fingerprint (the fixed-length bit-string) represent the negative (0) or positive (1) occurrences of certain features solely or combinations of multiple features.

PaDEL Descriptors software is used for generating fingerprints of the 792 Active and 402 Inactive molecules.

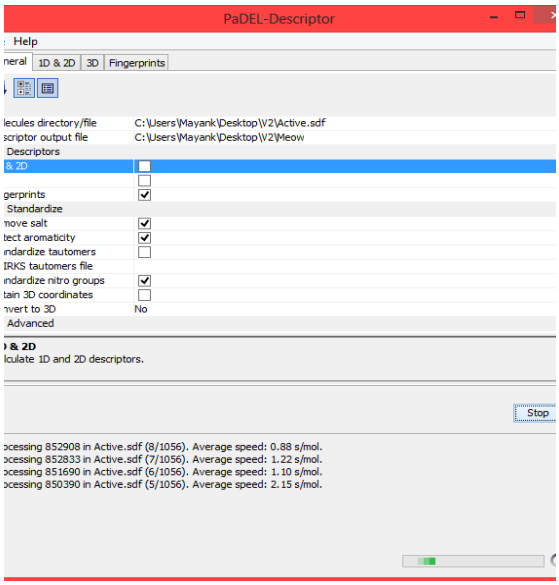


Fig7: PaDEL Descriptors calculating Fingerprints for an Active File.

```
Name,PubchemFP0,PubchemFP1,PubchemFP2,PubchemFP3,PubchemFP4,PubchemFP5,PubchemFP6,PubchemFP7,PubchemFP8,Pur
"844645",1,1,1,0,0,0,0,0,0,1,1,1,1,0,1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"846665",1,1,1,0,0,0,0,0,0,1,1,1,1,0,1,1,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"847110",1,1,0,0,0,0,0,0,0,1,1,1,0,0,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"850257",1,1,0,0,0,0,0,0,0,1,1,1,0,0,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"850390",1,1,0,0,0,0,0,0,0,1,1,1,0,0,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"851690",1,1,0,0,0,0,0,0,0,1,1,1,1,0,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"852833",1,1,1,0,0,0,0,0,0,1,1,1,1,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"852908",1,1,1,0,0,0,0,0,0,1,1,1,0,0,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"855609",1,0,0,0,0,0,0,0,0,1,1,1,0,0,1,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,
"855709",1,1,0,0,0,0,0,0,0,1,1,1,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"855829",1,1,1,0,0,0,0,0,0,1,1,1,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,
"856354",1,1,0,0,0,0,0,0,0,1,1,1,0,0,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"857079",1,1,1,0,0,0,0,0,0,1,1,1,0,0,1,1,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"857243",1,1,0,0,0,0,0,0,0,1,1,1,1,0,1,1,1,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"857297",1,1,1,0,0,0,0,0,0,1,1,1,0,0,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"858057",1,1,0,0,0,0,0,0,0,1,1,1,1,0,1,1,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"858667",1,1,0,0,0,0,0,0,0,1,1,1,1,0,1,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"863328",1,1,1,0,0,0,0,0,0,1,1,1,1,0,1,1,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
"863762",1,1,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,
"865786",1,1,1,0,0,0,0,0,0,1,1,1,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
```

Fig8: A fragment snapshot of a Fingerprints File after running through PaDEL

2.2.3. Descriptor Selection

Selection of fingerprints (descriptors) is performed using “frequency-based approach”.

Following equation (Fig 9) and (Fig10) is used for calculating frequency of each descriptor, in both inactive and active molecules:-

$$F_i^A = \frac{\sum_{j=1}^{NA} D_i^j}{NA} \times 100$$

Fig9

$$F_i^I = \frac{\sum_{j=1}^{NI} D_i^j}{NI} \times 100$$

Fig10

Where:-

F_i^A and F_i^I = Mean of the ith fingerprint in Active (A) and Inactive (I) molecules respectively.

NA and NI = total number of molecules in active and inactive datasets respectively.

D^j_i = value (0 or 1) of the ith fingerprint for the jth molecule.

Frequency Score (FS) for each fingerprint is calculated as following equation:-

$$FS_i = F_i^A - F_i^I$$

Fig11: Frequency Score Formula

Where: FS_i = inhibitory score for i^{th} descriptor.

For active molecules, descriptors should be highly positive whereas for inactive molecules, a higher negative score is preferred.

Magnitude represents the importance of the fingerprints.

