**Non-classical secreted protein prediction using Bayesian Hyperparameter Optimization and Stacked Ensemble Classifier**

# Abstract

Gram-positive bacteria play important role in biotechnology due to its single-membrane (monoderm) property facilitating isolation and purification of active protein. In addition to the “classical” protein secretion through their membrane and cell wall via Sec or Tat transporter, gram-positive bacteria also secrete protein via other pathways which is considered “non-classical’ way. The proteins secreted via “non-classical” pathway are difficult to identify due to the lack of any apparent signal peptide. Currently several machine learning methods have been developed to facilitate identification of such proteins. We would like to present an alternative predictive framework to identify the “non-classical” secreted proteins from sequence data.

First we utilize iFeature python package (Chen, Zhao et al. 2018) to generate 5 protein feature descriptors of the training and independent test data. Thereafter we scale the training and independent data separately and use scaled the training data to build independent classifiers. We compare the AUC of classifiers on each of the 5 protein feature descriptors and select protein feature descriptor with best AUC to tune their hyperparameter using Bayesian hyperparameter tuning package, Optuna. Finally, the optimized individual classifiers are used to form base models of Stacked Ensemble Classifier and Logistic Regression is employed as meta classifier to generate final prediction. The hyperparameter optimized Stacked Ensemble classifier can successfully predict non-classical secreted protein with AUC of 88%, ACC of 88% and F1\_score of 87% when ran against independent test data.

# Introduction

Protein secretion is a fundamental physiological process of cells in living organism ranging from bacteria to humans. In our modern world, bacterial protein secretion system from both gram-positive and gram-negative bacteria are employed in biotechnology industry for recombinant protein production. In particular a number of gram-positive bacteria play important role as hosts for production of heterologous protein due to its monoderm property. Gram-positive bacteria can secrete protein across their membrane and cell wall using classical Sec transolocon and Tat (twin-arginine translocation) (Green and Mecsas 2016) transporter as well as other means which is considered “non-classical” channel. Non-classical secretion is challenging to predict due to lack of signal peptide found in classical protein secretion (Wang, Chen et al. 2013). Given its importance in biotechnology industry, numerous studies have been performed on identification of non-classically secreted Gram-positive bacterial protein.

The prediction of non-classical secreted protein is essentially a binary classification task.

In recent study, (Zhang, Yu et al. 2020) curated benchmark dataset from all experimentally verified, non-classically secreted Gram-positive bacterial proteins which was used to train Light Gradient Boosting machine based model optimized using particle swarm optimization strategy. Earlier study by (Restrepo-Montoya, Pino et al. 2011) developed NClassG+ based on Support Vector Machine (SVM) to predict non-classically secreted Gram-positive bacterial protein based on different sequence transformation vectors with reasonable performance when tested on independent data set.

In this paper, we attempt to simplify the classification of non-classically secreted Gram-positive bacterial protein by identifying single best protein feature descriptor through comparison of AUC score generated by baseline classifier for 5 protein feature descriptors namely TPC, DPC, CTDC, CTriad and PAAC. Secondly, we construct stacked ensemble classifier using hyperparameter optimized Random Forest, Extremely Randomized Trees Classifier and Linear SVM as base models and Logistic Regression as meta-classifier. Hyperparameter optimization is performed using Bayesian approach through Optuna package (Akiba, Sano et al. 2019). Finally, the optimized stacked ensemble classifier is used to predict independent test data. Prediction result on independent test data indicate that the stacked ensemble classifier is able to achieve ROC-AUC score of 88%.

# Materials and Method

# Dataset

The training and independent data set were constructed by (Zhang, Yu et al. 2020) collected from <http://pengaroo.erc.monash.edu/>. There are 446 negative sample and 141 positive samples in training data set while the independent test set contains 34 samples for positive and negative respectively.

# Feature Extraction method

Feature extraction were performed using iFeature python package (Chen, Zhao et al. 2018) and of which 5 features, PAAC (Chou, 2001), CTDC (Dubchak et al), DPC (Bhasin and Raghava) , TPC and CTRIAD were selected for study to assess the potential for predicting non-classical protein secretion.

## PAAC - Pseudo Amino Acid Component

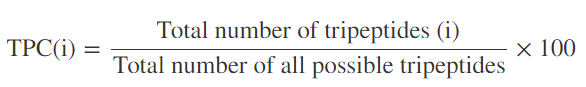
Pseudo Amino Acid Component (PAAC) was proposed by (Chou 2001) to encode and extract numerical vector from raw sequence.



where ,  is the frequency of the amino acid.  is  sequence correlation factor.

