Systematic Analysis and Identification of Substrates Secreted by Gram-negative Bacteria

Jiawei Wang

MSc (Computer Science)

A thesis submitted for the degree of Doctor of Philosophy at

Monash University in 2019

Faculty of Medicine, Nursing and Health Sciences

Department of Microbiology

Copyright notice

Abstract

Gram-negative bacteria are a particularly important cause of various infections and severe disease, causing a huge burden to public health. The collective term “Gram-negative” covers a very wide diversity of bacterial species across four of the six pathogens in the ESKAPE listed by the Infectious Diseases Society of America. These highly diverse species have evolved a wide range of secretion systems as their weapons to export substrate proteins into the surrounding milieu or adjacent target cells. These secreted proteins play vital roles in the struggle against stressful environments, and contribute toward bacterial pathogenesis and their competitive survival in bacterial populations. Benefiting from the advance in computational and experimental techniques, a considerable number of proteins secreted from bacteria have been discovered and further analyzed. These substrates of the various secretion systems differ significantly in the way they are secreted, their structural features and their biochemical properties and functions. Despite these impressive discoveries, there are far more unknown substrates yet to be discovered especially considering the avalanche of newly sequenced bacterial genomes and plasmids. With the purpose of facilitating statistical analysis and computational prediction of various types of substrate, this thesis aimed to develop a series of analytical and predictive toolkits based on machine learning with the intention to interlink them as an integrative platform and pipeline. Through providing seamless operations between laboratory-confirmed substrates, potential substrate prediction and their inter-relationship analysis, this streamlined tool suite is expected to provide insights into the known substrates and facilitate new substrate discoveries.

This thesis is organized as follows. Chapter 1 introduces the background of bacterial protein secretion systems and their secreted substrates, as well as the basic knowledge and current progress of the application of machine learning into substrate prediction. It also describes the challenges in analysing and predicting secreted substrates and emphasizes my contributions during the PhD study. Chapter 2 presents three methodologies and implemented predictors (Bastion3, Bastion4 and Bastion6) for the computational prediction of three well studied types of secreted substrates found in Gram-negative bacteria. Chapter 3 presents two combinable toolkits (BastionX and BastionHub) as an integrative system for comprehensive and systematic annotation, analysis and prediction of various types of secreted proteins. Chapter 4 presents two computational toolkits (POSSUM and DIFFUSER) to provide a streamlined and automatic feature generating service, so as to facilitate a general development of machine learning based predictors. Chapter 5 concludes the whole project

and discusses the future direction in developing more comprehensive and intelligent systems for use in “diagnosis” of bacterial capabilities in causing disease.

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Signature:

Print Name: Jiawei Wang

Date: 29/09/2019

Publications during enrolment

Publications included in the thesis

1. **Wang, J.** †, Yang, B.†, An, Y.†, Marquez-Lago, T., Leier, A., Wilksch, J., Hong, Q., Zhang, Y., Hayashida, M., Akutsu, T., Webb, G. I., Strugnell, R. A., Song, J.*, & Lithgow, T*. (2019). Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches. ***Brief Bioinform****, 20*(3), 931-951. doi:10.1093/bib/bbx164
2. **Wang, J.**, Yang, B., Leier, A., Marquez-Lago, T. T., Hayashida, M., Rocker, A., Zhang, Y., Akutsu, T., Chou, K. C., Strugnell, R. A., Song, J.*, & Lithgow, T.* (2018). Bastion6: a bioinformatics approach for accurate prediction of type VI secreted effectors. ***Bioinformatics****, 34*(15), 2546-2555. doi:10.1093/bioinformatics/bty155
3. **Wang, J.**, Li, J., Yang, B., Xie, R., Marquez-Lago, T. T., Leier, A., Hayashida, M., Akutsu, T., Zhang, Y., Chou, K. C., Selkrig, J.*, Zhou, T.*, Song, J.*, & Lithgow, T.* (2019). Bastion3: a two-layer ensemble predictor of type III secreted effectors. ***Bioinformatics****, 35*(12), 2017-2028. doi:10.1093/bioinformatics/bty914
4. **Wang, J.**, Xie, R., Li, J., Dai, W., Zhou, T., Akutsu, T., Webb, C., Stubenrauch, C., Zhang, Y., Song, J.*, & Lithgow, T.* BastionX: Systematic and accurate prediction of secreted substrates in Gram-negative bacteria within a distributed framework. *To be submitted*.
5. **Wang, J.**, Li, J., Xie, R., Hou, Y., Marquez-Lago, T., Leier, A., Zhou, T., Torres, V., Hay, I., Zhang, Y., Song, J.*, & Lithgow, T.* BastionHub: a universal platform for integrating and analyzing substrates secreted by Gram-negative bacteria. *To be submitted*.
6. **Wang, J.**†, Yang, B.†, Revote, J., Leier, A., Marquez-Lago, T. T., Webb, G., Song, J.*, Chou, K. C., & Lithgow, T.* (2017). POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. ***Bioinformatics****, 33*(17), 2756-2758. doi:10.1093/bioinformatics/btx302
7. **Wang, J.**, Xie, R., Li, J., Leier, A., Yang, B., Revote, J., Akutsu, T., Webb, G. I., Smith, A.I., Marquez-Lago, T., Zhang, Y., Lithgow, T.*, & Song, J.* DIFFUSER: A distributed framework for high-throughput generation of machine-learning features from DNA, RNA and protein sequences. *To be submitted*.

Publications not included in the thesis

1. Zhang, Y. †*, Xie, R. †, **Wang, J.** †*, Leier, A., Marquez-Lago, T. T., Akutsu, T., Webb, G. I., Chou, K. C., & Song, J.* (2018). Computational analysis and prediction of lysine malonylation sites by exploiting informative features in an integrative machine-learning framework. ***Brief Bioinform***. doi:10.1093/bib/bby079
2. Zhang, Y., Yu, S., Xie, R., Li, J., Leier, A., Marquez-Lago, T. T., Akutsu, T., Smith, A. I., Ge, Z., **Wang, J.***, Lithgow, T.*, & Song, J.* (2019). PeNGaRoo, a combined gradient boosting and ensemble learning framework for predicting non-classical secreted proteins. ***Bioinformatics***. doi:10.1093/bioinformatics/btz629
3. Xie, R.†, Li, J.*, **Wang, J.**†*, Dai, W., Leier, A., Marquez-Lago, T. T., Akutsu, T., Lithgow, T., Song, J.*, & Zhang, Y.* (2019). DeepVF: a deep learning-based hybrid framework for identifying virulence factors using the stacking strategy. ***Brief Bioinform****, Major revision*.
4. Grinter, R., Hay, I. D., Song, J., **Wang, J.**, Teng, D., Dhanesakaran, V., Wilksch, J. J., Davies, M. R., Littler, D., Beckham, S. A., Henderson, I. R., Strugnell, R. A., Dougan, G., & Lithgow, T.*

(2018). FusC, a member of the M16 protease family acquired by bacteria for iron piracy against plants. ***PLoS Biol**, 16*(8), e2006026. doi:10.1371/journal.pbio.2006026

5. Song, J., Wang, H., **Wang, J.**, Leier, A., Marquez-Lago, T., Yang, B., Zhang, Z., Akutsu, T., Webb, G. I.*, & Daly, R. J.* (2017). PhosphoPredict: A bioinformatics tool for prediction of human kinase-specific phosphorylation substrates and sites by integrating heterogeneous feature selection. ***Sci Rep**, 7*(1), 6862. doi:10.1038/s41598-017-07199-4
6. Zhao, Y., Zhang, X., Torres, V. V. L., Liu, H., Rocker, A., Zhang, Y., **Wang, J.**, Chen, L., Bi, W., Lin, J., Strugnell, R. A., Zhang, S., Lithgow, T.*, Zhou, T.*, & Cao, J.* (2019). An Outbreak of Carbapenem-Resistant and Hypervirulent Klebsiella pneumoniae in an Intensive Care Unit of a Major Teaching Hospital in Wenzhou, China. *Front Public Health, 7*, 229. doi:10.3389/fpubh.2019.00229

*† co-first authorship, * Corresponding authorship*

Thesis including published works declaration

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

This thesis includes 4 original papers published in peer reviewed journals and 3 unpublished publications. The core theme of the thesis is systematic and comprehensive analysis and prediction of substrates secreted by Gram-negative bacteria. The ideas, development and writing of all the papers in the thesis were the principal responsibility of myself, the student, working within the Department of Microbiology under the supervision of Prof. Trevor Lithgow and A/Prof. Jiangning Song.

The inclusion of co-authors reflects the fact that the work came from active collaboration between researchers and acknowledges input into team-based research.