Following Perl scripts are used for descriptors selection:-

```
#! C:/Perl/bin/perl
my $x,$m,$q=0,$number_compounds_active=0,$sum_des=0,$count=0;
my @molecule_des,@des_active,@count_array,@des_sum1=0,@finger_ac
open(f2,">>active_average.txt");
for(my $w=1;$w<=881;$w++)
{
$sum_des=0;
open(f1,"active.txt");
while($x=<f1>)
{
@molecule_des=split('\t',$x);
#print $molecule_des[4]."\t";
#print $molecule_des[1]."\t";
$sum_des=$sum_des+$molecule_des[$w];
$number_compounds_active=$number_compounds_active+1;
}
#print "\n".$number_compounds_active;
$finger_p_active=($sum_des/$number_compounds_active)*100;
$finger_active[$q]=$finger_p_active;
#print $finger_active[$q];
$q=$q+1;
$number_compounds_active=0;
close(f1);
}

foreach $a(@finger_active)
{
print f2 $a."\t";
}
#print "".$finger_p_active."\t";
```

Fig12: Perl Script for Frequency Score Calculation

```perl
#! C:/Perl/bin/perl
my $x,$y,$m;
my @active,@inactive,@freq_score;
open(f1,"active_average.txt");
open(f2,"inactive_average.txt");
while($x=<f1>)
{
@active=split('\t',$x);
}
close(f1);
while($y=<f2>)
{
@inactive=split('\t',$y);
}
for($i=0;$i<881;$i++)
{
$freq_score[$i]=$active[$i]-$inactive[$i
}

open(f3,">>freq_score.txt");

foreach $m(@freq_score)
{
print f3 $m."\t";
}
exit;
```

Fig13: Perl Script for Frequency Score Calculation

```
#! C:/Perl/bin/perl
my $x;
my @freq_score,@after_compare,@c
open(f1,"freq_score.txt");
while($x=<f1>)
{
@freq_score=split("\t",$x);
}
my $j=0;
for($i=0;$i<881;$i++)
{
if($freq_score[$i]>=0.6)
{
$after_compare[$i]=$freq_score[$
$counter[$j]=$i;
$j=$j+1;
}
}
open(f2,">>selected_descriptors.
foreach $x(@counter)
{
print f2 $x."\t";
}
exit;
```

Fig14: Descriptor Selection Program with threshold selected 0.6

```
1   #! C:/Perl/bin/perl
2   my $x,$y;
3   my @index_array,@each_line;
4   open(f1,"selected_descriptors.txt");
5   open(f3,">>active_selected_descriptors.
6   while($x=<f1>)
7   {
8   @index_array=split('\t',$x);
9   }
10  foreach $x(@index_array)
11  {
12  open(f2,"activedes.txt");
13  while($y=<f2>)
14  {
15  @each_line=split('\t',$y);
16  if($each_line[0]==$x)
17  {
18  print f3 $y;
19  }
20  }
21  close(f2);
22  }
```

Fig15: Perl Script for generating selected descriptors file

185	768	PubchemFP768	0	0	0	0	0	0
186	776	PubchemFP776	1	0	0	0	0	0
187	777	PubchemFP777	0	0	0	1	0	0
188	780	PubchemFP780	0	0	0	0	0	0
189	785	PubchemFP785	0	0	0	0	0	0
190	786	PubchemFP786	0	0	0	0	0	0
191	790	PubchemFP790	0	0	0	0	0	0
192	791	PubchemFP791	0	0	1	0	1	0
193	797	PubchemFP797	0	0	0	0	0	0
194	798	PubchemFP798	0	0	0	0	0	1
195	799	PubchemFP799	0	0	0	0	0	0
196	800	PubchemFP800	0	0	0	0	1	0
197	801	PubchemFP801	0	0	0	0	0	0
198	802	PubchemFP802	0	0	0	0	0	0
199	813	PubchemFP813	0	0	0	0	0	0
200	818	PubchemFP818	0	0	0	0	0	0
201	819	PubchemFP819	0	0	0	0	0	0
202	820	PubchemFP820	0	0	0	0	0	0
203	821	PubchemFP821	0	0	0	1	1	1
204	822	PubchemFP822	0	0	0	0	0	0
205	825	PubchemFP825	0	0	0	0	0	0
206	826	PubchemFP826	0	0	0	0	0	1
207	831	PubchemFP831	0	0	0	0	0	0
208	833	PubchemFP833	0	0	0	0	0	0
209	839	PubchemFP839	0	0	0	0	0	0
210								

Fig16: Selected Fingerprints File

The selected number of descriptors (fingerprints) is 242.

2.2.4. Model Generation

An ARFF file is generated to be fed to WEKA for Model Generation.

