PeNGaRoo, a combined gradient boosting and ensemble learning framework for predicting non-classical secreted proteins

Supplementary file

Table of Contents

1. Supplementary Experimental Illustration

SI1. Feature selection

Sophisticated features extracted from multiple different aspects may help improve the performance of the model compared to simpler features; however, the improvement may not always be significant (Guyon and Elisseeff, 2002). They may lead to a negative effect on the model training, such as the puzzle of dimensionality, decrease of performance and possible deviations in the model prediction (Guyon and Elisseeff, 2002; Wang, et al., 2017; Wen, et al., 2016; Zhang, et al., 2018). In order to identify the most contributing feature subsets and exclude the redundant features, the GainRatio method was used to perform feature selection by applying the Weka package (Frank, et al., 2004), which is a well-established feature selection method based on information theory (Frank, et al., 2004; Khatun, et al., 2019; Wang, et al., 2017). For the binary classification problem, the information entropy $H(X)$ can be defined as:

$$H(X) = -\sum_{i} P(x_i)\log_2\big(P(x_i)\big) \quad i = 1,2$$

where x_i is a set of values of X (two possible classes, e.g. positive or negative) and $P(x_i)$ denotes the prior probability of x_i. The entropy of the feature F_j can be defined as:

$$H\big(X|F_j\big) = \sum_{k=1}^{m} P_{F_j=F_k} H(X|F_j = F_k)$$

where *m* is the total number of features. Therefore, the gain ratio can be defined as:

$$GR\big(F_j\big) = \frac{H(X) - H(X|F_j)}{H(X)}$$

SI2. Commonly used machine learning algorithms

I K-nearest neighbor (KNN)

K-nearest neighbor (KNN) is a simple and commonly employed machine learning algorithm that can be used to solve classification and regression problems (Wang, et al., 2017; Zhang, et al., 2018). KNN predicts new samples by evaluating their similarities/distances to the *k* nearest known neighbors. It has been successfully applied in many bioinformatics studies (Chen, et al., 2018; Liang, et al., 2013; Shen and Chou, 2005; Zhang, et al., 2018). The choice of the parameter *k* plays a vital role in determining the performance of the KNN algorithm. In our study, we optimized the parameter *k* to minimize the classification error for values $k = 1, 2, 3, \dots, \lfloor \max\{\sqrt{featureNum}, featureNum/2\} \rfloor$, where *featureNum* is the number of features used for model training.

II Support vector machine (SVM)

Support vector machine (SVM) is an efficient machine learning algorithm and is suitable for solving binary classification, multiple classification or regression problems (Saini, et al., 2015; Song, et al., 2018). SVM has been widely applied to deal with many classification tasks in the fields of the bioinformatics and computational biology (An, et al., 2018; Wang, et al., 2017; Zhang, et al., 2018). In this study, we adopted the Gaussian radial basis kernel for training the SVM models using the software package e1071 (Meyer, et al., 2015) implemented in the R language. We used the grid search to optimize the two essential parameters of SVM: Cost$C \in \{2^{-10}, 2^{-9}, \dots, 1, 2^9, 2^{10}\}$ and Gamma $\gamma \in \{2^{-10}, 2^{-9}, \dots, 1, 2^9, 2^{10}\}$.

III Random forest (RF)

Random forest (RF) is a well-established and widely used machine learning algorithm developed by Leo Breiman (Breiman, 2001). In principle, RF is an ensemble classifier composed of multiple decision trees (Chen, et al., 2018; Song, et al., 2017; Wang, et al., 2017; Zhang, et al., 2018). RF has been successfully applied to solve different classification and regression tasks (Song, et al., 2018; Wang, et al., 2017; Xue, et al., 2018). In the RF, there are two key parameters that need to be specified: the number of the decision trees (*M*) and the number of randomly selected features (*mtry*). Here, we selected *M*=1000, and optimized the parameter *mtry* by its built-in function to train RF model using the randomForest package implemented in R (Liaw and Wiener, 2002).

IV Multi-Layer perceptron (MLP)

Multi-Layer perceptron (MLP) is one of the most widely used artificial neural network models (Dehzangi, et al., 2010; Mirjalili, et al., 2014). MLP has been widely applied to solve various classification problems in bioinformatics (Wang, et al., 2017; Wang, et al., 2006). In this study, we trained the MLP model using the Keras package implemented in R. Specifically, three hidden layers were added to the model, and the number of nodes in each hidden layer was set to 64, with a dropout rate of 0.05. The parameter *epochs* was set to 40 during model training.

2. Supplementary Tables and Figures

Table S1. Statistics of the species-specific data used by PeNGaRoo, including the initially collected data and the data after the sequence redundancy removal.

Species	Initially collected data	Data after sequence redundancy removal
Bacillus subtilis	13	12
Bacillus licheniformis	21	11
Bacillus anthracis	20	14
Bacillus cereus	6	0
Listeria monocytogenes	23	20
Listeria innocua	23	7
Staphylococcus aureus	20	18
Streptococcus pyogenes	20	13
Streptococcus pneumoniae	20	8
Streptococcus agalactiae	14	4
Mycobacterium tuberculosis	24	11
Mycobacterium smegmatis	21	16
Lactobacillus plantarum	15	15
Lactococcus lactis	13	8
Total	253	157

Table S2. Classification of amino acids based on the dipoles and volumes of their side chains.