## TPC – Tripeptide Composition

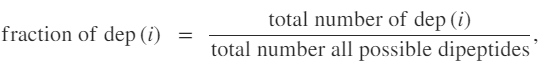
Tripeptide Composition is a percentage composition of each of the 9000 possible dipeptides formed by 20 amino acids and was calculated using the formula below:



where TPC(n) is a tripeptide n out of 8000 tripeptides (Gupta, Sharma et al. 2017).

## DPC – Dipeptide Composition

This protein feature represents information about amino acid composition along local order of amino acid. The fraction of each dipeptide was calculated using following equation:



where dep(i) is one out of 400 dipeptides (Bhasin and Raghava 2004).

## CTD – Composition Transition Distribution

Composition-transition-distribution (CTD) was introduced in 1995 by (Dubchak, Muchnik et al. 1995). In CTD, C stands for composition of amino acids, T (Transition) represents percentage with which frequency of amino acids with specific properties is followed by amino acids with other properties and D (distribution) determines the length of the sequence within which the 1st as well as 25, 50 and 75 percent of amino acids of certain characteristics are located (Meher, Sahu et al. 2018).

Composition (C) consists of 3 values representing the division of amino acids into three groups, Polarity, Hydrophobicity, Neutrality). The 3 values are calculated as follows:



where  is the frequency of , and there are twelve 'H', seven 'P', ten 'N'. is the length of the protein sequence, which is twenty-nine

## CTriad – Conjoint Triad

Conjoint Triad descriptor considers properties of one amino acid and its vicinal amino acids by regarding any three continuous amino acids as a single unit (Shen, Zhang et al. 2007).

# Stacked Ensemble Classifier

Stacked ensemble classifier combines individual classifier models to boost the overall performance beyond the performance of individual classifier. Stacked ensemble classifier has been successfully adopted in **protein-protein interactions prediction** project (Chen, Zhang et al. 2020).

The architecture of stacked ensemble classifier involves 2 or more models in level 0 which are called base models. The models in this level fit on training data and their predictions are compiled. A variety of models in level 0 allows different model to learn well on different aspect of the data which is then fed to level 1 (meta-classifier). Model in level 1, also called meta-classifier, learns how to best combine predictions of base model and produce final prediction. In our study, we use Linear Support Vector Machine (Cortes and Vapnik 1995), Extremely Randomized Trees Classifier (Geurts, Ernst et al. 2006) and Random Forest (Breiman 2001) as our base classifiers. Logistic Regression is used as meta-classifier to arrive at final prediction.

# Model Construction and Evaluation

The proposed stacked ensemble classifier and process flowchart is shown in Fig. 1.

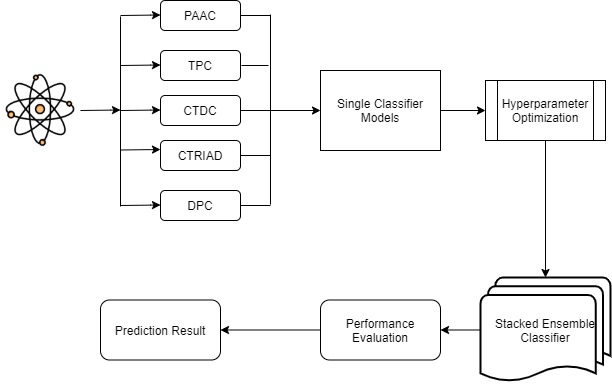


Fig. 1. Framework of Stacked Ensemble Classifier

Protein sequence data is the input of this model. From there 5 protein features are generated using iFeature python package, namely PAAC, TPC, DPC, CTDC and CTriad. Baseline stack models on default parameters are trained for each feature set and AUC of prediction on test data are compared. Best feature, CTDC is selected as the feature for training optimization. Best combination of Base Model namely Linear SVM, Extremely Randomized Trees Classifier and Random Forest models are trained and optimized individually using Bayesian Hyperparameter Optimization package, Optuna. Finally, the optimized single classifiers are stacked as Base Model and Logistic Regression is added as Meta Classifier to produce the final prediction on whether the secreted protein is classical (1) or non-classical (0).

The calculation process of our proposed Stacked Ensemble Classifier can be summarized as follows:

*Step 1:* Use iFeature python package to generate 5 feature descriptor sets from the protein data for comparison

*Step 2:* Training Data and Test Data are separately scaled using Min Max Scaler

*Step 3:* K Nearest Neighbors, Decision Tree, Linear SVM, Extremely Randomized Trees Classifier, Random Forest, SVM, Gaussian NB, XG Boost and a baseline Stacked Ensemble comprised of all classification models as base model with Logistic Regression as meta-classifier are trained separately on each of the 5 feature sets.