An ARFF (Attribute-Relation File Format) file is ASCII texts file that describing a series of instances, sharing a specific set of properties or attributes. The “Machine Learning Project” at the Department of Computer Science, University of Waikato developed this format for use in the Weka software.

These files have two distinctive parts:-

(i.) **Header** information,

(ii.) **Data** information.

The **Header** contains relation names, attributes list (columns in the data), their types.

Lines that begin with a % are comments. The **@RELATION**, **@ATTRIBUTE** and **@DATA** declarations are case insensitive.

```
1  @relation SRC
2  @attribute '1' {0,1}
3  @attribute '2' {0,1}
4  @attribute '3' {0,1}
5  @attribute '4' {0,1}
6  @attribute '5' {0,1}
7  @attribute '6' {0,1}
8  @attribute '209' {0,1}
9  @attribute 'class' {'Positive','Negative'}
10 @data
11 0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,Positive'
12 0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,Positive'
```

Fig17: A Sample ARFF File

The **Sequential Minimal Optimization** (SMO) classifier is chosen for model building in Weka.

A support vector machine requires the solution of a very large QP (quadratic programming) optimization problem for training. The SMO classifier breaks these problems into a list of smallest possible QP problems, which are solved logically, avoiding a time-consuming inner loop of numerical QP optimizations.

A linear amount of memory is required for SMO training, which enables it to handle very lengthy training datasets. Computation time for SMO is primarily that of SVM evaluation; making SMO fastest for linear SVMs and small data sets. On real- world sparse data sets, SMO can be more than 1000 times faster than the chunking algorithm.[23]

The (Gaussian) **RBF Kernel** is selected.

The RBF (radial basis function) kernel is an accepted kernel function used in various partitioning learning algorithms like support vector machine classification.

On 2 samples, x and x', RBF kernel is represented as feature vectors in some *input space* and can be defined as

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{\quad}\right)$$

Where $\|x-x'\|^2$ = squared Euclidean distance between 2 feature vectors.

s[5]

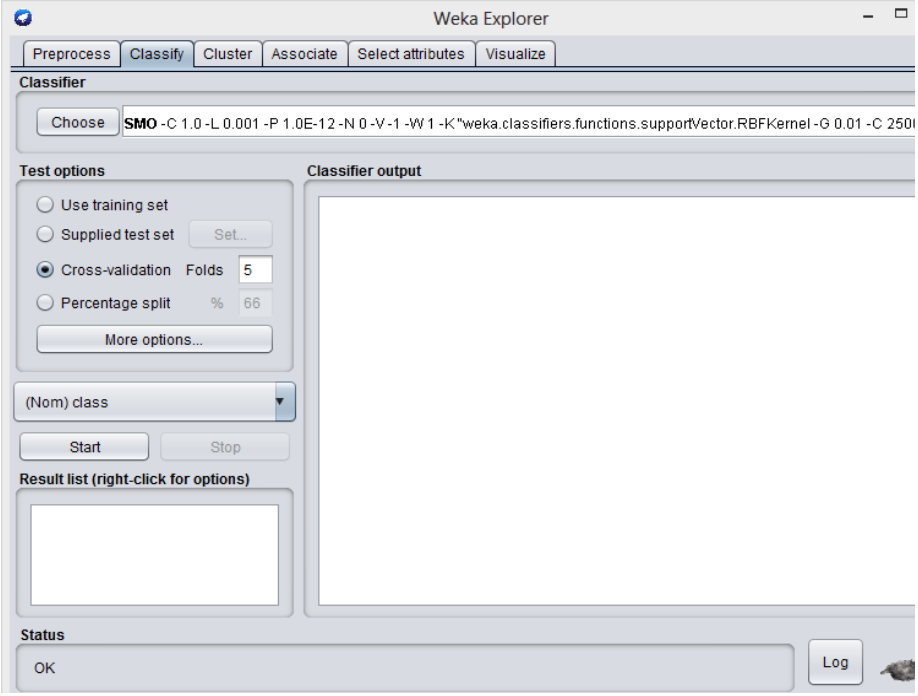
= a free parameter.

A similar definition involves a function: :

Value of RBF kernel (K) falls between zero and one; value decreases with distance. It is also referred as a similarity measure.[24]

The gamma parameter measures the influence of a singular training data point, where low values meaning 'far' and high values meaning 'close'. It is defined as inversed radius of influence of model selected samples.

Misclassification of training dataset is replaced against simplicity of the decision surface by C parameter. Decision surface is smooth in low C, whereas a high value allows model freedom to select more samples as support vectors, classifying all training samples correctly [25].