In the case of chapters 3 and 4, my contribution to the unpublished work involved the following:

For the manuscript “BastionX: Systematic and accurate prediction of secreted substrates in Gram-negative bacteria within a distributed framework” in chapter 3, I was responsible for designing and implementing toolkit, designing experiment and analyzing data, and writing manuscript; Mr. Ruopeng Xie contributed to the experiment conduction and toolkit development; Ms. Jiahui Li contributed to data analysis; Mr. Wei Dai contributed to experiment conduction; Prof. Tieli Zhou, Prof. Tatsuya Akutsu, Dr. Chaille Webb, Dr. Christopher Stubenrauch and Prof. Yanju Zhang provided input to the manuscript; A/Prof. Jiangning Song and Prof. Trevor Lithgow supervised this work.

For the manuscript “BastionHub: a universal platform for integrating and analyzing substrates secreted by Gram-negative bacteria” in chapter 3, I was responsible for designing and implementing toolkit, analyzing data, and writing manuscript; Ms. Jiahui Li contributed to data analysis; Mr. Ruopeng Xie and Mr. Yi Hou contributed to toolkit development; A/Prof. Tatiana T. Marquez-Lago, Dr. André Leier, Prof. Tieli Zhou, Dr. Torres Van, Dr. Iain Hay and Prof. Yanju Zhang provided input to the manuscript; A/Prof. Jiangning Song and Prof. Trevor Lithgow supervised this work.

For the manuscript “DIFFUSER: A distributed framework for high-throughput generation of machine-learning features from DNA, RNA and protein sequences”, I was responsible for designing and implementing toolkit, analyzing data, and writing manuscript; Mr. Ruopeng Xie contributed to experiment conduction and toolkit development; Ms. Jiahui Li, Dr. André Leier, Prof. Tatsuya Akutsu, Prof. Geoffrey I. Webb, Prof. A. Ian Smith, A/Prof. Tatiana T. Marquez-Lago and Prof. Yanju Zhang provided input to the manuscript; Mr. Bingjiao Yang and Mr. Jerico Revote provided technical support; Prof. Trevor Lithgow and A/Prof. Jiangning Song supervised this work.

In the case of chapters 2 and 4 my contribution to the published work involved the following:

Thesis Chapter	Publication Title	Status	Nature and % of student contribution	Co-author name(s) Nature and % of Co-author's contribution*	Monash student Y/N*
2	Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches	Published	65%. Designing and conducting experiments, analysing data, implementing toolkit and writing manuscript.	1) Yang, B., experiment input and data analysis, 15% 2) An, Y., data collection and experiment input, 6% 3) Marquez-Lago, T., manuscript input, 2% 4) Leier, A.,manuscript input, 2%	N N N N N N

				5) Wilksch, J., manuscript input, 2% 6) Hong, Q., technical support, 2% 7) Zhang, Y., manuscript input, 1% 8) Hayashida, M., technical support, 2% 9) Akutsu, T., manuscript input, 1% 10) Webb, G. I., manuscript input, 1% 11) Strugnell, R. A., manuscript input, 1% 12) Song, J., supervision 13) Lithgow, T., supervision	N N N N N N N
2	Bastion6: a bioinformatics approach for accurate prediction of type VI secreted effectors	Published	80%. Collecting data, Designing and conducting experiments, analysing data, constructing toolkit and writing manuscript.	1) Yang, B., experiment, toolkit and manuscript input, 10% 2) Leier, A., manuscript input, 2% 3) Marquez-Lago, T. T., manuscript input, 2% 4) Hayashida, M., technical support, 1% 5) Rocker, A., technical support, 1% 6) Zhang, Y., manuscript input, 1% 7) Akutsu, T., manuscript input, 1% 8) Chou, K. C., manuscript input, 1% 9) Strugnell, R. A., manuscript input, 1% 10) Song, J., supervision 11) Lithgow, T, supervision	N N N N N N N N N N N
2	Bastion3: a two-layer ensemble predictor of type III secreted effectors	Published	80%. Designing and conducting experiments, analysing data, and writing manuscript.	1) Li, J., data collection and data analysis, 5% 2) Yang, B., experiment input and manuscript input, 4% 3) Xie, R., experiment and toolkit input, 4% 4) Marquez-Lago, T. T., manuscript input, 2% 5) Leier, A., Hayashida, M., manuscript input, 2% 6) Akutsu, T., manuscript input, 1% 7) Zhang, Y., manuscript input, 1% 8) Chou, K. C., manuscript input, 1% 9) Selkrig, J., supervision 10) Zhou, T., supervision 11) Song, J., supervision 12) Lithgow, T., supervision	N N N N N N N N N N N N
4	POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles	Published	75%. Designing and implementing toolkit, analysing data, and writing manuscript.	1) Yang, B., experiment and toolkit input, 18% 2) Revote, J., technical support, 2% 3) Leier, A., manuscript input, 1% 4) Marquez-Lago, T. T., manuscript input, 1% 5) Webb, G., manuscript input, 1% 6) Song, J., supervision 7) Chou, K. C., manuscript input, 2% 8) Lithgow, T., supervision	N Y N N N N N N

I have not renumbered sections of submitted or published papers in order to generate a consistent presentation within the thesis.

Student signature: **Date:** 30/10/2019

The undersigned hereby certify that the above declaration correctly reflects the nature and extent of the student's and co-authors' contributions to this work. In instances where I am not the responsible author I have consulted with the responsible author to agree on the respective contributions of the authors.

Main Supervisor signature: **Date:** 30/10/2019

Acknowledgments

I would like to express my great appreciation and sincere gratitude to my fantastic supervisor Prof. Trevor Lithgow. It has been a huge honour and privilege to work with Trevor during my PhD study, from whom I have learned far more than expected. The continuous trust, support and freedom you offered motivate me to think freely and positively, to question and judge with scepticism, and most importantly as a result, to enjoy my research and life. You are an amazing role model that stimulates me to be a better person both in work and daily life: Trevor you always show dignity, patience and respect to everyone around, you take your time to help students as a teacher, and share your passion, professionalism and rigorous attitude to scientific research. Because of you and what you have done for me, it has been a happy, fruitful but short time to walk through my PhD journey. Every moment with you will always stay in my memory, continuously drive me to learn and grow, and finally, I hope, shape me as a person that makes you proud.

To A/Prof. Jiangning Song, my associate supervisor, your unlimited enthusiasm, constant efforts and strong concentration towards scientific research deeply impress me, and remind me of the future direction that I should carefully think about and plan. The door to your office is always open to me even at weekends whenever I have ideas or questions in research. You are always ready and passionate to discuss scientific ideas and share your experiences with me, from which I have benefitted greatly during my PhD study and thereafter. Without your full engagement and unreserved dedication, my PhD journey would not have been that meaningful and magnificent: life is always with twists and turns, through all ups and downs as well as the untrodden paths, our strong belief, positive mindset and tremendous efforts will ultimately tackle the roadmap and take us to the shore of new world.

To my PhD review committee: A/Prof. Vijay Dhanasekaran, A/Prof. Traude Beilharz, and A/Prof. David Powell, I am grateful to have you as my panel members during this PhD journey. Your critical comments and suggestions continuously improve the quality of my research and the presentation to audience with biological background, remind me of potential drawbacks that need to be tackled, and in return this further inspires me to pursue better solutions for my research targets. Your considerate arrangement, support and encouragement have made it possible to achieve an accelerated PhD process. Your critical attitude and strict criterion on scientific research have truly motivated me to achieve better with a higher standard, and will definitely have a long-term influence on my research career.

To all Lithgow lab members, I am lucky enough to call you colleagues, and the relaxed academic atmosphere benefits me to form a simple style of learning and working without any additional concerns and worries. Your comments on my research work cover my lack of biological knowledge, and your feedbacks on my weekly presentation and milestone seminars substantially improved my expression skills towards different target audiences. Special thanks go to Dr. Chaille Webb, Dr. Christopher Stubenrauch, and Dr. Andrea Rocker who have revised the initial draft of this thesis and your rapid response on my request is something I never expected. Special thanks to Dr. Iain Hay, Ms. Rebecca Bamert, Ms. Kher-Shing Tan, Dr. Rhys Grinter, Dr. Rhys Dunstan, Dr. Matthew Belousoff, Dr. Von Torres, Dr. Grishma Vadlamani, Mr. Eric Mandela, Ms. Natalia Rosas Bastidas, Ms. Manasa Bharathwaj and Ms. Yajie Zhao. I appreciate your emotional and academic encouragement and support along with my PhD period.

To all my collaborators, working together with you has made everything simple, enjoyable and effective. Specially to Prof. Tatsuya Akutsu, Prof. Geoffrey I. Webb, Prof. Kuo-Chen Chou, Prof. Richard A. Strugnell, Prof. Tieli Zhou, Dr. Tatiana Marquez-Lago, Dr. Andre Leier, Dr. Morihiro Hayashida, Dr. Jonathan Wilksch, Dr. Joel Selkrig, Dr. Zongyuan Ge, Mr. Jerico Revote, Mr. Qingyang Hong and Ms. Yi An, your involvement and contribution along with my PhD period have significantly improved the quality of my research work and publications.

