Development of Machine Learning Based Prediction Method for Laccases

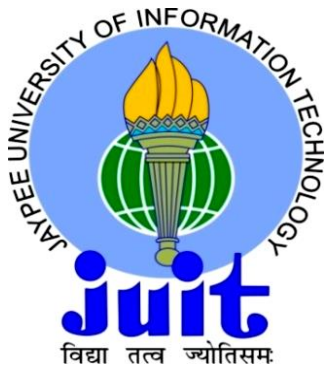
Project report submitted in partial fulfillment of the requirements for
the Degree of Bachelor of technology
In

Department of Biotechnology and Bioinformatics

By
Mritunjay Singh Chandel [151511]
Sanjeevani Ravi Srivastava [151505](**up to section I**)

Under the supervision of
Dr Jayashree Ramana
Assistant Professor (Senior Grade)

To



Department of Biotechnology and Bioinformatics

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT,Solan-173234,Himachal Pradesh

TABLE OF CONTENT

CANDIDATE'S DECLARATION

I hereby declare that the work presented in the report entitled "**Development of Machine Learning Based Prediction Methods for Laccases"** in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Biotechnology and Bioinformatics** submitted in the department of Biotechnology and Bioinformatics ,Jaypee University Of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from August 2018 to December 2018 under the supervision of **Dr Jayashree Ramana,** Assistant Professor(Senior Grade),Department of Biotechnology and Bioinformatics .

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Mritunjay Singh Chandel
151511

Sanjeevani Ravi Srivastava(contribution up to section I)
151505

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr Jayashree Ramana
Assistant Professor(Senior Grade)
Biotechnology and Bioinformatics
Dated:

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are thankful to Prof Dr Sudhir Kumar, Head of department of Biotechnology and Bioinformatics for the support.

We are thankful to our project coordinator Dr Uday Bhanu for his support and guidance.

We are highly thankful to Dr Jayashree Ramana for her guidance and constant supervision as well as for providing necessary information regarding the project and also for her support in the project.

We would like to express our gratitude towards our university members for their kind cooperation and encouragement which helped us in this project.

Our thanks and appreciations also go to our colleagues in developing the project and people who have willingly helped us out with their abilities.

SECTION I

ABSTRACT

Laccases are multicopper containing enzymes found in plants, fungi and microorganisms which oxidize different aromatic and non-aromatic compounds with the help of radical-catalyzed reaction process. These are useful biocatalysts for various applications in biotechnology. These are very diverse and all its activities rise from a single reaction. These are distributed in each and every domain of life so more knowledge is required for their identification. These have applications in paints, cosmetics, bioremediation, textiles, food, pulp, paper industry and useful in production of bioethanol from lignocellulose materials as feedstock.

There is a need to understand their physiological importance for further use in various biotechnological applications. Information collected from all the relevant sources regarding Lacasse and Non Laccase enzymes in order to make decisions are analyzed with the help of domain based method as well as similarity based method after redundancy removal. The results obtained from above mentioned bioinformatic approaches are not so accurate and efficient in identification of Laccase enzyme.

Therefore, there is a need to develop a machine learning model that allows identification of Laccases to be more accurate for generation of greater biotechnological applications.

Introduction

1.1 Laccase

Laccases are oxidase enzymes which contain copper and present in diverse plants, microorganisms and fungi. These enzymes carry on one electron oxidation resulting in crosslinking. Laccases function on phenols and various similar substrates. These play a role in emergence of lignin by increasing the oxidation of phenols. Laccases found in fungus *Pleurotus ostreatus*, play a role in the degradation of lignin and therefore classified as lignin modifying enzymes. They have the capacity to oxidise phenol as well as non-phenolic lignin related compounds which has gained attention of researchers. Laccases participate in cleavage of aromatic compounds.

Laccase was first found by Gabriel Bertrand in 1894 in Japanese lacquer tree (in its sap). It helps in the formation of lacquer, hence it is named laccase. These are largely found in higher plants and fungi. In the past few years, they have obtained application in the area of textile, food, pulp and paper industry. It has two main types laccase-1 and laccase-2. These are the oldest and widely studied enzyme by researchers. It consists of 15-30% of carbohydrate and a molecule mass of 60-90 kDa. These enzymes contain 4 copper ions per molecule that are responsible for oxidation of phenol and oxygen. Laccases are found in plants such as cabbages, potato, apples and other vegetables.

1.2 Sources of Laccases

Laccase are prevalent in higher plants and fungi. These are found in some bacterial species such as *S.lavendulae, S.cyaneus* and *Marinomonas mediterranea*. These occur in Basidiomycetes, Ascomycetes, white-rot fungi and many *Trichoderma* species.

1.3 Mechanism of Laccases

Laccases mixes the substrate oxidation with the electron reduction of oxygen to water. It includes three copper centers, which are classified as type 1, type 2, type 3. These sites are differentiated on the basis of spectroscopic, kinetic and computational features. The center gets attached with oxygen and reduces it to water. They act only on phenol.

Reaction including oxygen focuses on the product formed during two step electron reduction process where in, the first step peroxide intermediate is formed and in the second step native intermediate is formed which has a catalytic importance in oxidized mode of the enzyme.

1.4 Properties of Laccase

Laccases are glycoprotein of monomeric, dimeric and tetrameric type. The process of glycosylation plays a major role in thermal stability, degradation, secretion and copper retention. They exhibit significant heterogeneity. They make the bioremediation process more efficient using the industrial waste obtained from fungi, bacteria, higher plants and lichen. These belong to multicopper oxidase family and are responsible for monomers crosslinking process.