Weka Explorer
Preprocess
Classify
Cluster
Associate
Select attributes
Visualize
Classifier
Choose
SMO -C 1.0 -L 0.001 -P 1.0E-12 -N 0 -V -1 -W 1 -K "weka.classifiers.functions.supportVector.RBFKernel -G 0.01 -C 250
Test options
Use training set
Supplied test set
Set...
Cross-validation
Folds
5
Percentage split
%
66
More options...
Classifier output
(Nom) class
Start
Stop
Result list (right-click for options)
Status
OK
Log

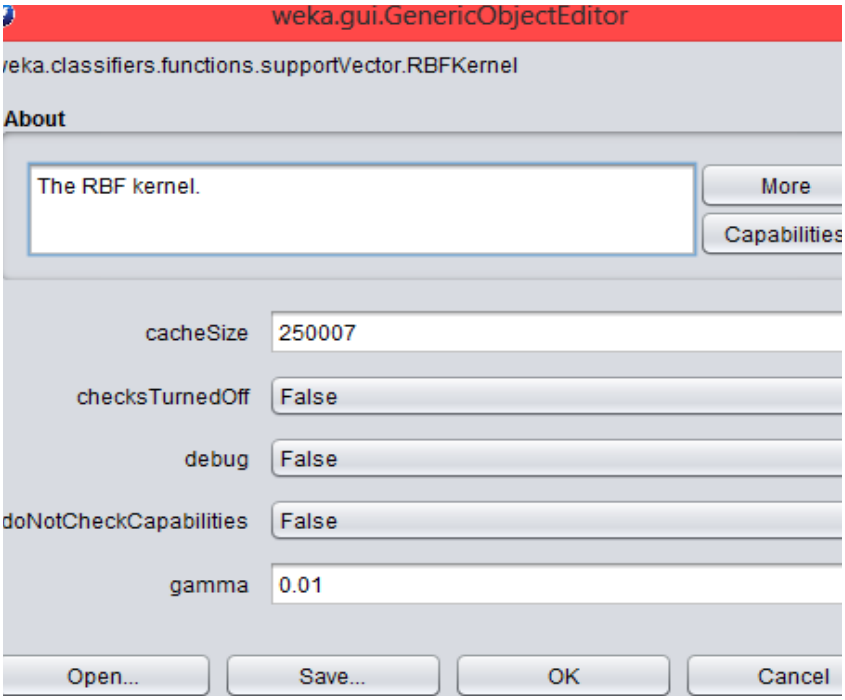


Fig18, 19,20: Configuration of WEKA

2.2.5. Model Testing

The model performance is evaluated with ten-fold cross-validation techniques, where training and testing were carried out ten times.

In each iteration, n-1 sets are used for training while a single set is used for testing. The training set is randomly divided into ten training and testing sets. To avoid any bias in the prediction model, an independent validation set is also used. Complete process is repeated ten times, and the results are reported after obtaining the average [26].

2.2.6. Model Optimisation

Model fitness is assessed using multiple standard parameters like true positive rate (TP), false positive rate (FP), precision, recall, F-measure, Matthew's Correlation Coefficient (MCC), ROC area, PRC area, Accuracy.[27]

The value of Gamma and C-Value, when modified, gives varying accuracy. The classifier can be optimized for values of parameters with best accuracy along with high MCC.

3. Results & Discussions

The results are derived in form of a Weka Classifier Specifications, which on application to a dataset will be able to distinguish between an inhibitor and non-inhibitor against HTRA1.