More personal thanks go to Prof. Yanju Zhang, who offered me the opportunity to work with and grow with her lab. This unexpected and enriched experience on supervising, teaching and getting along with students has developed my skills and will aid in the transformation as a research leader when I have my own lab one day. To Mr. Bingjiao Yang, Ms. Jiahui Li, Mr. Ruopeng Xie and Ms. Sha Yu, you mean much more to me than what you have imagined. Your companionship, friendship and assistance both in study and daily life have freed me of tedious details, and focused me on a big picture to plan the future. To Mr. Wei Dai and Mr. Yi Hou, your involvement and contribution have pushed forward our projects and enabled them to stay on the right track.

To my science neighbours: Dr. Yan Jiang, Dr. Meiling Gao, Dr Fengwang Li, Ms. Bella Wang and Mr. Jiayuan Huang, I will never forget our good times together, from the office breaks, soccer matches, to outdoor trips around Australia. Special thank you to Dr Mingling Han, your amiable attitude and great patience impressed me so much when I bothered you many time for tough transactional issues.

Finally, I would like to dedicate this step in my academic career to my family members. To my parents and my brother, you have provided unconditional and unreserved support on my education and a warm harbor for me to find the way back home whenever I need. To my wife Ms. Yanhua Chen, meeting and marrying you has made me to be the luckiest and happiest person in the world. Your transposition thinking, rational thought, cognitive capacity and communication ability have led to a more straightforward and concise interaction between us. This further inspires and enables me to act more decent, gracious and grateful in the grind of daily routine. If a spousal relationship is supposed to make each other a better person, you make it. To our pet Choco, you are such a distinctive and mild-mannered dog – I know you don't think you are a dog. You always sit quietly beside me and silently accompany me no matter if I am in serious work or state of relaxation and enjoyment. You always welcome me so warmly on my return home as if we haven't seen each other for long time, even if I just left a moment ago.

"O ever youthful, O ever weeping", Jack Kerouac writes in The Dharm Bums. We may not stay young forever, but be always young at heart. Keep learning, keep growing and keep living with all the richness your precious life deserves - as to memorize my PhD period.

My PhD study was financially supported by the Monash Graduate Scholarship (MGS), Monash International Postgraduate Research Scholarship (MIPRS) and Faculty Postgraduate Excellence Award.

Table of contents

List of Figures

Abbreviations

T1SS	type I secretion system
T2SS	type II secretion system
T3SS	type III secretion system
T4SS	type IV secretion system
T5SS	type V secretion system
T6SS	type VI secretion system
T1SE	type I secreted effector
T2SE	type II secreted effector
T3SE	type III secreted effector
T4SE	type IV secreted effector
T6SE	type VI secreted effector
OM	outer membrane
IM	inner membrane
ATP	adenosine triphosphate
RTX	repeat-in-toxin
PSSM	position-specific scoring matrix
mRMR	minimum redundancy maximum relevance
CFS	correlation feature selection
GR	gain ratio
ML	machine learning
SVM	support vector machine
NB	naïve Bayes
BN	Bayesian network
RF	random forest
ANN	artificial neural network
MLP	multilayer perceptron
HMM	hidden Markov model
KNN	k-nearest neighbor

LR	logistic regression
GBDT	gradient boosting decision tree
XGBoost	eXtreme gradient boosting
LightGBM	light gradient boosting machine
CV	cross-validation
LOOCV	leave-one-out CV test
TPR	true positive rate
FPR	false positive rate
TP	true positive
TN	true negative
FP	false positive
FN	false negative
SN	sensitivity
SP	specificity
ACC	accuracy
MCC	Matthews correlation coefficient
ROC	receiver operating characteristic
AUC	area under the curve
API	application programming interface

CHAPTER 1: Introduction

1.1 Secreted substrates in Gram-negative bacteria

Gram-negative bacteria are responsible for many serious infections worldwide, including emerging hospital-acquired infections, healthcare-associated infections and neglected tropical diseases. Reported by the US Centers for Disease Control and Prevention in a list of bacteria urgently requiring for new antibiotics (Solomon & Oliver, 2014) are notable examples including *Escherichia coli*, which causes most urinary tract infections, *Pseudomonas aeruginosa* that causes bloodstream infections and pneumonia; *Neisseria gonorrhoeae*, causative agent of the sexually transmitted disease gonorrhea; and *Klebsiella pneumoniae* which is responsible for several types of healthcare-associated infections including bloodstream infections, pneumonia and urinary tract infections. All of these examples are evolving to be extremely resistant to antibiotics (Pendleton et al., 2013; Santajit & Indrawattana, 2016). Understanding the molecular mechanisms behind Gram-negative bacterial infections and therefore finding efficient diagnostics and treatment strategies is crucial in preventing humans (as well as animals and plants) from the risk of suffering such diseases.

As one of the virulence "weapons" of Gram-negative bacteria, secretion systems are used to secrete diverse substrates into the surrounding environment, eukaryotic host cells, or into neighboring bacterial cells (Wandersman, 2013). Bacterial secreted substrates are known to be involved in a series of complex macromolecular interactions with host proteins, which in turn contributes toward the pathology of bacterial infections (Mattoo et al., 2007; Mukhtar et al., 2011; Stavrinides et al., 2008). Identifying these substrates, clarifying their secretion pathways and further uncovering their functions will provide insights into Gram-negative bacterial infection mechanisms.

Among the nine distinct secretion systems, the first six major pathways have been best-studied in Gram-negative bacteria (Costa et al., 2015). These are referred to as the type I to type VI secretion systems, abbreviated as T1SS, T2SS, T3SS, T4SS, T5SS and T6SS (Fig. 1.1). These protein secretion systems are complicated molecular machines, composed of multiple protein parts that work together to drive the secretion of their protein substrates across the bacterial membranes. The T1SS (Welch et al., 1981), T2SS (d'Enfert et al., 1987), T3SS (Galan & Curtiss, 1989), T4SS (Kuldau et al., 1990) and T6SS (Pukatzki et al., 2006)

span both the outer membrane (OM) and the inner membrane (IM), and use ATP hydrolysis in order to provide the energy for protein secretion. The ATP hydrolysis is catalyzed by distinct ATPases in each case. The T1SS is a machine built from four protein subunits, where one is a ATPase of the ABC family of proteins (Morgan et al., 2017). The T3SS has a ATPase that is structurally similar to the β-catalytic subunit of the F1 ATPase (Burkinshaw & Strynadka, 2014), and the T4SS requires ATP hydrolysis but it is not yet clear which subunit serves as the ATPase (Ripoll-Rozada et al., 2013), while the T6SS uses the ClpB ATPase to drive protein secretion (Barbosa & Lery, 2019; Brodmann et al., 2017). Since they use a two-step process, both the T2SS and the T5SS depend on the SecA ATPase to drive protein secretion across the IM, with the T2SS using an additional ATPase to drive a piston that forces the substrate from the periplasm across the OM. In all cases the substrates are exported out across the OM, either directly into the external environment or specifically into human (or other) host tissues, or into neighbouring bacteria.

In something of an exception to the general rule is the T5SS, which sits discretely in the OM, uses the potential energy in protein folding to drive secretion of the substrate protein across the OM, and consists of six distinct sub-types (T5SSa-T5SSf) based on the characteristics of the protein pore in the OM. Some of these T5SS sub-types are generally called "autotransporters", because the secreted substrate and the protein secretion pore are fused in a single polypeptide (Fan et al., 2016; Heinz et al., 2016; Henderson et al., 2004; Meuskens et al., 2019; Nicolay et al., 2015). All of these pores, of the T5SSa-T5SSf sub-types, are assembled in the OM by translocons of the Omp85 family (Heinz & Lithgow, 2014).

Each of the six main secretion systems has distinct characteristics in their secretion pathways and mechanisms (Fig. 1.1), and these characteristics impact on how substrates are selected for translocation. T1SS, T3SS, T4SS and T6SS export their substrates across the bacterial envelope via a one-step secretion mechanism: a recognition event in the cytoplasm chooses which protein substrates will be selected (from among the thousands of proteins present in the cytoplasm) to be translocated from the bacterial cytoplasm into the target cells or the extracellular milieu. In contrast, T2SS and T5SS export their substrates via a two-step secretion mechanism: substrates are first translocated into the periplasmic space in a similar fashion to all other periplasmic and cell envelope proteins (via the Sec or Tat systems), and (i) the T2SS then selects only one or very few proteins as substrates to be secreted to the extracellular milieu (Costa et al., 2015; Hay et al., 2018; Hay et al., 2017), while (ii) the T5SS functions simply by having the cell's Omp85 translocon integrate it into the OM. Since the

mechanism of T5SS secretion and the discovery of T5SS proteins is well understood, this thesis focuses on the T3SS, T4SS and T6SS, with future expectations of the same informatics strategies being applied to the T1SS and T2SS. The question of how those (non-T5SS) substrates are targeted and selected from thousands of common proteins for secretion, as well as their exact delivery procedure into the host cells or outside environments, is far from being completely understood.