No.	Dipole Scale1[1]	Volume Scale[2]	Class
1	—	—	Ala, Gly, Val
2	—	+	Ile, Leu, Phe, Pro
3	+	+	Tyr, Met, Thr, Ser
4	+ +	+	His, Asn, Gln, Tpr
5	+ + +	+	Arg, Lys
6	+’ +’ +’	+	Asp, Glu
7	+[3]	+	Cys

Note: [1]Dipole Scale (Debye): −, Dipole < 1.0; +, 1.0 < Dipole < 2.0; ++, 2.0 < Dipole < 3.0; +++, Dipole >3.0; +’ +’ +’, Dipole > 3.0 with an opposite orientation.

[2]Volume Scale (Å3): −, Volume < 50; +, Volume > 50.

[3]Cys is separated from the Class 3 due to its unique ability to form disulfide bonds.

Table S3. Classification of 20 standard amino acid types according to seven specific types of physicochemical properties.

Categorization	**Class 1**	**Class 2**	**Class 3**
Hydrophobicity	Polar R, K, E, D, Q, N	Neutral G, A, S, T, P, H, Y	Hydrophobicity C, L, V, I, M, F, W
Normalized van der Waals volume	0-2.78 G, A, S, T, P, D, C	2.95-4.0 N, V, E, Q, I, L	4.03-8.08 M, H, K, F, R, Y, W
Polarity	4.9-6.2 L, I, F, W, C, M, V, Y	8.0-9.2 P, A, T, G, S	10.4-13.0 H, Q, R, K, N, E, D
Polarizability	0-0.108 G, A, S, D, T	0.128-0.186 C, P, N, V, E, Q, I, L	0.219-0.409 K, M, H, F, R, Y, W
Charge	Positive K, R	Neutral A, N, C, Q, G, H, I, L, M, F, P, S, T, W, Y, V	Negative D, E
Secondary Structure	Helix E, A, L, M, Q, K, R, H	Strand V, I, Y, C, W, F, T	Coil G, N, P, S, D
Solvent Accessibility	Buried A, L, F, C, G, I, V, W	Exposed R, K, Q, E, N, D	Intermediate M, S, P, T, H, Y

Table S4. Description of the 11 parameters required by LightGBM.

Parameters	**Description**[1]	**Parameter tuning range**
learning_rate	shrinkage rate	[2^(-10), 0.9]
num_leaves	number of leaves in one tree	[20, 800]
max_depth	max depth of the tree	[5, 10]
min_data_in_leaf	minimal number of data in one leaf	[2, 32]
max_bin	max number of bins in which feature values will be bucketed	[32, 1024]
feature_fraction	percentage of features selected prior to the training of each tree	[0.5, 1]
min_sum_hessian	minimal sum hessian in one leaf	[0, 0.02]
lambda_l1	L1 regularization	[0, 0.01]
lambda_l2	L2 regularization	[0, 0.01]
drop_rate	only used in dart	[0, 1]
max_drop	max number of dropped trees at one iteration	[1, 100]

Note: [1]The description of the above parameters was retrieved from the official LightGBM document (http://lightgbm.readthedocs.io/en/latest/index.html).

Table S5. Predictive performance of models using different feature encoding methods based on the PSO parameter optimization strategy compared with those based on the initial parameter setting, One-by-one parameter optimization, and GA-based two-step parameter optimization. The performance was evaluated using the 5-fold cross-validation test. The values were expressed as mean ± standard deviation.

Model	PAAC	QSO	TPC	Pse-PSSM	AATP	CTriad	CTDT
Default	0.638 ± 0.036	0.610 ± 0.057	0.688 ± 0.033	0.666 ± 0.039	0.719 ± 0.043	0.609 ± 0.035	0.654 ± 0.039
One-by-one	0.663 ± 0.034	0.646 ± 0.045	0.721 ± 0.030	0.691 ± 0.047	0.712 ± 0.037	0.671 ± 0.035	0.632 ± 0.043
GA-based two-step	0.669 ± 0.045	0.648 ± 0.047	0.728 ± 0.029	0.718 ± 0.035	0.739 ± 0.029	0.676 ± 0.030	0.638 ± 0.029
PSO-based	0.688 ± 0.036	0.673 ± 0.035	0.744 ± 0.027	0.728 ± 0.039	0.747 ± 0.033	0.700 ± 0.030	0.666 ± 0.034