1.5 Effect of Carbon and Nitrogen origin

Organism growth media contains different amount of carbon and nitrogen content. Large amount of glucose and sucrose decreases the production of enzyme by blocking the initial step of laccase formation. This blockage can be improved with the help of cellulose like polymeric substrates. The production of Laccase is increased in the absence of nitrogen.

1.6 Influence of Temperature

The effect of temperature is restricted in the production of Laccase enzyme. 25°C is the perfect temperature for production of Laccase in the presence of light source and 30°C in dark.

1.7 Effect of pH

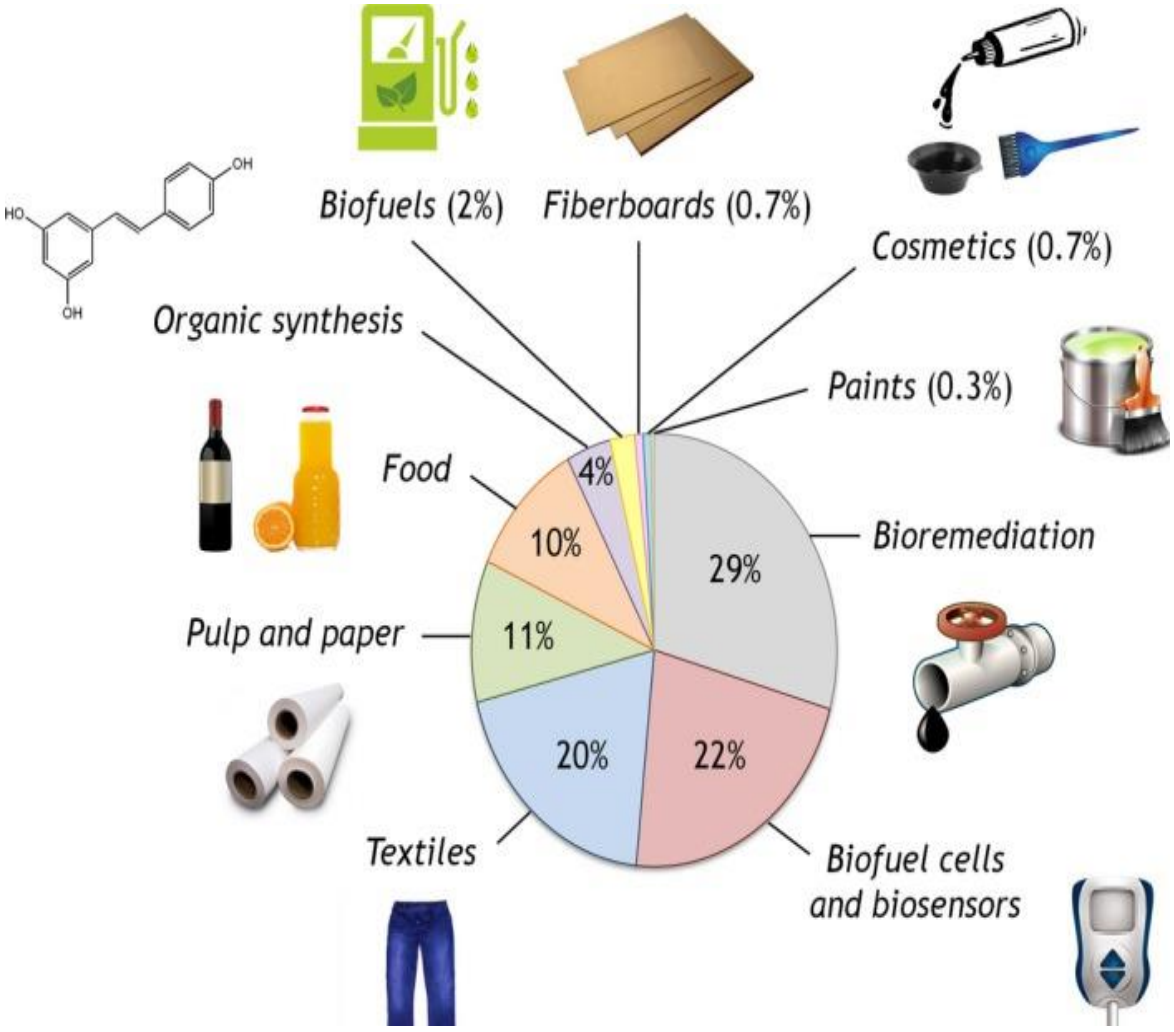
The effect of pH is restricted in the production of Laccase. Optimum value varies depending upon various reaction between substrates for laccases. At high pH concentration substrate oxidation increases. Most researches demonstrate the optimum pH between 4.5 and 6.0 for the production of Laccases.

1.8 Effect of Agitator

It is a factor which affects the production of laccases. Due to growth of fungus mycelia gets damaged as a result the laccase production decreases. Agitation doesn’t affect the production of laccases according to the research.

2. Application of Laccases

Laccases are important because of their ability of oxidizing toxic and non -toxic substrates. They have an application in textile industry, food processing industry, pharmaceutical and chemical industry. These are specific excellent catalyst and ecologically sustainable. Some of the applications are mentioned below.



(Fig.1)https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5658592/

2.1 Dye Decolorization

Textile industry in wet processing uses big amount of water and chemicals which include organic as well as inorganic compounds. Laccase are used for the production of synthetic dyes because they degrade the dye, dye when exposed to light or water looses its color. They are used in the market in the form of hair dyes because they cause less irritation (replace H_2O_2) and are user friendly.