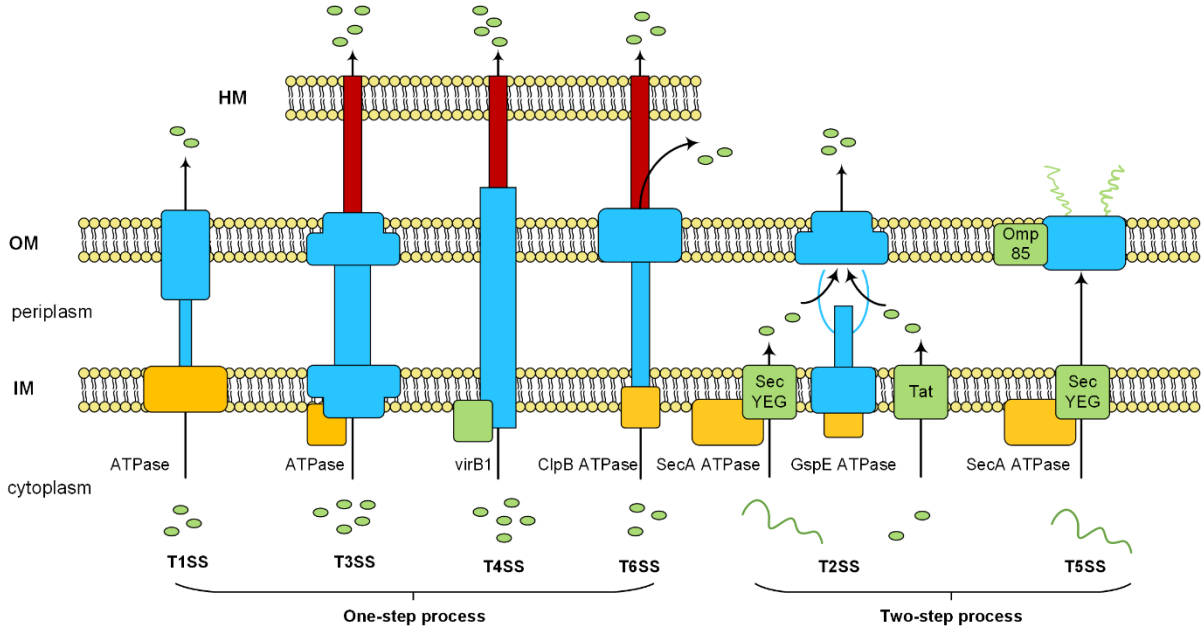


Fig 1.1. Summary of six secretion systems in Gram-negative bacteria. Shown coloured blue is the machinery of the T1SS, T2SS, T3SS, T4SS and T6SS, which spans both the inner membrane (IM) and the outer membrane (OM), and the T5SS which sits only in the OM of Gram-negative bacteria. Shown in red are additional components of these protein secretion systems that serve as adaptors to direct substrates into host membranes (HM). As detailed in the text, ATPases (yellow) drive protein secretion, and shown in green are the additional general "translocons" needed to assist protein transport, but which are not *per se* part of the protein secretion system.

In addition to the diversity of secretion mechanisms, the substrates themselves play different roles with distinct functions: type I and type II proteins are usually enzymes, often hydrolases, that facilitate access and uptake of nutrients from the environment, while the majority of type III, IV and VI substrates are also referred to as "effectors" whose functions are thereby implied to be a direct imitator of a host-cell functionality. In contrast, the zinc-binding protein substrate YezP (Wang et al., 2015), the manganese-binding protein TseM (Si et al., 2017), and the iron uptake assistance protein TseF (Lin et al., 2017), are all

experimentally validated type VI substrates that are directly secreted into the external environment to combat multiple stresses and host immunity, through competitive control of ion absorption.

Due to the diverse nature of substrates, the numbers of experimentally validated substrates vary across bacterial species, with respect to different hosts and according to various survival strategies (Burstein et al., 2016; Pearson et al., 2015; Zou et al., 2013). With the large repertoire of secretion systems available to bacteria, many species are capable of secreting dozens of different substrates (Burstein et al., 2009; Burstein et al., 2015; Burstein et al., 2016; Tobe et al., 2006) through one or more secretion systems (Chen et al., 2011; Dean & Kenny, 2009; Folders et al., 2001). Examples include *Legionella* that can secrete more than 100 substrate proteins through its T4SS (An et al., 2017; Jeong et al., 2017; Qiu & Luo, 2013; Schroeder, 2017) and *E. coli* where some strains have the capability to secrete various proteins through as many as five different secretion systems (T1SS, T2SS, T3SS, T4SS and T6SS) (An et al., 2017; Christie, 2016; Dean & Kenny, 2009; Navarro-Garcia et al., 2019; Patrick et al., 2010; Schwarz et al., 2012; Slater et al., 2018; Tobe et al., 2006). Accordingly, the experimental validation protocols of those substrates completely vary from one type to another, and even largely differ within the same secretion type, which makes it particularly difficult to predict the identity of new substrates that have different sequence signatures compared to known substrates.

1.2 Characteristics of the various secretion system substrates

Based on the known substrates and the current knowledge of their biochemical properties and secretion mechanisms, bioinformatics has been used to explore the characteristics of their protein sequences. The broad aim has been to locate possible residue repeats, patterns and motifs, and in turn guide the computational identification of new substrates.

T1SS mainly secretes substrates that are members of a protein family, termed repeat-in-toxin (RTX) proteins (Kanonenberg et al., 2013). RTX proteins usually possess distinctive glycine-rich repeats (GGxGxDxxx, where x is any amino acid) to specifically bind calcium in their C-terminus (Welch, 2001). A group of very large toxins, the Multifunctional Autoprocessing RTX (MARTX) (Satchell, 2007) represents a division of the RTX family, differing from other RTX proteins in terms of some structural elements and *rtx* gene cluster organization (Linhartova et al., 2010). Unlike other RTX proteins, MARTX proteins harbor an 18-residue-long motif x(V/I)xxGxxNx(V/I)xxGDGxDx in their C-terminus and an N-terminal motif of

either 20 residues [GxxG(N/D)(L/I)(T/S)FxGAG(A/G)xNx(L/I)x(RH)] or 19 residues [T(K/H)VGDGx(S/T)VAVMxGxAN(I/V)x] (Linhartova et al., 2010). Using previously known characteristics and motifs, different methods including pattern search, Hidden Markov Model (HMM) search and RPS-BLAST search have been applied to identify new RTX proteins from hundreds of Gram-negative bacterial genome sequences (Linhartova et al., 2010). By way of contrast, the proteins secreted by the T2SS do not show any conserved sequence-based motifs, leading to suggestions that there may be "motifs" encoded in structural features of the folded proteins destined for secretion (Dalbey & Kuhn, 2012).

Both type III and IV substrates are directly injected into the host cell cytoplasm by their secretion systems. Like the T2SS, there are no obviously identifiable patterns in the sequences of the type III and IV substrates. However, deletion analyses of some model substrates have suggested a requirement for elements situated at their N-terminus and C-terminus, respectively (McDermott et al., 2011). For example, the delivery of these substrates into host cells by the Dot/Icm T4SS in Legionella requires a C-terminal translocation signal (Nagai et al., 2005). T3SS examples too include intrinsically disordered sequence features (Buchko et al., 2010) as well as chaperone-binding domain where structural motifs come together as three-dimensional signals (Birtalan et al., 2002; Ernst et al., 2018; Lilic et al., 2006), in the first 30/100 amino acids at the N-terminus. As most of the key components of the type III secretion system are conserved across species (Pallen et al., 2005), its targeting mechanisms may also be conserved (McDermott et al., 2011). Two overlapping N-terminal domains in type III substrates are thought to be responsible to mediate their secretion: (1) Residues from 1 to 25 form a highly variable secretion signal (Michiels & Cornelis, 1991), and is sometimes highly tolerant of mutations (Russmann et al., 2002). (2) Residues from 15 to 30 (or 15 to 100) in many substrates form a chaperone domain that enables a cognate chaperone protein to bind them (Lee & Galan, 2004). Removal of either domains prevents substrate recognition by the type III secretion system and in turn prevents their subsequent secretion (Lloyd et al., 2001; Michiels & Cornelis, 1991; Sory et al., 1995). By contrast, the secretion signals of type IV substrates remain difficult to clarify, as they have two distinct secretion mechanisms, i.e. T4a and T4b (Cascales & Christie, 2003). While some T4a substrates in certain species harbor conserved residues (Vergunst et al., 2005), their mutagenesis doesn't affect the secretion of these substrates, indicating the presence of additional recognition domains (Hohlfeld et al., 2006). This also applies to the T4b substrates whose secretion mechanisms are more sophisticated and requires a possible

combination of multiple physical and structural features to provide necessary secretion signatures (McDermott et al., 2011).