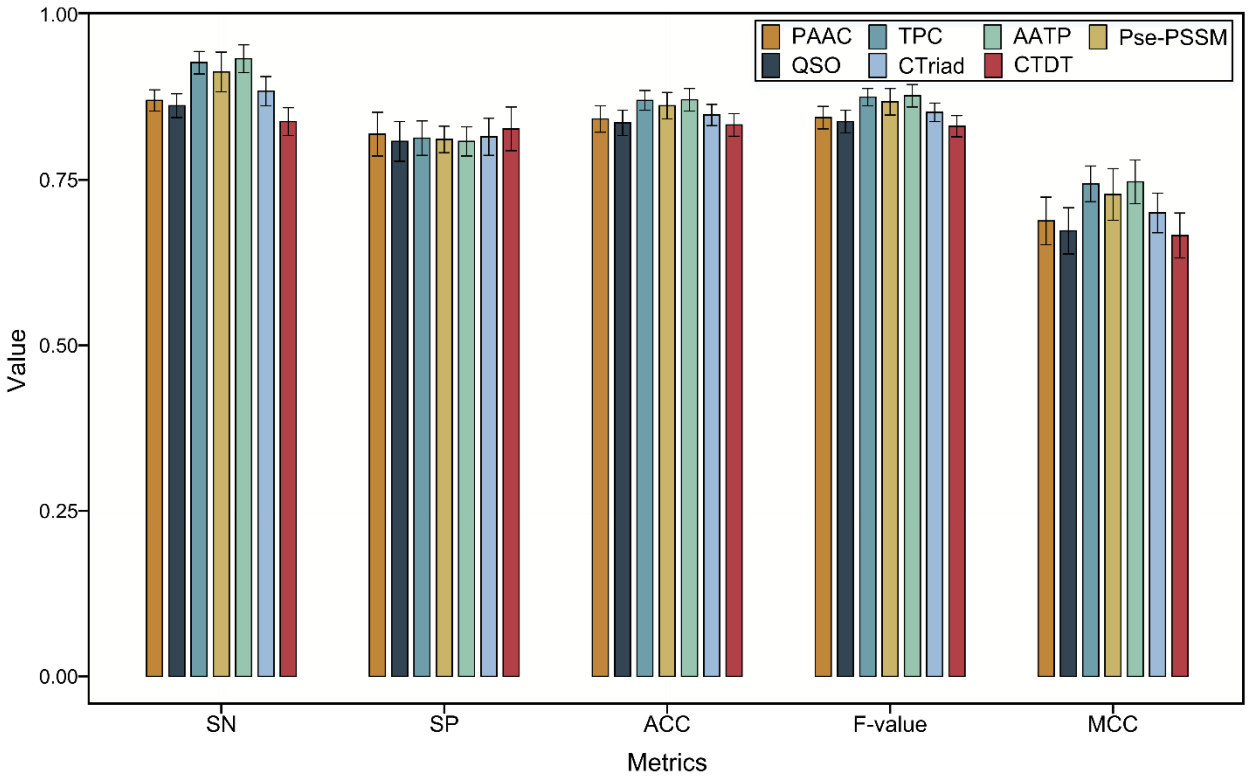

Fig. S1: Performance comparison between models trained using different sequence encoding methods on the 5-fold cross-validation test.

Table S6. Comparison of the predictive performance between models trained using the original features and the features combined with the GainRatio method.

Encoding	Dim	SN	SP	ACC	F-value	MCC
PAAC	50	0.870±0.016	0.819±0.033	0.842±0.020	0.844±0.017	**0.688±0.036**
QSO	50	0.842±0.014	0.810±0.016	0.826±0.014	0.827±0.011	0.653±0.026
	100	0.862±0.018	0.808±0.030	0.836±0.019	0.838±0.017	**0.673±0.035**
TPC	50	0.892±0.016	0.807±0.024	0.850±0.013	0.853±0.012	0.701±0.025
	100	0.930±0.018	0.807±0.025	0.868±0.016	0.874±0.014	0.741±0.032
	150	0.916±0.021	0.816±0.023	0.866±0.017	0.870±0.016	0.737±0.034
	200	0.913±0.028	0.805±0.019	0.860±0.020	0.865±0.019	0.723±0.039
	250	0.916±0.026	0.805±0.021	0.860±0.017	0.865±0.018	0.726±0.036
	300	0.912±0.021	0.809±0.030	0.860±0.019	0.865±0.017	0.726±0.036
	350	0.912±0.032	0.801±0.019	0.856±0.018	0.861±0.019	0.718±0.037
	400	0.927±0.017	0.813±0.026	0.870±0.015	0.875±0.013	**0.744±0.027**
Pse-PSSM	40	0.913±0.030	0.811±0.020	0.862±0.020	0.868±0.020	**0.728±0.039**
AATP	50	0.904±0.024	0.802±0.023	0.853±0.017	0.857±0.017	0.710±0.034
	100	0.929±0.014	0.808±0.025	0.868±0.017	0.874±0.014	0.742±0.031
	150	0.923±0.030	0.806±0.020	0.864±0.020	0.869±0.019	0.734±0.040
	200	0.913±0.030	0.811±0.020	0.862±0.020	0.868±0.020	0.728±0.039
	250	0.913±0.030	0.811±0.020	0.862±0.020	0.868±0.020	0.728±0.039
	300	0.917±0.024	0.809±0.021	0.864±0.020	0.868±0.019	0.731±0.040
	350	0.914±0.030	0.811±0.023	0.862±0.018	0.867±0.017	0.729±0.036
	400	0.927±0.021	0.809±0.033	0.867±0.023	0.873±0.021	0.741±0.045
	420	0.933±0.021	0.808±0.022	0.877±0.017	0.871±0.017	**0.747±0.033**
CTriad	50	0.814±0.038	0.798±0.023	0.805±0.025	0.804±0.029	0.611±0.052
	100	0.846±0.030	0.796±0.026	0.820±0.025	0.823±0.025	0.642±0.048
	150	0.845±0.022	0.807±0.025	0.826±0.019	0.826±0.018	0.652±0.036
	200	0.849±0.030	0.812±0.033	0.829±0.023	0.830±0.023	0.661±0.044
	250	0.860±0.030	0.808±0.020	0.833±0.020	0.835±0.021	0.668±0.039
	300	0.865±0.040	0.811±0.025	0.837±0.030	0.839±0.031	0.676±0.059
	343	0.884±0.022	0.815±0.028	0.848±0.016	0.852±0.014	**0.700±0.030**
CTDT	21	0.838±0.021	0.827±0.033	0.833±0.017	0.831±0.016	**0.666±0.034**