Following results are obtained with varying values of C, Gamma :-

Serial	C	Gamma	Accuracy	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
1	1	0.01	75.2094	0.918	0.575	0.759	0.918	0.831	0.407	0.672	0.751	Positive
				0.425	0.082	0.725	0.425	0.536	0.407	0.672	0.502	Negative
				0.752	0.409	0.747	0.752	0.732	0.407	0.672	0.667	
2	1	0.1	74.9581	0.9	0.547	0.764	0.9	0.827	0.404	0.676	0.754	Positive
				0.453	0.1	0.697	0.453	0.549	0.404	0.676	0.5	Negative
				0.75	0.397	0.742	0.75	0.733	0.404	0.676	0.669	
3	1	1	67.7554	0.991	0.94	0.675	0.991	0.803	0.151	0.525	0.675	Positive
				0.06	0.009	0.774	0.06	0.111	0.151	0.525	0.363	Negative
				0.678	0.627	0.708	0.678	0.57	0.151	0.525	0.57	
4	0.1	0.01	66.3317	1	1	0.663	1	0.798	0	0.5	0.663	Positive
				0	0	0	0	0	0	0.5	0.337	Negative
				0.663	0.663	0.44	0.663	0.529	0	0.5	0.553	
5	1	0.001	66.3317	1	1	0.663	1	0.798	0	0.5	0.663	Positive
				0	0	0	0	0	0	0.5	0.337	Negative
				0.663	0.663	0.44	0.663	0.529	0	0.5	0.553	
6	10	0.01	77.3869	0.861	0.398	0.81	0.861	0.835	0.48	0.732	0.79	Positive
				0.602	0.139	0.688	0.602	0.642	0.48	0.732	0.548	Negative
				0.774	0.311	0.769	0.774	0.77	0.48	0.732	0.708	
7	1	10	67.5042	0.994	0.953	0.673	0.994	0.802	0.138	0.52	0.673	Positive
				0.047	0.006	0.792	0.047	0.089	0.138	0.52	0.358	Negative
				0.675	0.634	0.713	0.675	0.562	0.138	0.52	0.567	
8	1	100	67.5042	0.994	0.953	0.673	0.994	0.802	0.138	0.52	0.673	Positive
				0.047	0.006	0.792	0.047	0.089	0.138	0.52	0.358	Negative
				0.675	0.634	0.713	0.675	0.562	0.138	0.52	0.567	
9	0.1	0.001	66.3317	1	1	0.663	1	0.798	0	0.5	0.663	Positive
				0	0	0	0	0	0	0.5	0.337	Negative
				0.663	0.663	0.44	0.663	0.529	0	0.5	0.553	
10	0.1	0.1	66.3317	1	1	0.663	1	0.798	0	0.5	0.663	Positive
				0	0	0	0	0	0	0.5	0.337	Negative
				0.663	0.663	0.44	0.663	0.529	0	0.5	0.553	
11	0.1	1	66.3317	1	1	0.663	1	0.798	0	0.5	0.663	Positive
				0	0	0	0	0	0	0.5	0.337	Negative
				0.663	0.663	0.44	0.663	0.529	0	0.5	0.553	
12	0.1	10	66.3317	1	1	0.663	1	0.798	0	0.5	0.663	Positive
				0	0	0	0	0	0	0.5	0.337	Negative
				0.663	0.663	0.44	0.663	0.529	0	0.5	0.553	

13	10	1	69.263	0.984	0.881	0.688	0.984	0.809	0.221	0.551	0.687	Positive
				0.119	0.016	0.787	0.119	0.207	0.221	0.551	0.39	Negative
				0.693	0.59	0.721	0.693	0.607	0.221	0.551	0.587	
14	10	0.1	73.6181	0.857	0.502	0.771	0.857	0.812	0.381	0.677	0.755	Positive
				0.498	0.143	0.639	0.498	0.559	0.381	0.677	0.487	Negative
				0.736	0.381	0.726	0.736	0.727	0.381	0.677	0.665	
15	10	0.001	75.1256	0.904	0.55	0.764	0.904	0.828	0.407	0.677	0.754	Positive
				0.45	0.096	0.704	0.45	0.549	0.407	0.677	0.502	Negative
				0.751	0.397	0.744	0.751	0.734	0.407	0.677	0.67	
16	10	0.0001	66.3317	1	1	0.663	1	0.798	0	0.5	0.663	Positive
				0	0	0	0	0	0	0.5	0.337	Negative
				0.663	0.663	0.44	0.663	0.529	0	0.5	0.553	
17	100	0.0001	75.1256	0.905	0.552	0.764	0.905	0.828	0.407	0.677	0.754	Positive
				0.448	0.095	0.706	0.448	0.548	0.407	0.677	0.502	Negative
				0.751	0.398	0.744	0.751	0.734	0.407	0.677	0.669	
18	100	0.001	76.2982	0.864	0.435	0.796	0.864	0.829	0.451	0.714	0.778	Positive
				0.565	0.136	0.678	0.565	0.616	0.451	0.714	0.529	Negative
				0.763	0.335	0.756	0.763	0.757	0.451	0.714	0.694	