2.2 Bioremediation and Biodegradation

Contamination of water and soil are the major results of rapid industrialization and excessive use of pesticides. It is a serious environmental concern now-a-day. Chemicals used in the production of pesticides are carcinogenic and have a mutagenic effect. Laccase are used for degradation of phenols, as chlorination rate increases the degradation decreases. In presence of nitrogen laccase reduces lignin content from sugarcane up to 36% in 24 hours .

2.3 Paper and Pulp Industry

Preparation of paper requires chlorine and oxygen containing oxidants which are needed for separation as well as degradation of lignin. But problems such as recycling, cost, toxicity still exist.

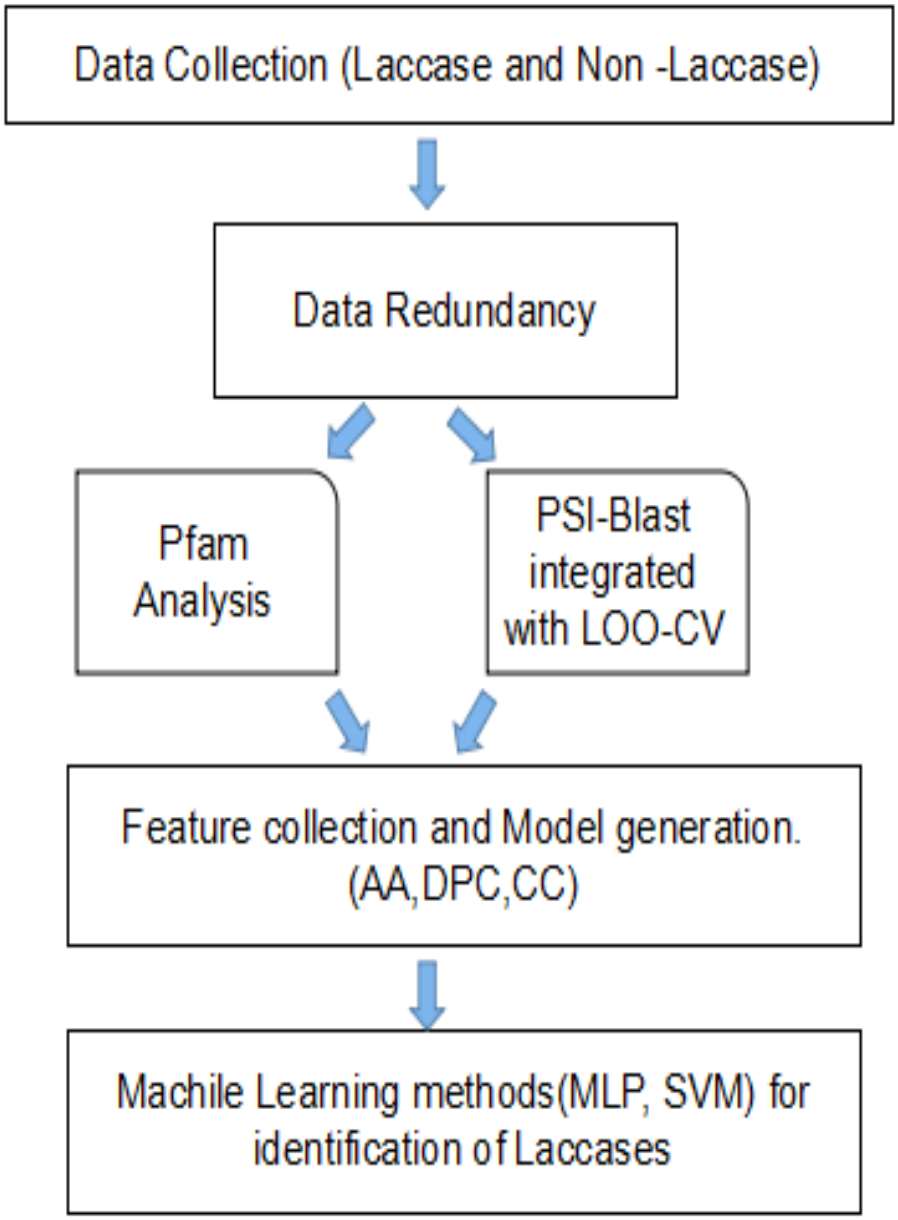
2.4 Food Processing Industry

Laccase plays a role in removal of unwanted phenolic compound in the process of baking, stabilization of wine, juice processing. It improves the sensory and functional characteristics. It increases the stability and shell life of beer. High temperature effects the beer resulting in haze formation which is inhibited by laccase. Phenol compounds are responsible for flavor and color in the juice but due to polymerization and oxidation color and aroma changes, this process is called enzymatic darkening. Laccase is helpful in removing phenol which results in color stability and membrane filtration. With the help of laccase there is an improvement in baked products, volume and softness increases, stickiness decreases. There is a decrease in processing cost and increase in storability.

2.5 Other Applications

Laccase remove foul smell from the garbage dumping sites. These act as a catalyst for organic substances and are used in the designing of biofuel cell. They are useful in bioremediation and fermentation of organic matter. They play a major role in hardening of cuticles in insects and oxidation of toxic polymers to non-toxic polymers.

3. Workflow of Project



(Fig.2)

4. Data Collection

Data collected from NCBI in the category of protein database.

Laccase dataset (positive dataset)

Some specific keywords used:

"Laccase NOT hypothetical NOT putative NOT partial NOT potential NOT patent"

Some of the predefined parameters used:

NOT hypothetical: Data that is not real.

NOT putative: That hasn't been proven or is uncertain.

NOT partial: Incomplete sequence data.

NOT potential: Something that is possible but not actual.

NOT patent: Not novel or new.