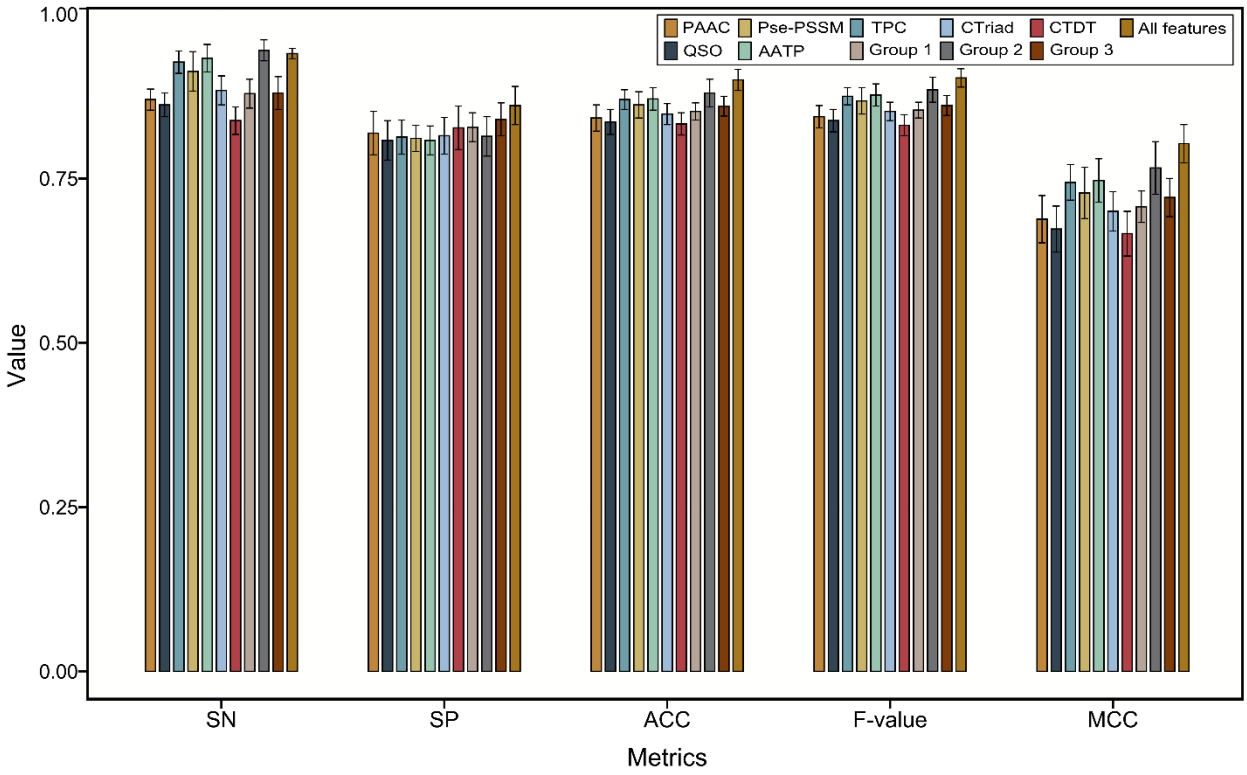

Fig. S2: Performance comparison between single feature-based models, group-based one-layer ensemble models and the final two-layer ensemble model on the benchmark training dataset.