19	100	0.01	73.1156	0.797	0.398	0.798	0.797	0.797	0.398	0.699	0.77	Positive
				0.602	0.203	0.6	0.602	0.601	0.398	0.699	0.495	Negative
				0.731	0.332	0.731	0.731	0.731	0.398	0.699	0.678	
20	100	0.1	73.7856	0.857	0.498	0.772	0.857	0.813	0.386	0.68	0.757	Positive
				0.502	0.143	0.641	0.502	0.563	0.386	0.68	0.49	Negative
				0.738	0.378	0.728	0.738	0.729	0.386	0.68	0.667	
21	100	1	69.263	0.984	0.881	0.688	0.984	0.809	0.221	0.551	0.687	Positive
				0.119	0.016	0.787	0.119	0.207	0.221	0.551	0.39	Negative
				0.693	0.59	0.721	0.693	0.607	0.221	0.551	0.587	
22	1000	1	69.263	0.984	0.881	0.688	0.984	0.809	0.221	0.551	0.687	Positive
				0.119	0.016	0.787	0.119	0.207	0.221	0.551	0.39	Negative
				0.693	0.59	0.721	0.693	0.607	0.221	0.551	0.587	
23	1000	0.01	72.3618	0.802	0.43	0.786	0.802	0.794	0.375	0.686	0.762	Positive
				0.57	0.198	0.593	0.57	0.581	0.375	0.686	0.483	Negative
				0.724	0.352	0.721	0.724	0.722	0.375	0.686	0.668	
24	1000	0.001	75.8794	0.835	0.391	0.808	0.835	0.821	0.452	0.722	0.784	Positive
				0.609	0.165	0.652	0.609	0.63	0.452	0.722	0.529	Negative
				0.759	0.315	0.755	0.759	0.757	0.452	0.722	0.698	

25	1000	0.0001	75.8794	0.87	0.46	0.788	0.87	0.827	0.437	0.705	0.772	Positive
				0.54	0.13	0.678	0.54	0.601	0.437	0.705	0.521	Negative
				0.759	0.349	0.751	0.759	0.751	0.437	0.705	0.688	
26	10000	0.0001	76.4657	0.854	0.41	0.804	0.854	0.828	0.459	0.722	0.783	Positive
				0.59	0.146	0.671	0.59	0.628	0.459	0.722	0.534	Negative
				0.765	0.322	0.759	0.765	0.761	0.459	0.722	0.699	
27	10000	0.001	73.5343	0.798	0.388	0.802	0.798	0.8	0.409	0.705	0.774	Positive
				0.612	0.202	0.606	0.612	0.609	0.409	0.705	0.501	Negative
				0.735	0.325	0.736	0.735	0.736	0.409	0.705	0.682	
28	10000	0.01	72.3618	0.802	0.43	0.786	0.802	0.794	0.375	0.686	0.762	Positive
				0.57	0.198	0.593	0.57	0.581	0.375	0.686	0.483	Negative
				0.724	0.352	0.721	0.724	0.722	0.375	0.686	0.668	
29	10000	1	69.263	0.984	0.881	0.688	0.984	0.809	0.221	0.551	0.687	Positive
				0.119	0.016	0.787	0.119	0.207	0.221	0.551	0.39	Negative
				0.693	0.59	0.721	0.693	0.607	0.221	0.551	0.587	
30	10000	0.00001	75.8794	0.869	0.458	0.789	0.869	0.827	0.438	0.705	0.772	Positive
				0.542	0.131	0.677	0.542	0.602	0.438	0.705	0.521	Negative
				0.759	0.348	0.751	0.759	0.751	0.438	0.705	0.688	

Fig21: Table of Results (Values of Accuracy and other params for various C, Gamma combinations)