In total 5401 sequences were downloaded irrespective of length and sequence similarity.

Non-Laccase dataset (negative dataset)

Some specific keywords used:

""Organism name" NOT Laccase NOT hypothetical NOT putative NOT partial NOT potential NOT patent"

This dataset was designed on the basis of organism name used in the making of positive dataset.

For example: Organism **X** is having 20 number of sequences in positive dataset then we have downloaded 20+ sequences of organism **X** having minimum sequence length of 400 amino acids for negative dataset.

Some of the specific organism having high sequence similarity with Lacasse were also added to the negative dataset:

- Lytic polysaccharide monooxygenases (LPMOs)
- Ascorbate oxidase
- Ferro oxidase
- Nitrite reductase
- Ceruloplasmin

These sequences of specific organisms were added in order to prove that our machine learning based prediction method for Laccase is specific and sensitive.

In total 3000 sequences were downloaded irrespective of length and sequence similarity.

5. Decrease Redundancy

CD-HIT is a program broadly used for clustering of biological sequences to improve performance and decrease redundancy in the sequences. CD-HIT was created for clustering of protein sequences to obtain databases with lower redundancy and was further used for clustering of nucleotide sequences. With the help of commands used in CD-HIT user can compare sequences. Currently CD-HIT has many utility scripts and programs to run the CD-HIT job.

Cd hit tool kit version v4 6.1 was used on the positive as well as on the negative dataset. Criteria that were used for filtering on both the dataset were:

- Minimum sequence length

  Sequence with less than or equal to 400 amino acids were included.
- Minimum sequence similarity

  Sequence with less than or equal to 40% similarity were included.

The command used: “cdhit -i input.fa -o output.fa -c **0.40** -l **400** -n 2”

Sequence left in positive dataset after filtration were: 1047

Sequence left in negative dataset after filtration were: 1076

Further manual screening was performed in which part of the sequence containing non amino acid alphabets were removed.

Final dataset obtained:

Positive dataset-1041 sequences

Negative dataset-1059 sequences

6. Pfam Analysis

Pfam is a collection of protein families including multiple sequence alignment and annotations obtained using Hidden Markov models. It provides a full and precise classification of protein families along with their domains. With the help of Pfam efficiency of annotating genomes has been improved. This classification is widely accepted by biologists and scientist because of large number of proteins classified in it.

Pfam is the creator of IPfam which enlists domain-domain interactions between proteins on the basis of structure databases and domain mapping onto structures.

Pfam has several features for each family such as multiple alignment, protein domain architecture, well known proteins, relation to other databases and species distribution.

Analysis using pfam database version 32.0 (September 2018, 17929 entries). Pfam Analysis was performed in order to prove sequence alignment methods are not good enough to identify Laccases.

In support of the above statement result obtained were as follows:

Command used: ./pfam-scan.pl -fasta sequence40_400.fasta -dir

Total number of sequences in positive dataset were 1041 out of which 350 were identified as Laccase sequences after pfam analysis and rest 697 were identified as non-Laccase sequences. This states that only 33.58% sequences in positive dataset were identified correctly as Laccase.

7. PSI Blast using LOO-CV

PSIBLAST repeatedly searches for more than one protein databases for finding similar sequence to the query sequence. It is similar to BLAST but it uses position specific scoring matrices during the search.

PSIBLAST stands for position-specific iterated Blast, it is more sensitive than blast because it finds distant related sequences which are not present in Blast. It uses multiple alignment for formation of a new PSSM using high scoring sequences. Iteration by PSIBLAST will stop if a new sequence is found.

PSIBLAST uses a search method based on statistics to find section of similarity between the query sequence and database sequence and forms gapped alignment of those sections. PSIBLAST works on the algorithm of blast. The three distinctive features of PSIBLAST are - use of PSSM matric. Composition on the basis of statistics and iterative searching.

Blastpgp package was downloaded and was integrated in LOO-CV python script in order to prove that even domain-based methods are not good enough to identify Laccases.

In support of the above statement result obtained were as follows:

Integration of blastpgp command in LOO-CV script:

- os.system(blastpgp -d training.txt -i testing.txt -j 3 -h 0.001 >> output)

Python script for LOO-CV:

```
import os
fh=open("sequence40_400.fasta","r")
seq= fh.read()
arrseq=seq.split("\n\n")
i=1
for i in [0,len(arrseq)]:
    testing=arrseq[i]
    training_1=arrseq[0:i-1]
    training_2=arrseq[i+1:-1]
    training= training_1+training_2
    with open('testing.txt', 'w') as f:
        for item in testing:
            f.write("%s" % item)
    with open('training.txt', 'w') as f:
        for item in training:
            f.write("%s" % item)
            f.write("\n\n")
    os.system('blastpgp -d training.txt -i testing.txt -h 0.001 -j 3 >> Output')
```

In each iteration data was divided into training and testing dataset. Only one sequence out of 1041 sequences was taken in testing dataset and remaining in training dataset. Blastpgp was performed in every iteration and results were appended into one file.

Out of 1041 laccase sequences 6 sequences were still not identified as laccase.

8. Division of Data

Both positive and negative datasets were divided into test set and training set.

For positive dataset:

Test set-194 sequences

Training set-847 sequences

For negative dataset:

Test set-200 sequences

Training set-859 sequences

9. Feature Collection

Collection of features was done using ProtrWeb server on each and every dataset (positive test set, positive training set, negative test set, negative training set).