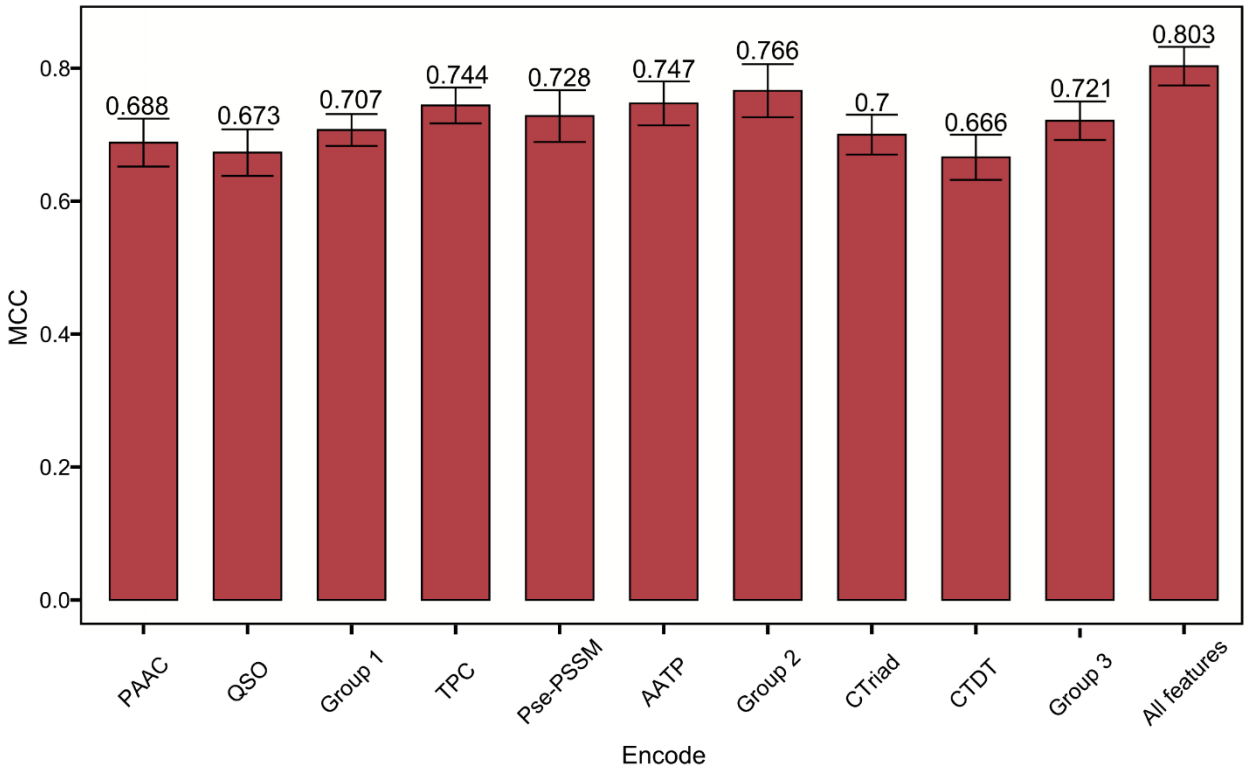

Fig. S3: Performance comparison in terms of the MCC value between single feature-based models, group-based one-layer ensemble models and the final two-layer ensemble models on the benchmark training dataset.

Table S7. Performance comparison of different LightGBM classifiers using the leave-one-out cross-validation test.

Model	Encoding	SN	SP	ACC	F-value	MCC
Sequence-derived features	PAAC	0.870±0.024	0.820±0.022	0.845±0.017	0.848±0.017	0.691±0.034
	QSO	0.850±0.024	0.804±0.017	0.827±0.013	0.831±0.014	0.655±0.026
	Group 1	**0.880±0.019**	**0.830±0.015**	**0.855±0.010**	**0.859±0.010**	**0.712±0.020**
Evolutionary information-based features	TPC	0.935±0.018	0.817±0.029	0.876±0.018	0.883±0.017	0.758±0.036
	Pse-PSSM	0.904±0.013	**0.831±0.028**	0.867±0.017	0.872±0.015	0.737±0.032
	AATP	0.936±0.018	0.819±0.030	0.878±0.021	0.885±0.019	0.761±0.040
	Group 2	**0.946±0.017**	0.821±0.031	**0.883±0.018**	**0.890±0.016**	**0.773±0.034**
Physicochemical property-based features	CTRIAD	0.890±0.018	0.821±0.036	0.855±0.024	0.860±0.022	0.713±0.047
	CTDT	0.838±0.019	0.814±0.016	0.826±0.013	0.828±0.014	0.652±0.027
	Group 3	**0.896±0.015**	**0.829±0.019**	**0.863±0.016**	**0.867±0.015**	**0.727±0.032**
Final model	All features	0.938±0.012	0.851±0.022	0.895±0.015	0.899±0.014	0.792±0.030

Note: Performance was expressed as mean ±standard deviation. The best performance value within each groups of feature-based models is highlighted in bold.