The best classifier currently obtained shows an accuracy of **77.39%**

Its C Value is **10.0** and Gamma Parameter is **0.01**

The value of MCC is **0.480**

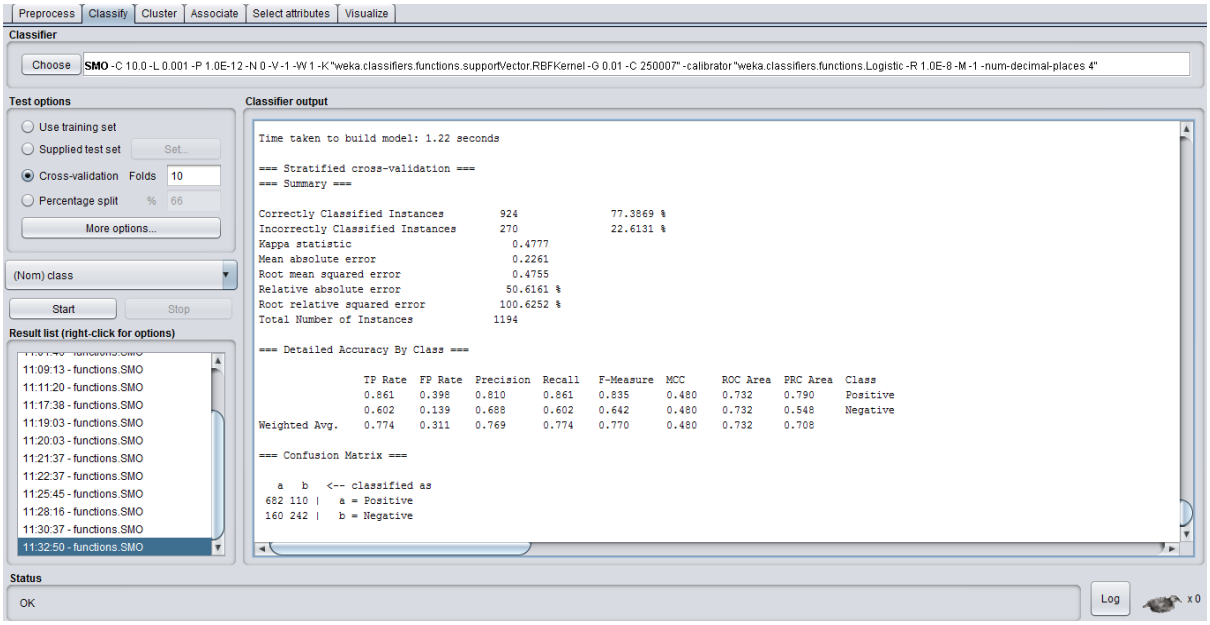


Fig21: WEKA Results

3.2. Discussion

With iteratively varying values of C and Gamma Parameter, it is understood that values of accuracy change with every combinations.

The values of accuracy solely are not reliable measure of quality of a model as it fails to consider all parameters. Thus we also need to consider other factors like TP Rate, Recall, MCC, etc

In model 9, 10, 11, 12, Value of accuracy remains 66.33% for four values of gamma when c is 0.1

In model 21, 22, Value of accuracy remains 69.26% for two values of c when gamma is 1.

In model 19, value of accuracy is 73.12% whereas MCC is just 0.398

Hence, for selection of a model, we need to look at multiple parameters.

Thus, selecting Model 6, with C value 10, gamma 0.01:

It gives accuracy 77.39% with MCC 0.480

4. Conclusions

In this study, we have developed multiple SVM models using SMO classifiers and RBF kernels, in multiple cycles.

The first few cycles involved using datasets of varying sizes for understanding the protocol.

Later cycles involved larger datasets with varying values of C and Gamma parameters.

The learning involved in former cycles was, the dataset needs to be sufficiently sized to avoid biased model training, followed by skewed results. The results obtained were of high accuracy but the model developed was not of any significance as it failed to classify the new datasets accurately.

Later cycles, brought learning that a model with high accuracy doesn't necessarily means it is of paramount significance as we also need to consider other factors like TP Rate, Precision, Recall and MCC.

The best model obtained with C value 10 & gamma 0.01, gave an accuracy 77.39% with MCC 0.480

The model obtained is still not very reliable for classifying novel datasets accurately. If we need a very high dependence then we need to improve it further. This model can be cross validated and enhanced using other methods like ANN, etc.

5. References

1. Clausen, T., Southan, C., & Ehrmann, M. (2002). The HtrA family of proteases: implications for protein composition and cell fate. *Molecular cell*, *10*(3), 443-455.
2. Ehrmann, M., & Clausen, T. (2004). Proteolysis as a regulatory mechanism. *Annu. Rev. Genet.*, *38*, 709-724.
3. Hu, S. I., Carozza, M., Klein, M., Nantermet, P., Luk, D., & Crowl, R. M. (1998). Human HtrA, an evolutionarily conserved serine protease identified as a differentially expressed gene product in osteoarthritic cartilage. *Journal of Biological Chemistry*, *273*(51), 34406-34412.
4. Grau, S., Richards, P. J., Kerr, B., Hughes, C., Caterson, B., Williams, A. S., ... & Ehrmann, M. (2006). The role of human HtrA1 in arthritic disease. *Journal of Biological Chemistry*, *281*(10), 6124-6129.
5. Li, Y., Polur, I., Lee, P. L., Servais, J. M., & Xu, L. (2009). 104 A POSSIBLE ROLE OF HTRA1, A SERINE PROTEASE, IN PATHOGENESIS OF OSTEOARTHRITIS. *Osteoarthritis and Cartilage*, *17*, S63.
6. Tsuchiya, A., Yano, M., Tocharus, J., Kojima, H., Fukumoto, M., Kawaichi, M., & Oka, C. (2005). Expression of mouse HtrA1 serine protease in normal bone and cartilage and its upregulation in joint cartilage damaged by experimental arthritis. *Bone*, *37*(3), 323-336.
7. Grau, S., Baldi, A., Bussani, R., Tian, X., Stefanescu, R., Przybylski, M., ... & Ehrmann, M. (2005). Implications of the serine protease HtrA1 in amyloid precursor protein processing. *Proceedings of the National Academy of Sciences*, *102*(17), 6021-6026.
8. Marx, J. (2006). Gene offers insight into macular degeneration. *Science*, *314*(5798), 405-405.