Features collected are as follows:

- Amino Acid Composition
- Dipeptide Composition
- Normalized Moreau-Broto Autocorrelation
- Moran Autocorrelation
- Geary Autocorrelation
- C/T/D
- Conjoint triad
- Sequence-Order-Coupling Number
- Quasi-sequence-coupling Number
- Pseudo-Amino Acid Composition
- Amphiphilic Pseudo-Amino Acid Composition

SECTION II

10. PSSM Script

The PSSM(Profile Scoring specific matrix) is a motif descriptor. It attempts to catch intrinsic variability characteristic of sequence patterns. Every coefficient in this matrix represents the occasions of a given nucleotide being assigned its position. More regularly than the absolute frequencies, the relative frequencies are classified in the profile. Frequently in this way, the coefficients in the Position Weight Matrix are directly processed as log-likelihood esteems agreeing with the change log(Mij/pi), where Mij is the likelihood of nucleotide i at position j in the Matrix M, and pi is the foundation likelihood of nucleotide i.

We used PSI-BLAST generated PSSM profiles as a training feature. In this case, PSI-BLAST iterative search was performed against the swiss-prot NCBI database, with a cut-off E-value of 0.001. Calculation is performed for both negative and positive dataset and further it was divided into test set and training set of negative and positive dataset.

10.1 PSSM Algorithm:

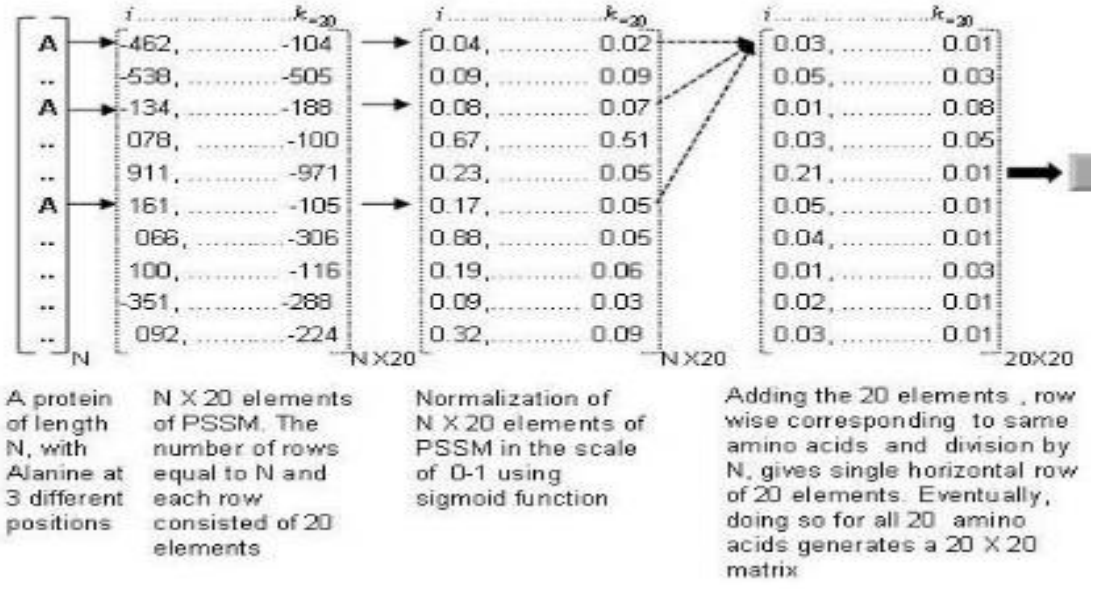
```
Var_x=open_file("laccase_sequence_file/non_laccase_sequence_file")
Var_y= read_file_into_String
Var_z=split_string_into_individual_sequence
Var_k=find_number_of_sequences

for o in range(1 to k+1):
        var_a=insert_1st_sequence
        var_b=split_seq_by_"\n"
        var_acc_seq=Join_seq_into_one_string
        var_seqlen=length_of_acc_seq

        OPEN FILE:
```

```
{
        Right_acc_seq_into_it}

os.system(formatdb_swissprot')
os.system('BLAST_PACKAGE DATABASE TESTING_FILE
NUMBER_OF_ITERATION ERROR_VALUE
“PSSM_MATRIX_GENERATION(-Q)” OUTPUT_FILE')
CHANGE OUTPUT_FILE > OUTPUT_FILE.csv

Os.systen(Use_regular_expression_for_proper_indentation)

with open("Open_result_file.csv") as f
    var_lines=read_file_into_string
data=split.lines_”\n”
pop_index[0]
var_data_eachcell=[i.split_” ” for i in data]
for j in range(1 to 21)
    for i in range(1 to length_of_data_eachcell))
        var_temp=float(data_eachcell[index(i)][index(j)])

        var_sig=Use_sigmoid_function
        replace_[index i]_[index j]data=str(sig)

        var_tmp_list=[[0 for j in range(21)] for i in range(21)]
for i in range(21)
    update_var_tmp_list[0][i]=data_eachcell[0][i]
    update_tmp_list[i][0]=data_eachcell[0][i]
for i in range(1 to21)
    for j in range(1 to 21)
        var_tmp_list_sum=empty_array
        for k in range(1 to length_of_data_eachcell)
            Check_if(data_eachcell[k][0]==tmp_list[j][0]):

              tmp_list_sum.append(float(data_eachcell[k][i]))
            update_sum_sig=sum(tmp_list_sum)
              update_sum_sig=(sum_sig/int(length_of_sequence))
              update_tmp_list[i][j]=str(sum_sig)
update_tmp_list = (list(map(lambda x: x[:0]+x[1:], tmp_list)))
str2ryt=[" ".join(i) for i in tmp_list]
str2ryt_1=" ".join(str2ryt)
open_file_laccase_PSSM.txt_in_append_mode as f_ryt:
    Append_str2ryt_1_into_f_ryt
```

Using the above code, a matrix is generated of 20*N dimensions for each of the single sequence. 20 is the number of amino and N is sequence length. Each element represents the frequency of occurrence of each of the 20 amino acids at a particular position in the alignment. Subsequently, sigmoid function was used to normalize the final PSSM wherein, each matrix element was scaled to a range 0-1.