Type VI substrates are either exported into the surrounding environment to promote nutrient uptake and survival (Lin et al., 2017; Si et al., 2017; Wang et al., 2015), or directly injected into eukaryotic host cells or competing bacterial cells (Cianfanelli et al., 2016; Durand et al., 2014; Russell et al., 2014). To date, no commonly agreed upon domains or motifs have been observed across type VI substrates, but there are some sequence characteristics that have been implicated in structurally or functionally similar substrates. For example, some VgrG and Hcp proteins belonging to type VI substrates often harbor a conserved C-terminal domain (Cianfanelli et al., 2016; Jamet & Nassif, 2015; Ma et al., 2017a; Pukatzki et al., 2009), leading to speculation that a protein may be a type VI substrate if it is appended with an additional C-terminal extension region (Lien & Lai, 2017). Type VI substrates also possess domains or motifs related to their biochemical or biological activity: examples being from a group of phospholipase substrates with a conserved motif GxSxG, HxKxxxxD (Russell et al., 2013) or GxSxG (Flaugnatti et al., 2016). Other domains or motifs in type VI substrates responsible for their secretion include Rhs/YD repeats (Koskiniemi et al., 2013; Ma et al., 2017b; Murdoch et al., 2011; Whitney et al., 2014), PAAR motifs (Ma et al., 2014; Rigard et al., 2016; Whitney et al., 2014), TTR motifs (Flaugnatti et al., 2016; Shneider et al., 2013) and MIX motifs (Salomon, 2016; Salomon et al., 2014; Salomon et al., 2015).

These observed patterns, domains and motifs have been used to identify new substrates, and each of them has accordingly succeeded in identifying at least one other substrates that was then experimentally validated. However, these bioinformatics analyses are highly limited to, and dependent on, the existing knowledge of biochemical features and transport mechanisms of the substrates. They differ case by case in terms of the type, bacterial species or functions of the substrates. In addition, some experimentally determined motifs designate protein families and/or protein function, so that they are not exclusively related with secreted substrates. As such, these conserved motifs will potentially lead to a high false positive retrieval result. Identifying proteins that are similar in sequence provides no real benefit to determining the extent of variation in substrates – it introduces a bias instead where there is a stronger proportion of such proteins.

1.3 Machine learning in biological sequence-based prediction

With the advances in high-throughput sequencing technologies, there has been a rapid increase in the discovery and accumulation of biological sequences. To deal with the rapid growth of these datasets and interpreting such big data, machine learning algorithms have been increasingly applied to (i) gain insights into complex biological systems and (ii) elucidate the mechanisms of diseases, therefore providing an indispensable and integral 'ingredient' of cutting-edge, cross-disciplinary biological and biomedical research (Camacho et al., 2018; Larranaga et al., 2006; Libbrecht & Noble, 2015). Machine learning describes a set of computational approaches "trained" on a given dataset to learn the relationships between the samples and is thus capable of mining patterns that could be used to predict new samples. Machine learning algorithms accept features in the format of matrix or vectors, which in the case of biological sequences can be sequence patterns that are meaningful in their own right, or higher-order patterns reflective of protein structural features. Formulating the sequence- based prediction as a classification problem (e.g. classifying proteins into secreted substrates versus none secreted substrates in this thesis), the following major stages are generally involved when applying machine learning techniques: dataset collection and curation, sample representation, model construction, model validation and service development (Fig. 1.2).

1.3.1 Dataset collection and curation

To develop a machine-learning predictor to address a specific task, the first step is to construct a reliable benchmark dataset that can represent the diversity of the data to the best extent possible. In most cases, the benchmark dataset will consist of a training dataset and an independent test dataset. The training dataset is used for model training, while the independent test dataset is used for validating the trained model and assessing its prediction performance on unseen data samples. Ideally, a subset or a few case study samples, which are regarded as the most representative or recently experimentally validated samples with no overlap with those in the training dataset should be collected and used to further assess the quality and performance of the developed predictor.

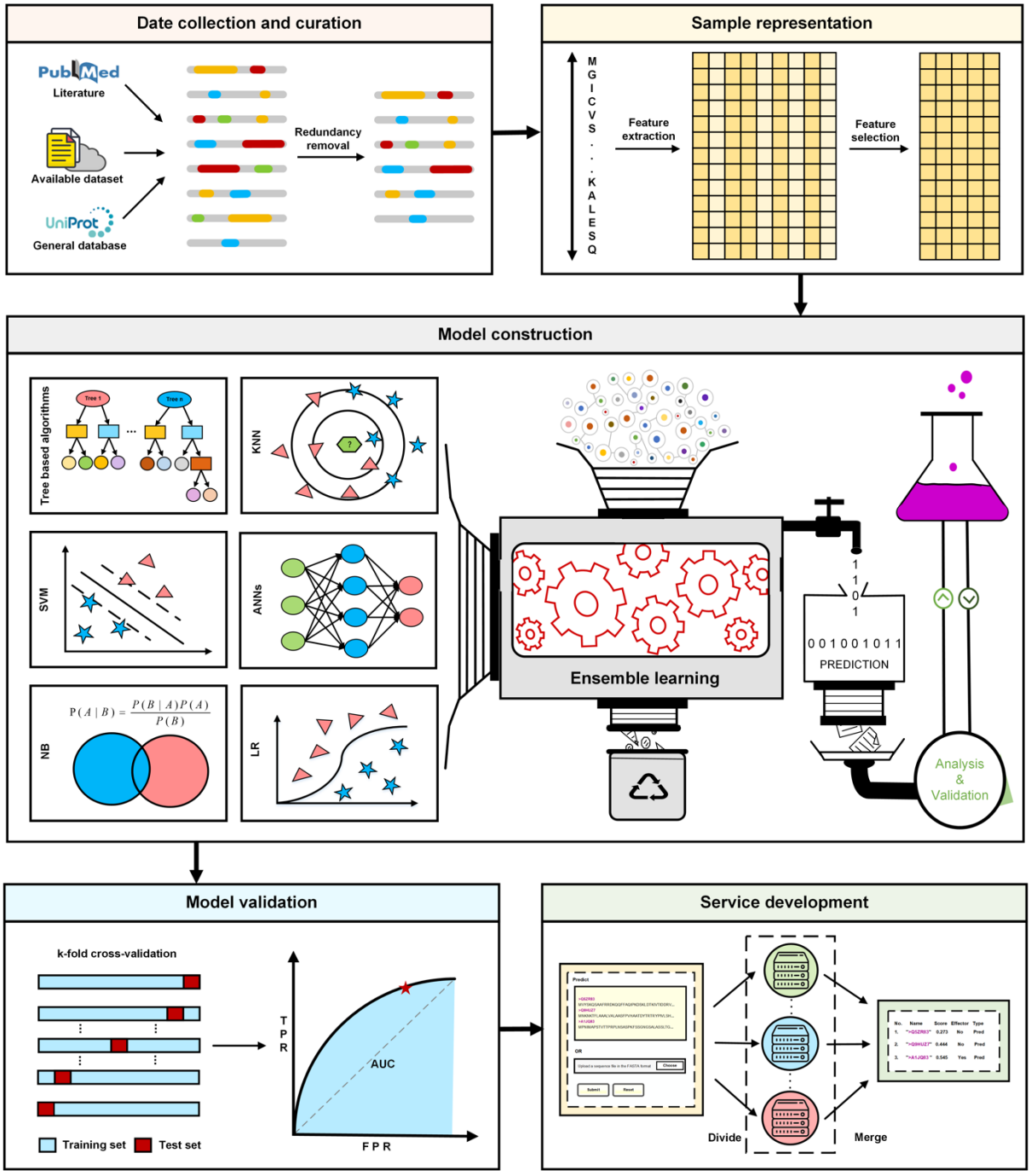


Fig 1.2. Illustration of a common workflow of application of machine learning techniques in sequence based modelling and analysis.

After an extensive and thorough search and collection from public databases and literature, one can obtain an initial dataset. A subsequent step is to remove the sequence homology of the dataset by clustering all sequences at a given threshold of sequence identity. This is done in order to remove sequence redundancy and avoid biased model training that might

otherwise be introduced (Chou, 2011). Removing sequence redundancy using a lower threshold will generally lead to more rigorous benchmark datasets and result in more reliable and less over-fitted models (Chou, 2011). Experimentally validated substrates are typically ones that are highly homologous because they are the most obvious choices to test for wet lab scientists. However, as the numbers of experimentally validated biological samples are often not very large, the selection of a sequence identity threshold should be exercised with caution, such that an appropriate trade-off between the sequence homology of the dataset and its size is achieved. Several software tools, including CD-HIT (Huang et al., 2010) and UCLUST (Edgar, 2010), can be used to cluster and remove the sequence homology of a dataset.