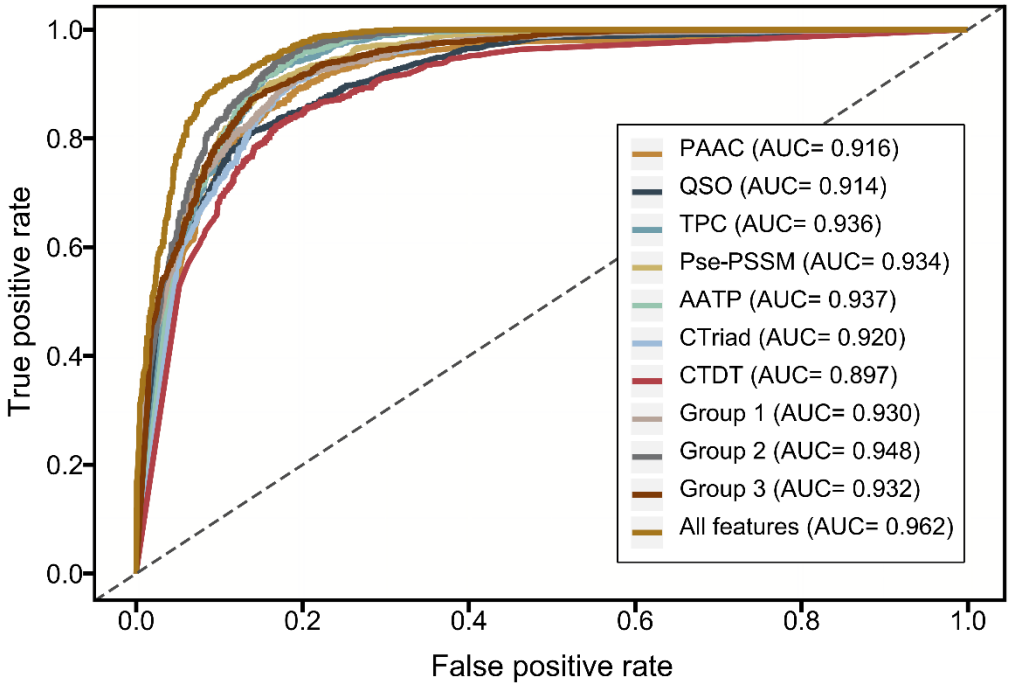

Fig. S4 ROC curves of the models trained using different sequence encoding methods evaluated using the leave-one-out cross-validation test. The AUC values were calculated and shown in the inset.

Table S8. Performance comparison between the models trained using different types of single features, the ensemble models based on groups of features, and our proposed final method PeNGaRoo using the independent test set.

Model	Encoding	SN	SP	ACC	F-value	MCC
Sequence-derived features	PAAC	**0.824**	0.588	0.706	0.737	0.424
	QSO	0.794	0.588	0.691	0.720	0.391
	Group 1	**0.824**	**0.618**	**0.721**	**0.747**	**0.451**
Evolutionary information-based features	TPC	**0.794**	0.559	0.676	0.711	0.363
	Pse-PSSM	0.706	0.529	0.618	0.649	0.239
	AATP	0.765	**0.618**	**0.691**	**0.712**	**0.387**
	Group 2	0.765	0.559	0.662	0.693	0.331
Physicochemical property-based features	CTRIAD	0.676	0.676	0.676	0.676	0.353
	CTDT	**0.853**	0.676	**0.765**	**0.784**	**0.538**
	Group 3	0.794	**0.735**	**0.765**	0.771	0.530
Final model	All features	0.824	0.735	0.779	0.789	0.561

Note: Performance was expressed as mean ±standard deviation. The best performance value within each groups of feature-based models is highlighted in bold.

Table S9. Performance comparison of different classifiers (single feature-based models and ensemble models) based on four machine learning algorithms on the independent test set.

Machine learning algorithms	**Encoding**	**SN**	**SP**	**ACC**	**F-value**	**MCC**
KNN	PAAC	0.765	0.559	0.662	0.693	0.331
	QSO	**0.853**	0.500	0.676	0.725	0.377
	Group 1	0.824	0.588	0.706	0.737	0.424
	TPC	0.706	0.618	0.662	0.676	0.325
	Pse-PSSM	0.794	0.618	0.706	0.730	0.418
	AATP	0.706	0.559	0.632	0.658	0.268
	Group 2	0.735	**0.676**	0.706	0.714	0.412
	CTriad	**0.853**	0.176	0.515	0.637	0.040
	CTDT	0.824	0.588	0.706	0.737	0.424
	Group 3	**0.853**	0.294	0.574	0.667	0.177
	All features	**0.853**	0.588	**0.721**	**0.753**	**0.457**
SVM	PAAC	**0.735**	0.647	0.691	0.704	0.384
	QSO	**0.735**	0.735	0.735	**0.735**	0.471
	Group 1	**0.735**	0.676	0.706	0.714	0.412
	TPC	0.529	0.765	0.647	0.600	0.303
	Pse-PSSM	0.500	0.794	0.647	0.586	0.308
	AATP	0.500	0.794	0.647	0.586	0.308
	Group 2	0.471	**0.824**	0.647	0.571	0.314
	CTriad	0.647	0.735	0.691	0.677	0.384
	CTDT	**0.735**	0.676	0.706	0.714	0.412
	Group 3	0.676	0.706	0.691	0.687	0.383
	All features	0.676	**0.824**	**0.750**	0.730	**0.505**
RF	PAAC	0.765	0.647	0.706	0.722	0.415
	QSO	0.735	0.559	0.647	0.676	0.299
	Group 1	0.794	**0.676**	0.735	0.75	0.474