Sigmoid function **f(x) = 1/1+e−x**

To make the input for tensorflow and weka we summed up every one of the columns in the PSSM comparing to a similar amino acid in the sequence followed by division of each element by the length of the sequence.



(Fig.3)Reference: https://doi.org/10.1186/1471-2105-9-62

11. Weka Optimization

Weka is the gathering of a machine learning algorithm for information mining undertakings. The calculations can either be connected directly to a dataset or can be called from your own Java code. Weka contains devices for data pre-processing, classification, regression, clustering, association rules, and visualization. It can also be appropriate for growing new machine learning methods.

Current stable version, Weka 3.8 was used for the classification of data. SMO package with RBF kernel was used for the classification. Classification was performed on all of the 12 features.

11.1 Training set

INPUT DATASET: Negative training and Positive training dataset was combined into single one .csv file with labels assigned to each class. “neg” class label was given for the Negative training set and “pos” class label was given for the Positive training set.

11.2 Testing set

INPUT DATASET: Negative and Positive test dataset were combined into one .csv file with labels assigned to each class. “neg” class label was assigned for the Negative test set and “pos” class label was assigned for the Positive test set.

11.3 Results

Optimization was performed on the training dataset using SMO classifier with RBF kernel. Wide range of “C” and “Gamma” was used in order to optimize the results. Following were the results:

FOR AAC

C	Gamm	TP	FP	Pricision	Recall	F-Measure	MCC	ROC Area	PRC Area
105	0.45	0.83	0.17	0.83	0.83	0.83	0.66	0.83	0.774

FOR Amphiphilic-Pseudo-Amino Acid

C	Gamm	TP	FP	Pricision	Recall	F-Measure	MCC	ROC Area	PRC Area
500	0.01	0.79	0.21	0.786	0.785	0.785	0.57	0.786	0.724

Conjoint Training

C	Gamm	TP	FP	Pricision	Recall	F-Measure	MCC	ROC Area	PRC Area
170	0.01	0.71	0.29	0.713	0.712	0.712	0.43	0.712	0.651

C_T_D

C	Gamm	TP	FP	Pricision	Recall	F-Measure	MCC	ROC Area	PRC Area
500	0.01	0.71	0.29	0.713	0.712	0.712	0.43	0.712	0.651

Geary Autocorrelation

C	Gamm	TP	FP	Pricision	Recall	F-Measure	MCC	ROC Area	PRC Area
1	0.01	0.94	0.06	0.946	0.939	0.939	0.89	0.939	0.915

Moran Autocorrelation

C	Gamm	TP	FP	Pricision	Recall	F-Measure	MCC	ROC Area	PRC Area
17	0.01	0.58	0.42	0.578	0.578	0.578	0.16	0.578	0.545
17	0.03	0.58	0.42	0.578	0.578	0.578	0.16	0.578	0.545

Normalized Moreau Broto Autocorrelation

C	Gamm	TP	FP	Pricision	Recall	F-Measure	MCC	ROC Area	PRC Area
60	0.01	0.65	0.35	0.654	0.653	0.653	0.31	0.653	0.6

Pseudo Amino Acid Composition

C	Gamm	TP	FP	Pricision	Recall	F-Measure	MCC	ROC Area	PRC Area
1300	0.01	0.79	0.21	0.787	0.787	0.787	0.57	0.787	0.726
1200	0.01	0.79	0.21	0.788	0.787	0.787	0.58	0.787	0.726

Quasi Sequence Order Description

C	Gamm	TP	FP	Pricision	Recall	F-Measure	MCC	ROC Area	PRC Area
720	0.02	0.8	0.2	0.797	0.797	0.797	0.59	0.797	0.737

Sequence Order Coupling Number

C	Gamm	TP	FP	Pricision	Recall	F-Measure	MCC	ROC Area	PRC Area
9000	0.01	0.64	0.36	0.648	0.637	0.631	0.29	0.638	0.589

Dipeptide Amino Acid Composition

C	Gamm	TP	FP	Pricision	Recall	F-Measure	MCC	ROC Area	PRC Area
1	0.01	0.85	0.18	0.825	0.849	0.837	0.67	0.835	0.775

FOR PSSM

C	Gamm	TP	FP	Pricision	Recall	F-Measure	MCC	ROC Area	PRC Area
70	0.45	0.84	0.16	0.843	0.842	0.842	0.69	0.842	0.788

Training dataset results(fig.4)

FOR ACC (C=105, Gamma=0.45)

TP	FP	Precision	Recall	F-Meası	MCC	ROC Ar	PRC Area	class
0.835	0.19	0.81	0.835	0.822	0.645	0.823	0.758	pos
0.81	0.165	0.835	0.81	0.822	0.645	0.823	0.773	neg
0.822	0.177	0.823	0.822	0.822	0.645	0.823	0.765	

FOR Geary Autocorrelation (C=1, Gamma=0.01)

TP	FP	Precision	Recall	F-Meası	MCC	ROC Ar	PRC Area	class
0.515	0.09	0.847	0.515	0.641	0.464	0.713	0.675	pos
0.91	0.485	0.659	0.91	0.765	0.464	0.713	0.646	neg
0.716	0.29	0.752	0.716	0.704	0.464	0.713	0.66	