1.3.2 Sample representation

Usually, machine-learning algorithms are not applied to raw biological sequence data as input but instead work with features that are provided in the form of vectors (Chou, 2011). As such, effectively representing a biological sequence (also known as feature encoding) is key to the success of constructing accurate predictors (Chou, 2011; Wang et al., 2019b; Zhang et al., 2018). Existing feature extraction methods can be categorized into five major groups, including sequence-based features, physicochemical property-based features, position-specific scoring matrix (PSSM) profile-based features, predicted structural features and other profile-based features. Sequence-based features are primarily based on the statistical information pertinent to the frequency or compositions of sequence elements within the biological sequences. Physicochemical property-based features are primarily associated with the descriptions and the encoding of physicochemical properties of biological sequences. PSSM profile-based features are specific for protein sequences and are derived from PSSM profiles (generated by performing PSI-BLAST search against a reference genome dataset) through different types of matrix transformations (Wang et al., 2017). Predicted structural features are extracted based on predicted outputs, primarily generated by third-party software packages that take protein and RNA sequences as input. Other types of profile-based features are generally extracted based on various types of sequence-derived profiles generated by other toolkits. These profiles, to name a few, are sequence conservation score (Glaser et al., 2005), enriched gene ontology terms (Chou & Cai, 2004), and protein functional domains (Chou & Cai, 2004).

The original features extracted from biological sequence data are often high-dimensional and contain potentially redundant information. This will introduce noise, slow down or bias the model training process, and typically result in reduced model performance (Awada et al., 2012; Saeys et al., 2007). Accordingly, feature selection techniques play an important role on 3-fold objectives (Guyon & Elisseeff, 2002): by enabling faster model construction, improving the prediction performance and generalization of the models (Bermingham et al., 2015), and providing more simplified modes for easier interpretation (James et al., 2013). Commonly used feature selection methods include the Gain ratio based on information theory (Shannon, 1948), the Minimum Redundancy Maximum Relevance (mRMR) (Peng et al., 2005) based on mutual information, and the Correlation Feature Selection (CFS) based on the correlate relationship of features (Hall, 1999).

1.3.3 Model construction

Once features are extracted and further selected, machine learning algorithms can be applied to train and construct models. Currently, a wide range of machine learning algorithms have been successfully developed and applied in the fields of bioinformatics and computational biology (Angermueller et al., 2016; Larranaga et al., 2006; Min et al., 2017). These ranges from classical machine learning methods, such as support vector machine (SVM) (Cortes & Vapnik, 1995), Naïve Bayesian (NB) (Friedman et al., 1997), Bayesian networks (BNs) (Jensen, 1996), random forest (RF) (Breiman, 2001), K-nearest neighbor (KNN), logistic regression (LR) (Freedman, 2009), Gradient Boosting Decision Tree (GBDT) (Friedman, 2001; Friedman, 2001), extreme gradient boosting (XGBoost) (Chen & Guestrin, 2016) and Light Gradient Boosting Machine (LightGBM) (Ke et al., 2017), to emerging deep learning algorithms, such as convolutional neural network (CNN) (Krizhevsky et al., 2017), recurrent neural network (RNN) (Giles et al., 1994), long short term memory network (LSTM) (Hochreiter & Schmidhuber, 1997), generative adversarial network (GAN) (I et al., 2014), reinforcement learning (Bartlett, 2002) and bidirectional encoder representations from transformers (BERT) (Devlin et al., 2018).

Usually, different features represent different characteristics of biological sequences, and can capture distinct sequence patterns from various perspectives, which therefore differ in their data distributions (Chen & Jeong, 2009). Incorporating such knowledge could further improve the prediction performance, as compared to models that have been trained using a single feature only (An et al., 2018; Chen & Jeong, 2009; Zou. et al., 2013). However,

compared to the models trained by simply combining these features, the ensemble learning strategies have been demonstrated to significantly improve the model performance (Chen et al., 2017; Chen & Jeong, 2009; Wan et al., 2017; Zhang et al., 2018; Zou et al., 2015). Commonly used ensemble learning strategies when integrating multiple single models include simple voting (Burstein et al., 2009; Sen et al., 2019; Zou et al., 2013), weighted/unweighted averaging (Zhang et al., 2018; Zhang et al., 2019), and stacking (Garg & Gupta, 2008; Wolpert, 1992; Xiong et al., 2018).

1.3.4 Model performance assessment

Upon construction of a model, its performance should be assessed objectively and rigorously. To this end, benchmarking experiments should be properly undertaken, including (1) *k*-fold cross-validation (CV) test (typical values for *k* are 5 or 10); (2) leave-one-out CV test (LOOCV), and (3) independent test. *k*-fold CV randomly partitions the benchmark data set into *k* equal-sized subsets, and repeats the tests five times. For each CV test, one subset is selected as the testing data, while the four remaining subsets are used to train the classifier. In this way, *k*-fold CV uses each subset once for testing and *k* times for training. The *k* results generated from these tests are further averaged as the overall performance of the model. LOOCV can be regarded as a special case of the *k*-fold CV with *k*=*n*, where *n* is the size of the benchmark data set. Similarly, LOOCV selects one sample out as the testing data, and use the remaining samples to trained the classifier. Accordingly, this process will be repeated *n* times, and the results will be then averaged as the overall performance of the model. Independent test further assesses a model using the independent dataset separately, without any overlap with the model's training data. The independent test represents a seperate and rigorous performance validation conducted based on the independent dataset that is usually collected independently and curated without any overlap with the training dataset (Larranaga et al., 2006).

To comprehensively and quantitatively assess a model's performance to solve a binary classification problem, a number of metrics are commonly used in the field of computational biology and bioinformatics, including Precision (PRE), Sensitivity (SN), Specificity (SP), Accuracy (ACC), F-value and Matthew's correlation coefficient (MCC) (Matthews, 1975). These metrics are defined as:

$$PRE = \frac{TP}{TP + FP}, 0 \leq SN \leq 1$$

$$SN = \frac{TP}{TP + FN}, 0 \leq SN \leq 1$$

$$SP = \frac{TN}{FP + TN}, 0 \leq SP \leq 1$$

$$ACC = \frac{TP + TN}{TP + FN + FP + TN}, 0 \leq ACC \leq 1$$

$$F - value = 2 \times \frac{TP}{2TP + FP + FN}, 0 \leq F - value \leq 1$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}, -1 \leq MCC \leq 1$$

where TP, TN, FP and FN represent the numbers of true positives, true negatives, false positives and false negatives, respectively. Additionally, in binary classification, the receiver operating characteristic (ROC) curve (Fawcett, 2006), a graphical plot of the true positive rate (TPR) versus the false positive rate (FPR) at various threshold values, are commonly used to visualize a model's diagnostic ability as its predictive threshold varies. The area under the curve (AUC) value of a ROC plot can then be calculated to quantify the model's predictive performance.

1.3.5 Public service development

After a model is constructed and evaluated, it is common to implement it in form of a web server, web service API or standalone toolkit to provide public access to its interested users. A user-friendly and easy-to-use web server can facilitate the process of using predictive models without going through complicated algorithmic details. A web service API can be helpful for command line users that wish to automatically retrieve results through scripts. However, as the increasing scale and complexity of biological sequence data often necessitates high throughput demands, distributed server design should be taken into consideration when constructing these servers. Alternatively, a standalone toolkit could largely reduce the web server's load, by allowing local execution on user's individual computing facilities. Both web service API (application program interface) and command line based standalone toolkit enables users to integrate a developed predictor into their own downstream or pipeline.

1.4 Progress and challenges in machine learning based secreted substrate analysis and prediction

1.4.1 Progress in secreted substrate predictors

In light of the biological significance of bacterial substrates, a considerable number of computational approaches have been developed to predict different types of substrates (An et al., 2018; McDermott et al., 2011; Zeng & Zou, 2017). Since machine learning was first applied in type III substrate prediction (Arnold et al., 2009; Samudrala et al., 2009), many such methods have been developed to accurately predict type I (Luo et al., 2015), type III (Arnold et al., 2009; Dong et al., 2015; Dong et al., 2013; Goldberg et al., 2016; Löwer & Schneider, 2009; Samudrala et al., 2009; Tay et al., 2010; Wang et al., 2011; Wang et al., 2013a; Wang et al., 2013b; Xue et al., 2018; Yang et al., 2010; Yang et al., 2013), type IV (Burstein et al., 2009; Hong et al., 2019; Wang et al., 2014; Xiong et al., 2018; Zou et al., 2013) and type VI substrates (Sen et al., 2019). Chapter 2 presents three recent studies on type III, IV and VI secretion system substrate identification. As these substrate predictors have been widely used to assist follow-up experimental validation or further included into other toolkits or integrative toolkits (Dong et al., 2015; Eichinger et al., 2016; Jehl et al., 2011), their capability and practicality could be further expanded to comprehensively predict each type of substrates within a uniform toolbox. Towards this target, (Dhroso et al., 2018) developed a toolkit to discover substrates across various secretion systems, but mixed all substrates together from the analysis. As a result, they could identify substrates without annotation of their exact type, which largely reduces its practical usability for biologists.