	TPC	0.706	0.5	0.603	0.64	0.21
	Pse-PSSM	0.735	0.5	0.618	0.658	0.242
	AATP	0.706	0.471	0.588	0.632	0.182
	Group 2	0.735	0.471	0.603	0.649	0.213
	CTriad	0.765	0.647	0.706	0.722	0.415
	CTDT	**0.853**	0.559	0.706	0.744	0.431
	Group 3	0.824	**0.676**	**0.750**	**0.767**	**0.505**
	All features	0.824	**0.676**	**0.750**	**0.767**	**0.505**
MLP	PAAC	0.824	0.559	0.691	0.727	0.396
	QSO	0.676	**0.824**	0.750	0.730	0.505
	Group 1	0.765	0.794	**0.779**	0.776	**0.559**
	TPC	0.676	0.559	0.618	0.639	0.237
	Pse-PSSM	**0.853**	0.441	0.647	0.707	0.323
	AATP	0.676	0.647	0.662	0.667	0.324
	Group 2	0.794	0.529	0.662	0.701	0.335
	CTriad	0.676	0.676	0.676	0.676	0.353
	CTDT	0.794	0.647	0.721	0.740	0.446
	Group 3	0.706	0.765	0.735	0.727	0.471
	All features	0.824	0.706	0.765	**0.778**	0.533

Note: For each of machine learning models, the best performance value for each metric across different encoding methods-based models and ensemble models is shown in bold font.

Table S10. Performance comparison of two-layer ensemble models based on five machine learning algorithms on the independent test set.

	SN	SP	ACC	F-value	MCC
two-layer ensemble LightGBM models	0.824	0.735	**0.779**	**0.789**	**0.561**
two-layer KNN ensemble models	**0.853**	0.588	0.721	0.753	0.457
two-layer SVM ensemble models	0.676	**0.824**	0.750	0.730	0.505
two-layer ensemble RF models	0.824	0.676	0.750	0.767	0.505
two-layer ensemble MLP models	0.824	0.706	0.765	0.778	0.533

Note: The best performance value for each metric across different machine learning algorithms is shown in bold font.

Table S11. The key differences between the proposed PeNGaRoo method and SecretomeP.

	PeNGaRoo	SecretomeP
Positive samples for model training	Experimentally validated non-classical secreted effectors	Secreted proteins after removing signal peptides
Training dataset	Positive: 141; Negative: 446	Positive: 152; Negative: 140
Independent dataset	Positive: 34; Negative: 34	None
Imbalanced problem solving	Yes	No
Feature extraction	PAAC, QSO, TPC, Pse-PSSM, AATP, CTriad, CTDT	Threonine contents, amino acid composition (AAC), Transmembrane helices, Gravy, Protein disorder, Secondary structure
Machine learning algorithm	LightGBM with optimized parameters	Neural networks
Ensemble strategy	Two-layer ensemble method	None
Maximum allowed number of sequences per submission on the server	500	100

Table S12. Performance comparison between PeNGaRoo and the state-of-the-art method SecretomeP for predicting non-classical secreted Gram-positive bacterial proteins on the independent test dataset.

	Classifier	SN	SP	ACC	F-value	MCC
Independent test	SecretomeP	0.353	0.824	0.588	0.462	0.2
	PeNGaRoo	0.824	0.735	0.779	0.789	0.561

Table S13. Performance comparison between PeNGaRoo, SecretomeP and SecretP based on the dataset of non-classical secretory proteins previously compiled by (Bendtsen et al., 2005)[1].

UniProt ID	SecretP[2]		SecretomeP[3]		PeNGaRoo	
	Score[4]	Result	Score	Result	Score	Result
P49814	0	N	0.059	N	0.824	Y
Q06320	1	Y	0.725	Y	0.758	Y
P0DJM2	2	N	0.377	N	0.963	Y
P80868	2	N	0.082	N	0.990	Y
Q8Y422	2	N	0.090	N	0.985	Y
P37869	2	N	0.128	N	0.991	Y
P9WNK7	1	Y	0.647	Y	0.637	Y
P9WNK5	1	Y	0.855	Y	0.687	Y
P39810	2	N	0.831	N	0.840	Y
P39738	1	Y	0.857	Y	0.527	Y
P09124	2	N	0.127	N	0.992	Y
P9WN39	2	N	0.534	Y	0.735	Y
P28598	2	N	0.037	N	0.999	Y
P02968	2	N	0.291	N	0.578	Y
P26901	2	N	0.741	Y	0.734	Y
P21881	2	N	0.150	N	0.896	Y
P21882	2	N	0.051	N	0.997	Y
P21880	2	N	0.062	N	0.992	Y
O53083	1	Y	0.061	N	0.182	N
P39634	2	N	0.091	N	0.986	Y