*Step 4:* Best performing Feature Set and Model combination in terms of AUC when predicting independent test set are selected for optimization.

*Step 6:* Optimize each of the classifier in Base Model using Bayesian hyperparameter optimization package, Optuna (Akiba, Sano et al. 2019).

*Step 7:* Optimized classifiers are used as Base Model and stacked with Logistic Regression as Meta Classifier to form Optimized Stacked Ensemble Classifier. Default parameters are used for Logistic Regression.

This paper uses Scikit-learn package (Pedregosa, Varoquaux et al. 2011) to implement all the machine learning models and Stacked Ensemble. Ten-fold cross validation is adopted to perform hyperparameter optimization and model evaluation. Samples are randomly divided into 10 parts; 9 parts are employed as training set and the remaining one are employed as test set. The performance of model is assessed based on Area Under the Receiver Operating Characteristic Curve (ROC-AUC).

# Result and Discussion

## Comparison of baseline model results from the 5 feature sets and the effect of Stacked Ensemble.

**Table 1**

AUC of model run against independent test data

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *Feature descriptor* | *K Nearest Neighbors (%)* | *Linear SVM (%)* | *SVM (%)* | *Extremely Randomized Trees (%)* | *Random Forest (%)* | *XG Boost Classifier (%)* | *Stacked Ensemble (%)* |
| TPC | 60.29 | 72.06 | 52.94 | 52.94 | 72.06 | 66.18 | 75.00 |
| DPC | 70.59 | 70.59 | 60.29 | 58.82 | 61.76 | 67.65 | 73.53 |
| **CTDC** | **70.59** | **79.41** | **75.00** | **73.53** | **75.00** | **75.00** | **86.76** |
| CTriad | 73.53 | 72.06 | 61.76 | 52.94 | 51.47 | 58.82 | 64.71 |
| PAAC | 63.24 | 75.00 | 57.35 | 50.00 | 58.82 | 58.82 | 66.18 |

Among the feature descriptors, CTDC performs best across all classifiers individually and when stacked. Stand alone classifier generally performs worse than Stacked Ensemble Classifier regardless of protein feature descriptor. Based on the experiment above, we conclude that CTDC probably better captures useful patterns from protein sequence to predict non-classical secreted protein as compared to other encodings. Stacked ensemble model clearly has an edge over individual classifier due to its ability to combine prediction from multiple weak classifiers to come up with final prediction.

## Base Model Selection

After identifying CTDC as the preferred feature descriptor, we proceed to find the combination of Base Models with minimum complexity and maximum AUC score to finalize the architecture of the stacked ensemble model.

Through iterative process, we identified combination of Linear SVM, Extremely Randomized Trees Classifier and Random Forest as base model and Logistic Regression as meta-classifier produce highest AUC of 86.76% with least number of classifiers in base model.

## Hyperparameter Optimization and its effect

To further improve the performance of our Stacked Ensemble Classifier, we turn into hyperparameter tuning of individual base model. Hyperparameters are important as they directly control behaviours of training algorithm and significantly affect the performance of machine learning models. Unlike internal model parameters which can be learned from data during model training phase, hyperparameters have to be set before training phase (Wu, Chen et al. 2019). Hyperparameter tuning or optimization is the process of finding a set of hyperparameter values that allows machine learning model to achieve best performance on the data.

In general, there are 2 methods of hyperparameter optimization namely manual tuning and automated search. Manual tuning takes time, requires experience of expert users and not easily reproducible (Wu, Chen et al. 2019). Grid search is a form of automated search that do not require supervision once parameters are set but require long run times as it trains models with each combination of possible hyperparameters and evaluate performance on cross validation set. Random Search (Bergstra and Bengio 2012) was proposed to improve the efficiency of grid search by trying random combination of a range of values. However, both methods are uninformed by previous evaluations thus at times spend much of the evaluation time on unpromising hyperparameters.

Bayesian optimization works for hyperparameter tuning where the objective function of optimization is unknown. It combines prior information about the unknown function with sample information, to obtain posterior information of the function distribution by using Bayesian formula. Thereafter, based on this posterior information, we can deduce where the function obtains the optimal value (Wu, Chen et al. 2019).

To optimize the hyperparameter of base models, we adopt Bayesian Hyperparameter Optimization method implemented in Optuna package. The Optuna package can be downloaded at <https://github.com/optuna/optuna>.