FOR PSSM (c=70, Gamma=0.45)

TP	FP	Precision	Recall	F-Meası	MCC	ROC Ar	PRC Area	class
0.784	0.11	0.874	0.784	0.826	0.678	0.837	0.791	pos
0.89	0.216	0.809	0.89	0.848	0.678	0.837	0.776	neg
0.838	0.164	0.841	0.838	0.837	0.678	0.837	0.783	

Test dataset results(fig.5)

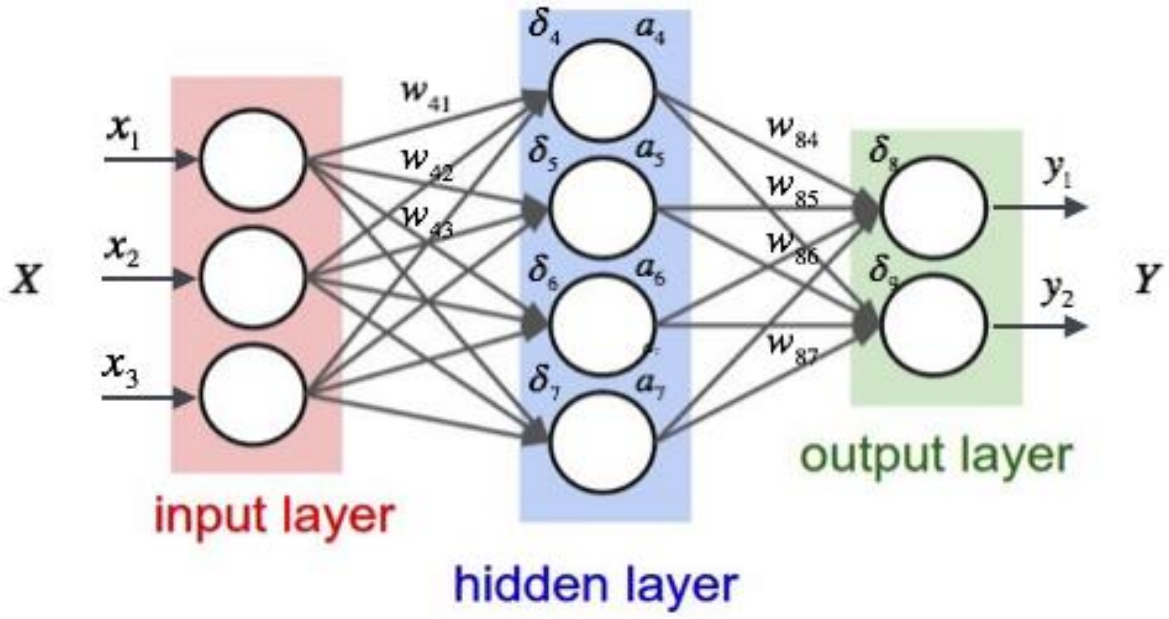
Bellow mentioned three features outperformed the rest on training set:

1. Amino Acid Composition
2. Geary Autocorrelation
3. Profile Scoring specific matrix

So, we generated the results for the same features on test dataset using the same value of “C” and “gamma” and result Fig.5 depicts the output. We obtained similar or nearby values for all the following parameters: True Positive value, False Positive value, Precision, Recall, MCC etc for PSSM and ACC. This predicts that we have achieved the optimization Globally for these two features. Geary Autocorrelation performed exceptionally well on training set but failed to perform satisfactorily on test set. In results for the test set positive class, it depicted only 51.5% sequence as true positive and 9% as false positive whereas for the test set negative class it recognized 91% as true positive and 48.5% as false positive.

It is clear from the confusion matrix of Geary Autocorrelation that overfitting is taking place so this feature is not good enough for the prediction of Laccase enzyme sequence. Out of 12 features of laccase sequence we have left with two, that are giving more than 80% accuracy for the prediction of laccase enzyme sequence and that are AAC(Amino acid composition) and PSSM(Profile scoring specific matrix). In order to confirm our results, we generated the results for these two features on "TensorFlow: neural network".

TensorFlow: neural network uses number of dense layers with different number of neurons in each layer. Through backpropagations and repeated iteration, neurons are taught to predict the desired result with high accuracy.



(Fig 4.)Reference: Creating a Neural Network from Scratch-TensorFlow for Hackers (Part IV)

12.TensorFlow

TensorFlow is free and opensource software since Nov9, 2015. Before 2015 TensorFlow was used within the Google firm for the research and production purpose. They have released it under the “Apache License 2.0” in the year 2015. Since then it’s been used for differentiable programming and dataflow. TensorFlow was developed by Google Brain department which specifically deals with machine learning and Artificial intelligence. Number of companies are using TensorFlow, some of them are Airbnb, Coca-cola, DeepMind, GE Healthcare, intel, twitter, NERSC etc.

TensorFlow is available for 64bit MacOS, windows, Linux and also for mobile computing including Android and iOS. Its flexible architecture allows its availability for such a wide range of platforms(CPUs, TPUs, GPUs). TensorFlow calculations are communicated as stateful dataflow graphs. The name TensorFlow gets from the activities that such neural network perform on multidimensional data, which are alluded to as tensors

12.1 Why TensorFlow

1. Uses Keras API.
2. Multiple level of abstraction.
3. Trains and deploys model easily.
4. Training set results and testing set result are predicted simultaneously.
5. Flexibility in adding the number of dense layer and number of neurons in each layer.