Technically, most existing methods select a certain machine learning algorithm as a base to train predictive model with some features, e.g. support vector machine (SVM) (Dhroso et al., 2018; Dong et al., 2013; Samudrala et al., 2009; Sato et al., 2011; Wang et al., 2011; Wang et al., 2013b; Yang et al., 2010; Zou et al., 2013), Naive Bayes (NB) (Tay et al., 2010), random forest (RF) (Luo et al., 2015; Yang et al., 2013), Markov Model (MM) (Wang et al., 2013a), and Convolutional Neural Network (CNN) (Hong et al., 2019; Xue et al., 2018). To further improve the prediction accuracy, some approaches utilize a combination of multiple machine learning algorithms: (Löwer & Schneider, 2009) adopted both ANN and SVM for model training, and (Dong et al., 2015; Goldberg et al., 2016) combined a BLAST-based predictor and SVM-based classifier in type III substrate recognition, (Burstein et al., 2009; Xiong et al., 2018) trained and integrated multiple machine learning algorithms for more accurate identification of type IV substrates, and (Sen et al., 2019) trained and integrated multiple machine learning algorithms, including MLP, SVM, KNN, NB and RF, for identification of type VI substrates.

As the number of feature extraction methods and machine learning algorithms increased, many ensemble strategies were applied to obtain more powerful and stable models through exploring how to effectively take advantage of multiple features in combination with various machine learning algorithms. For example, for the type IV substrate prediction (Hong et al., 2019; Zou et al., 2013), the authors trained different models using a certain machine learning algorithm with distinct features, and integrated them by using the majority vote strategy. As an alternative, (Burstein et al., 2009) combined seven types of features, trained four models based on SVM, MLP, NB and BN, and finally integrated them as an ensemble model via the majority vote strategy. (Xiong et al., 2018) characterized type IV substrates by a single feature, based on which eight preliminary models were trained and then integrated using a stacking strategy. (Sen et al., 2019) combined nine features, and trained five individual models based on five machine learning algorithms, then further integrated these models as the final model based on the voting strategy to predict type VI secreted substrates.

Recent work (An et al., 2018; Zeng & Zou, 2017) provide comprehensive surveys and deep performance evaluation of currently available methods and tools for the prediction of three major types of substrate proteins, namely type III, IV and VI substrates. These efforts reveal that the current methods for substrate prediction differ significantly from one another in terms of machine learning algorithms, dataset collection and curation, feature extraction and selection, prediction performance, availability via designated web servers and/or stand-alone software, and applicability. Based on these observations, (An et al., 2018) further proposed and built new ensemble models by simply integrating the existing substrate predictors for type III and IV substrate prediction, which outperformed all reviewed predictors based on the benchmark test and in specific test cases, indicating the necessity to and providing new insights into developing more powerful predictors.

1.4.2 Progress in secreted substrate knowledgebases

As substrate data collection and curation are often time-consuming and require specific biological knowledge, it is beneficial to gather various types of experimentally validated substrates for statistical analyses and new substrate discovery. Considerable efforts have been made to collect various types of secreted substrates with detailed attribute information, providing analytic functions that would assist users in systematic analysis, and integrating predictive toolkits for new substrate discovery (Zeng & Zou, 2017). These efforts greatly

facilitate known substrate analysis in terms of their functions, secretion system dependence and bacterial species, and at the same time allow for a better prediction of secreted substrates.

T3SEdb (Tay et al., 2010), T3DB (Wang et al., 2012) and BEAN2.0 (Dong et al., 2015) collected and annotated type III substrate proteins, but each approach differed in the numbers and functional modules. SecReT4 (Bi et al., 2013) and SecReT6 (Li et al., 2015) presented and annotated type VI and IV substrate proteins, respectively. SecretEPDB (An et al., 2017) further integrated previous known datasets and manually collected additional substrates to build a more universal resource for type III, IV and VI substrate proteins. EffectiveDB (Eichinger et al., 2016; Jehl et al., 2011) provided a very large number of predicted type III, IV and VI substrate proteins across multiple bacterial species, picking up experimentally validated substrates but without means for browsing them or investigating their detailed information.

Beyond providing data annotation and basic functions to investigate known secreted substrates, these toolkits offer various advanced functions to facilitate potential substrate prediction. Specifically, to provide type III substrate protein prediction, T3SEdb employs a selectable NaiveBayes or BayesNet model, T3DB integrates BPBAac (Wang et al., 2011), T3SEpre and a Markov model, and BEAN 2.0 integrates an updated model based on BEAN (Dong et al., 2013). Lastly, EffectiveDB integrates EffectiveT3 (Arnold et al., 2009) and T4SEpre (Wang et al., 2014), and includes the algorithms EffectiveCCBD and EffectiveELD for type III or IV substrate prediction.

1.4.3 Progress in feature generating toolkits

To successfully construct machine learning based predictors for accurate secreted substrate screening, it is crucial to extract representative features from the raw protein sequences, as to encode their underlying relationships. Effective feature extraction directly contributes to a successful machine learning based predictor, but usually involves complicated mathematical formulae and expert programming skills. To quicken the development and improve the performance of sequence based predictors, many computational efforts have been made to generate features that serve as inputs of machine-learning models (Cao et al., 2013; Li et al., 2006; Liu et al., 2015a; Liu et al., 2015b; Liu et al., 2016; Liu, 2017a; Liu et al., 2017b; Rao et al., 2011; Xiao et al., 2015; Zhang et al., 2017).

These feature generators target different types of sequences, including protein, DNA and RNA sequences, extract different aspects of features, and are distributed in different formats, such as online and standalone toolkits. Among them, Propy (Cao et al., 2013), protr/ProtrWeb (Xiao et al., 2015) and PROFEAT (Li et al., 2006; Rao et al., 2011; Zhang et al., 2017) extract various features from protein sequences considering sequence information and physiochemical information. repDNA (Liu et al., 2015a) and repRNA (Liu et al., 2016) generate various modes of feature vectors based on DNA and RNA sequences respectively, while Pse-in-One (Liu et al., 2015b) integrates these features together to provide a universal service for feature generation based on protein, DNA and RNA sequences. By additionally integrating model construction and assessment, BioSeq-Analysis (Liu, 2017a) provides a pipeline for users to automatically analyze protein, DNA and RNA sequences.

1.4.4 Challenges in analyzing and predicting secreted substrates

The number of experimentally validated substrates makes it possible to computationally analyze both known substrates and known non-substrates to predict new substrates. As substrates with new characteristics are discovered, these existing computational methods and tools reveal certain drawbacks and limitations. The increasing diversity of substrates calls for more powerful, heterogeneous and informative machine learning features to capture as many characteristics as possible from currently known substrates. Different features often have different data distributions, and their patterns could be mined and learned with different efficiency by alternative machine learning algorithms. A future direction is to intelligently integrate multiple machine learning algorithms with a wide array of feature encoding methods, especially considering the number of new efficient and powerful machine learning algorithms and feature encoding methods grows over time.

From a functionality point of view, existing substrate predictors can only predict substrates of one specific secretion system, which limits the practical usage in common scenarios. One group (Dhroso et al., 2018) attempted to discover all substrates from the various secretion systems, but the output was not very meaningful considering that it did not clarify which secretion system an identified was predicted to be secreted by. Existing substrate knowledgebases usually target one or a few types of substrates and offer limited options for users to analyze known substrates and predict potential substrates. Without a universal platform to integrate substrates and associated analytic and predictive tools, it is expected to be a cumbersome task if users have to investigate or compare known substrates across

secretion systems, or specify the closest relationship among certain known substrates for a given protein candidate. The latter case represents a common demand: as predictive models can only recognize a potential substrate, further locating its potential homologs and functional analogs from comparisons to known substrates could inspire users to infer possible structural and functional attributes, and guide design of the following experimental validation protocols. Deeply understanding users' needs and fully imagining their usage scenarios can create new tools to provide practical and seamless service for future substrate analysis and prediction.

Existing substrate predictors have been often presented in the form of web servers, which is being challenged by the rapid accumulation of protein data. Due to the limited computing power, existing predictors allow a small number of sequences per submission, which significantly slows down the substrate prediction procedure. To enable a predictor to process genome-scale sequence data, which is common in practical use, distributed computing technologies promise to provide a solution for the development of more efficient and powerful frameworks. Providing standalone toolkits could be considered as another complementary solution to reduce the computing load of central servers by allowing users to execute the prediction tasks locally at their own computing facilities. This additionally benefits users, as standalone toolkits could be further integrated into their pipeline for automatic downstream analysis and modelling.