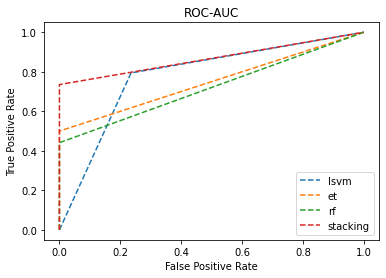
Table 2 below compares the AUC of Stacked Ensemble Classifier with and without hyperparameter optimization.

**Table 2**

Comparison of AUC with and without hyperparameter optimization

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | *Linear SVM (%)* | *Extremely randomized Trees (%)* | *Random Forest (%)* | *Stacked Ensemble (%)* |
| **Without** Hyperparameter Optimization | 79.41 | 72.06 | 75.00 | 86.76 |
| **With** Hyperparameter Optimization | 77.90 | 75.00 | 70.60 | 88.20 |

Overall, the Bayesian based hyperparameter optimization produced better stacked ensemble model performance while consuming reasonable additional parameter tuning time.



# Conclusion

We present a stacked ensemble classifier with Bayesian hyperparameter optimization to predict non-classical secreted protein. Through stacking different classifiers, we achieve better predicting ability beyond what a single classifier could do. We further tuned the hyperparameters of base models through Bayesian optimization method. The AUC after hyperparameter tuning improved from 86.75% to 88.2% when tested on independent test data. Accuracy is 88.2% and f1 score is 86.7%.

We believe this model can be further enhanced as there is still room for improvement in AUC and Accuracy. Different stacking architecture with additional base layer before meta classifier as well as meta-classifier hyperparameter optimization are next research direction. In view of the small training data size, we believe the model will benefit from training on a bigger data set to ensure generalisability.

# References:

Akiba, T., et al. (2019). Optuna: A next-generation hyperparameter optimization framework. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.

Bergstra, J. and Y. Bengio (2012). "Random search for hyper-parameter optimization." The Journal of Machine Learning Research **13**(1): 281-305.

Bhasin, M. and G. Raghava (2004). "ESLpred: SVM-based method for subcellular localization of eukaryotic proteins using dipeptide composition and PSI-BLAST." Nucleic acids research **32**(suppl\_2): W414-W419.

Breiman, L. (2001). "Random forests." Machine learning **45**(1): 5-32.

Chen, C., et al. (2020). "Improving protein-protein interactions prediction accuracy using XGBoost feature selection and stacked ensemble classifier." Computers in Biology and Medicine **123**: 103899.

Chen, Z., et al. (2018). "iFeature: a python package and web server for features extraction and selection from protein and peptide sequences." Bioinformatics **34**(14): 2499-2502.

Chou, K. C. (2001). "Prediction of protein cellular attributes using pseudo‐amino acid composition." Proteins: Structure, Function, and Bioinformatics **43**(3): 246-255.

Cortes, C. and V. Vapnik (1995). "Support-vector networks." Machine learning **20**(3): 273-297.

Dubchak, I., et al. (1995). "Prediction of protein folding class using global description of amino acid sequence." Proceedings of the National Academy of Sciences **92**(19): 8700-8704.

Geurts, P., et al. (2006). "Extremely randomized trees." Machine learning **63**(1): 3-42.

Green, E. R. and J. Mecsas (2016). "Bacterial secretion systems: an overview." Virulence Mechanisms of Bacterial Pathogens: 213-239.

Gupta, S., et al. (2017). "Prediction of anti-inflammatory proteins/peptides: an insilico approach." Journal of translational medicine **15**(1): 1-11.

Meher, P. K., et al. (2018). "nifPred: Proteome-wide identification and categorization of nitrogen-fixation proteins of diaztrophs based on composition-transition-distribution features using support vector machine." Frontiers in microbiology **9**: 1100.

Pedregosa, F., et al. (2011). "Scikit-learn: Machine learning in Python." The Journal of Machine Learning Research **12**: 2825-2830.

Restrepo-Montoya, D., et al. (2011). "NClassG+: A classifier for non-classically secreted Gram-positive bacterial proteins." BMC bioinformatics **12**(1): 21.

Shen, J., et al. (2007). "Predicting protein–protein interactions based only on sequences information." Proceedings of the National Academy of Sciences **104**(11): 4337-4341.

Wang, G., et al. (2013). "How are the non-classically secreted bacterial proteins released into the extracellular milieu?" Current microbiology **67**(6): 688-695.

Wu, J., et al. (2019). "Hyperparameter optimization for machine learning models based on Bayesian optimization." Journal of Electronic Science and Technology **17**(1): 26-40.

Zhang, Y., et al. (2020). "PeNGaRoo, a combined gradient boosting and ensemble learning framework for predicting non-classical secreted proteins." Bioinformatics **36**(3): 704-712.