P39138	2	N	0.110	N	0.427	N
Q9RLT9	2	N	0.116	N	0.083	N
Q8YA96	2	N	0.068	N	0.019	N
Q8Y459	2	N	0.070	N	0.721	Y
P54375	1	Y	0.760	Y	0.960	Y
P9WGE7	1	Y	0.306	N	0.890	Y
P39797	2	N	0.464	N	0.768	Y
P54327	2	N	0.103	N	0.373	N
P54331	2	N	0.736	Y	0.939	Y
P54332	2	N	0.631	Y	0.411	N
P39800	2	N	0.707	Y	0.745	Y
P80875	2	N	0.452	N	0.610	Y

Note: [1]The original list contained a total of 35 proteins, of which only one entry was a Gram-positive bacterium non-classical secretory protein and the other were Gram-positive non-classical secretory proteins. Therefore, after removing this protein, and another two obsolete proteins (i.e. Q4EL63_LISMO and C1L1X5_LISMC), 32 non-classical Gram-positive bacterial proteins remained and were used to assess the performance of the three methods.

[2]The prediction results of SecretP were extracted from the reference paper entitled "SecretP: Identifying bacterial secreted proteins by fusing new features into Chou's pseudo-amino acid composition".

[3]The prediction results of SecretomeP were generated by the web server presented in the article titled "Non-classical protein secretion in bacteria".

[4]For the prediction results of SecretP, “0”, “1” and “2” represent the types of CSP, non-classically secreted protein and non-secreted protein, respectively.

3. Supplementary References

An, Y., *et al.* Comprehensive assessment and performance improvement of effector protein predictors for bacterial secretion systems III, IV and VI. *Briefings in bioinformatics* 2018;19(1):148-161.

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

Chen, Z., *et al.* Large-scale comparative assessment of computational predictors for lysine post-translational modification sites. *Briefings in bioinformatics* 2018.

Dehzangi, A., Phon-Amnuaisuk, S. and Dehzangi, O. Enhancing protein fold prediction accuracy by using ensemble of different classifiers. *Australian Journal of Intelligent Information Processing Systems* 2010;26(4):32-40.

Frank, E., *et al.* Data mining in bioinformatics using Weka. *Bioinformatics* 2004;20(15):2479-2481.

Guyon, I. and Elisseeff, A. An introduction to variable and feature selection. *Journal of Machine Learning Research* 2002;3(6):1157-1182.

Khatun, M.S., Hasan, M.M. and Kurata, H. PreAIP: Computational Prediction of Anti-inflammatory Peptides by Integrating Multiple Complementary Features. *Frontiers in genetics* 2019;10:129.

Liang, L., *et al.* MS- k NN: protein function prediction by integrating multiple data sources. *Bmc Bioinformatics* 2013;14 Suppl 3(Suppl 3):61-64.

Liaw, A. and Wiener, M. Classification and regression by randomForest. *R news* 2002;2(3):18-22.

Meyer, D., *et al.* e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien. R package version 1.6–7. 2015. In.; 2015.

Mirjalili, S., Mirjalili, S.M. and Lewis, A. Let a biogeography-based optimizer train your multi-layer perceptron. *Information Sciences* 2014;269:188-209.

Saini, H., *et al.* Probabilistic expression of spatially varied amino acid dimers into general form of Chous pseudo amino acid composition for protein fold recognition. *Journal of theoretical biology* 2015;380:291-298.

Shen, H. and Chou, K.C. Using optimized evidence-theoretic K-nearest neighbor classifier and pseudo-amino acid composition to predict membrane protein types. *Biochemical & Biophysical Research Communications* 2005;334(1):288-292.

Song, J., *et al.* PREvaIL, an integrative approach for inferring catalytic residues using sequence, structural, and network features in a machine-learning framework. *Journal of Theoretical Biology* 2018;443:125-137.

Song, J., *et al.* PhosphoPredict: A bioinformatics tool for prediction of human kinase-specific phosphorylation substrates and sites by integrating heterogeneous feature selection. *Scientific Reports* 2017;7(1):6862.

Song, J., *et al.* iProt-Sub: a comprehensive package for accurately mapping and predicting protease-specific substrates and cleavage sites. *Briefings in Bioinformatics* 2019, 20(2): 638-658.

Wang, J., *et al.* Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches. *Briefings in Bioinformatics* 2019, 20(3): 931-951.

Wang, L.N., *et al.* Computational prediction of species-specific malonylation sites via enhanced characteristic strategy. *Bioinformatics* 2017;33(10):1457-1463.

Wang, Z., *et al.* Optimized multilayer perceptrons for molecular classification and diagnosis using genomic data. *Bioinformatics* 2006;22(6):755-761.

Wen, P.P., *et al.* Accurate in silico prediction of species-specific methylation sites based on information gain feature optimization. *Bioinformatics* 2016;32(20):3107-3115.

Xue, L., *et al.* DeepT3: deep convolutional neural networks accurately identify Gram-Negative Bacterial Type III Secreted Effectors using the N-terminal sequence. *Bioinformatics* 2018.

Zhang, Y., *et al.* Computational analysis and prediction of lysine malonylation sites by exploiting informative features in an integrative machine-learning framework. *Briefings in Bioinformatics* 2018.