12.2 TensorFlow: neural network Algo

Importing Tensorflow Package

Var x= Keras_dataset

Import “Training data set”, “Test data set”, “Training data class”, “Test data class”

var y= Keras Model

{

Introducing Dense Layer1(number of nurons, input shape(no. of feature in each model))

Introducing Dense Layer2(number of nurons, activation Function (function_name))

Introducing Dense Layer1(number of nurons)

.

.

.

Introducing Dense LayerN(number of nurons)

}

Modelcompiler(optimizer=”Optimizer_name”, loss, metrics)

Modelfit(“training_file_name”,”training_file_class”,Epoch=”Number_of_iterations”, batchsize=”batch_zize”)

Modelevaluate(“test_file_name”, “test_file_class”)

12.3 TensorFlow result:

PSSM

Number_of_layers	Number_of_neuron	Training_se	Test_set	epoch	batch_size
3	20,19,5	0.81	0.76	90	500
4	20,10,5,3	0.8062	0.77	90	500
4	20,10,5,3	0.75	0.8	70	500
4	20,10,5,3	0.79	0.82	70	900
5	4,4,4,4,4	0.77	0.83	80	900

AAC

Number_of_layers	Number_of_neuron	Training_se	Test_set	epoch	batch_size
4	80,40,50,70	0.8	0.66	80	900
4	70,70,50,70	0.82	0.68	80	900
5	100,70,50,80,90	0.83	0.7	80	900

Fig (5.) TensorFlow Results

TensorFlow results proved that PSSM(Profile Scoring Specific matrix) and AAC(Amino Acid Composition) features are good enough to predict whether the given sequence is laccase or non laccase. Using these two features we can extract the laccase sequences from the large datasets.

PSSM has given the accuracy rate of about 75%-85% using the different number of layers and neurons in each layer. Five dense layers were used and 4 neurons were assigned to each layer, as a result we have obtained one of the best results obtaining the accuracy of about 83%(±5%) for the test set. The batch size is of about 900 out of 1705 sequences, that is there will be two steps in each iteration. The number of iterations are 80.

AAC has given the accuracy of about 80%-83% for the training set and 65%-75% for the test set. Using five dense layers, 100 neurons for the first layer, 70 neurons for the second, 50 neurons for the third, 80 for the forth

layer and 90 for the fifth layer, gave us the best output where number of iterations are 80 and the batch size is of 900. All these parameters are optimized to get the best result. Test set and training set accuracy difference is in range of ±5%, therefore we have considered this set of results having best accuracy.

13. Comparison Between TensorFlow and Weka

we have used the latest version of both weka and TensorFlow. Weka 3.8 is latest and advance version of weka that is available as open source whereas TensorFlow 2.0 Alpha is the latest version provided by google as the opensource software.

Weka provides confusion matrix with other parameters like specificity, MCC, ROC area, PRC area, Precision, recall, F-Measure in results, which in turn helps in gaining wide-angle knowledge and information to predict the results more precisely, whereas in TensorFlow we have to write a program separately for all these calculations. TensorFlow provides result simultaneously for both test and training set.

We have to generate results separately for test set and training set in WEKA, which is a tedious process whereas in case of TensorFlow, training and test set accuracy is calculated simultaneously. We must assure that the accuracy difference is not more than ±5% between test set accuracy and training set accuracy.

14. Conclusion

We started this project by proving that domain based method as well as similarity based method are not good enough to predict whether the given sequence is of laccase or not. We constructed the set of features of laccase and non laccase sequence dataset, and with the help of these features we have trained the model to predict the laccase sequence with high accuracy. Three features out of twelve, predicted the results with accuracy of more than 80% in weka 3.8. The three features are Amino Acid Composition, Geary Autocorrelation and Profile Scoring Specific Matrix. Out of these three features, Geary Autocorrelation performed exceptionally well on training set but failed to perform satisfactorily on test set, which means that there is overfitting. To confirm the results, we performed the model optimization using TensorFlow software which is much more advance than weka 3.8. TensorFlow predicted the results with accuracy of 80-85% for Amino Acid Composition and Profile Scoring Specific Matrix. In between these two features, Profile Scoring Specific Matrix outperformed the Amino Acid Composition. So, in order to predict the laccase enzyme sequences, we must prefer using Profile Scoring Specific Matrix in TensorFlow along with the above mentioned parameters given in Fig 5.

References

[1] Miguel Alcalde and Diana M. Mate, Laccase: a multi-purpose biocatalyst at the forefront of biotechnology, 2016

[2] Alcalde M., Laccases: biological functions, molecular structure and industrial applications In Industrial Enzymes. Structure, Function and Applications, Springer, 2007

[3] Brijwani K., Rigdon A., and Vadlani P.V., Fungal laccases: production, function, and applications in food processing., 2010

[4] Conrad L.S., Sponholz W.R. and Berker O., Treatment of cork with a phenol oxidizing enzyme, USA, 2000, US6152966.

[5] Cd-hit: http://weizhongli-lab.org/cd-hit/

[6] Pfam:https://pfam.xfam.org/

[7] Blastpgp-http://nebc.nox.ac.uk/bioinformatics/docs/blastpgp.html

[8] ProtrWeb-http://protr.org/

[9] Aarti Garg and Dinesh Gupta, VirulentPred: a SVM based prediction method for virulent proteins in bacterial pathogens, 2008

[10] weka: https://www.cs.waikato.ac.nz/ml/weka/

[11] weka: https://www.cs.waikato.ac.nz/ml/weka/book.html
[12] TensorFlow:https://www.tensorflow.org/overview/

[13] TensorFlow: https://www.tensorflow.org/tutorials/

[14] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5658592/