9. Shimomachi, M., Hasan, M. Z., Kawaichi, M., & Oka, C. (2013). HtrA1 is induced by oxidative stress and enhances cell senescence through p38 MAPK pathway. *Experimental eye research*, *112*, 79-92.

10. Lorenzi, T., Lorenzi, M., Altobelli, E., Marzioni, D., Mensà, E., Quaranta, A., ... & Procopio, A. D. (2013). HtrA1 in human urothelial bladder cancer: a secreted protein and a potential novel biomarker. *International journal of cancer*, *133*(11), 2650-2661.

11. Xu, Y., Jiang, Z., Zhang, Z., Sun, N., Zhang, M., Xie, J., ... & Wu, D. (2014). HtrA1 Downregulation Induces Cisplatin Resistance in Lung Adenocarcinoma by Promoting Cancer Stem Cell-Like Properties. *Journal of cellular biochemistry*, *115*(6), 1112-1121.

12. Lorenzi, T., Niţulescu, E. A., Zizzi, A., Lorenzi, M., Paolinelli, F., Aspriello, S. D., ... & Lombardi, T. (2014). The novel role of HtrA1 in gingivitis, chronic and aggressive periodontitis. *PloS one*, *9*(6), e96978.

13. Cai, B., Zeng, J., Lin, Y., Lin, Y., Lin, W., Lin, W., ... & Wang, N. (2015). A frameshift mutation in HTRA1 expands CARASIL syndrome and peripheral small arterial disease to the Chinese population. *Neurological Sciences*, *36*(8), 1387-1391.

14. Canfield, A. E., Hadfield, K. D., Rock, C. F., Wylie, E. C., & Wilkinson, F. L. (2007). HtrA1: a novel regulator of physiological and pathological matrix mineralization?.

15. Yang, Z. R. (2004). Biological applications of support vector machines. *Briefings in bioinformatics*, *5*(4), 328-338.

16. Byvatov, E., & Schneider, G. (2002). Support vector machine applications in bioinformatics. *Applied bioinformatics*, *2*(2), 67-77.

17. Yap, C. W. (2011). PaDEL-descriptor: An open source software to calculate molecular descriptors and fingerprints. *Journal of computational chemistry*, *32*(7), 1466-1474.

18. Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

19. Xue, L., Godden, J. W., & Bajorath, J. (1999). Database searching for compounds with similar biological activity using short binary bit string representations of molecules. *Journal of chemical information and computer sciences*, *39*(5), 881-886.

20. McGregor, M. J., & Pallai, P. V. (1997). Clustering of large databases of compounds: Using the MDL "keys" as structural descriptors. *Journal of chemical information and computer sciences*, *37*(3), 443-448.

21. Pickett, S. D., Luttmann, C., Guerin, V., Laoui, A., & James, E. (1998). DIVSEL and COMPLIB-Strategies for the design and comparison of combinatorial libraries using pharmacophoric descriptors. *Journal of chemical information and computer sciences*, *38*(2), 144-150.

22. Sheridan, R. P., Miller, M. D., Underwood, D. J., & Kearsley, S. K. (1996). Chemical similarity using geometric atom pair descriptors. *Journal of Chemical Information and Computer Sciences*, *36*(1), 128-136.

23. Platt, J. (1998). Sequential minimal optimization: A fast algorithm for training support vector machines.

24. Chang, Y. W., Hsieh, C. J., Chang, K. W., Ringgaard, M., & Lin, C. J. (2010). Training and testing low-degree polynomial data mappings via linear SVM. *Journal of Machine Learning Research*, *11*(Apr), 1471-1490.

25. Schölkopf, B., Tsuda, K., & Vert, J. P. (2004). *Kernel methods in computational biology*. MIT press.

26. Shashua, A. (2009). Introduction to machine learning: Class notes 67577. *arXiv preprint arXiv:0904.3664*.

27. Bekkar, M., Djemaa, H. K., & Alitouche, T. A. (2013). Evaluation measures for models assessment over imbalanced data sets. *Iournal Of Information Engineering and Applications*, *3*(10).