1.5 Research contributions

The overall target of my PhD project was to conduct a comprehensive analysis and prediction of bacterial secretion system substrates based on machine learning, and therefore construct a uniformed platform to systemically analyze known substrates, predict potential substrates and recognize their mutual relationship. Accordingly, contributions during my PhD candidature are summarized in Fig. 1.3.

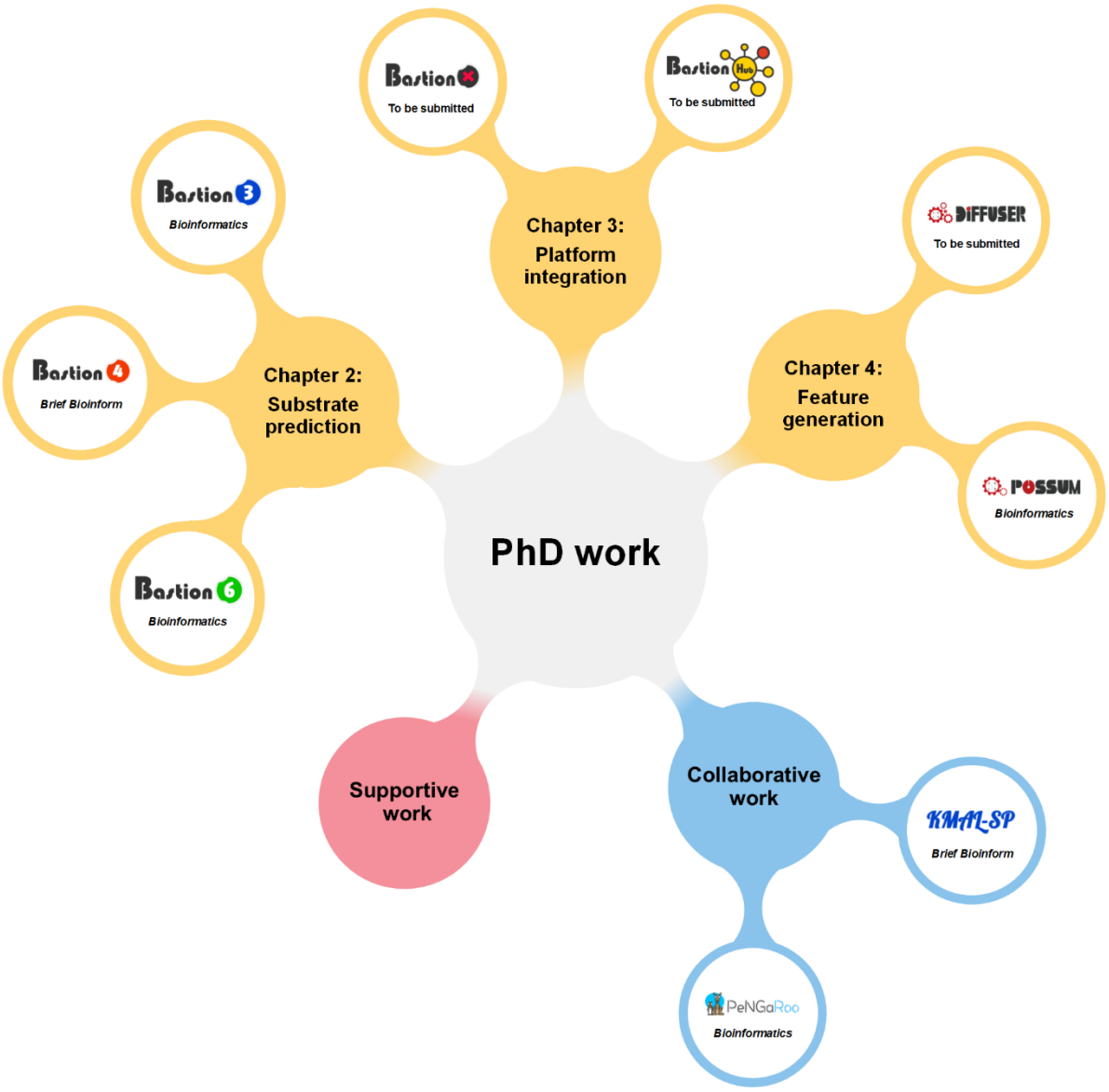


Fig 1.3. Summary of the work done during my PhD study. First-authored publications are marked in yellow (Wang et al., 2017; Wang et al., 2018; Wang et al., 2019a; Wang et al., 2019b), co-first or corresponding author publications in blue (Zhang et al., 2018; Zhang et al., 2019), and other co-authored papers in pink (Grinter et al., 2018; Song et al., 2017; Zhao et al., 2019).

1.5.1 Computational identification of single types of secreted substrates in Gram-negative bacteria

Computational prediction provides for a preliminary screening of potential substrates, which narrows down the work load for following experimental validation. However, it remains a big challenge to accurately predict these substrates, as there are no observable signals or patterns existing in their protein sequences. Towards more accurate substrate prediction, we have conducted comprehensive bioinformatics analyses on three well-studied types of substrates (III, IV and VI), systematically investigated previous computational methods on their

performance, and finally proposed new machine learning based methods with demonstrated improved performance. To achieve this, our efforts cover main and key stages during machine learning model construction, including: (1) comprehensively exploring and comparing a wide range of features across various aspects of sequence-based features, physiochemical features, structure based features and evolutionary information based features; (2) exploiting multiple classical and new machine learning methods, such as SVM, NB, RF, KNN, LR, and LightGBM; and (3) further integrating those features with machine learning algorithms through investigating various ensemble strategies. Three state-of-the-art toolkits, *i.e.* Bastion3 (Wang et al., 2019a), Bastion4 (Wang et al., 2019b) and Bastion6 (Wang et al., 2018), have been developed with user-friendly web interfaces to provide public service for accurate and robust prediction of type III, IV and VI secreted substrates, respectively. The conducted analysis, proposed methodologies, together with developed toolkits, are expected to enhance researchers' understanding on substrates in Gram-negative bacteria, facilitate later experimental validation, and further inspire toolkit development in or beyond substrate prediction.

1.5.2 Integrative system for identification and annotation of secreted substrates in Gram-negative bacteria

Beyond predictors that could predict a certain type of substrate, an integrative system was constructed to systematically analyze and predict various types of substrates. Development of this system consisted of a two-stage process, resulting in BastionX and BastionHub. BastionX has been developed to comprehensively predict each type of substrates. This is an integrative toolkit suite that includes an online server utilizing a distributed framework and a standalone toolkit. BastionX's distributed framework enables high-throughput prediction by a 10-fold improvement in terms of prediction efficiency, while its standalone toolkit allows local sequence analysis and seamless integration into a user's own pipeline for downstream analysis. Together these characteristics contribute to the practical annotation of genome-scale protein sequences with their possible substrate types, and therefore provide the landscape of substrate distributions within bacterial genomes. BastionX was then integrated into a universal platform - BastionHub - to annotate, analyze and predict various types of secreted substrates in Gram-negative bacteria. By providing a range of functional analytic modules, and interconnecting them as an interactive system, BastionHub offers a one-stop service on known substrate investigation, potential substrate prediction, and relationship analysis between known and potential substrates.

1.5.3 Computational toolkits to facilitate development of machine learning based predictors

As a key and indispensable step in machine learning-based analysis and modelling, feature extraction is inundated with mathematical derivation and formulation, and is therefore time-consuming and complicated. Aiming at speeding up the development and improving the performance of machine learning models, two contributions have been made to facilitate automatic feature generating. A specialized toolkit POSSUM has been developed to enable users to generate a broad spectrum of PSSM-based features based on protein sequences. POSSUM implements a wide range of such algorithms existing in previous literature, provides an easy-to-use interface including both web server and standalone toolkit, and offers necessary functionality and flexibility for users to customize their features of interest. DIFFUSER, in the format of web server and standalone toolkit, was further developed to extend POSSUM to generate a broader spectrum of heterogeneous features from biological sequences, including DNA, RNA and protein sequences. The online server of DIFFUSER was designed and implemented within a distributed architecture, and thus achieved a 10-fold improvement in computational throughput. Representing the most comprehensive feature generator, DIFFUSER provides an all-in-one service to generate a great variety of heterogeneous features from biological sequence data in a high-throughput manner. Both POSSUM and DIFFUSER have been well demonstrated in practical applications, and therefore contribute to accelerate machine leaning based analysis and modelling to a wider scientific community.

In summary, a series of computational programmes have been made in my PhD study to facilitate comprehensive and systematic annotation, analysis and prediction of substrates secreted by Gram-negative bacteria. Additional contributions to streamlined and automatic feature generation have expedited the development of substrate predictors, which can further contribute to the acceleration of general machine learning-based research.

All bioinformatics resources and tools generated during this PhD project have been implemented and made publicly available at Monash University through the collaboration with the Monash e-Research Centre. These resources will facilitate bacterial substrate discoveries, biochemical property identification, structure or function inference and experimental validation, as well as machine learning based modelling and analysis in biomedicine and bioinformatics research (Fig. 1.4).

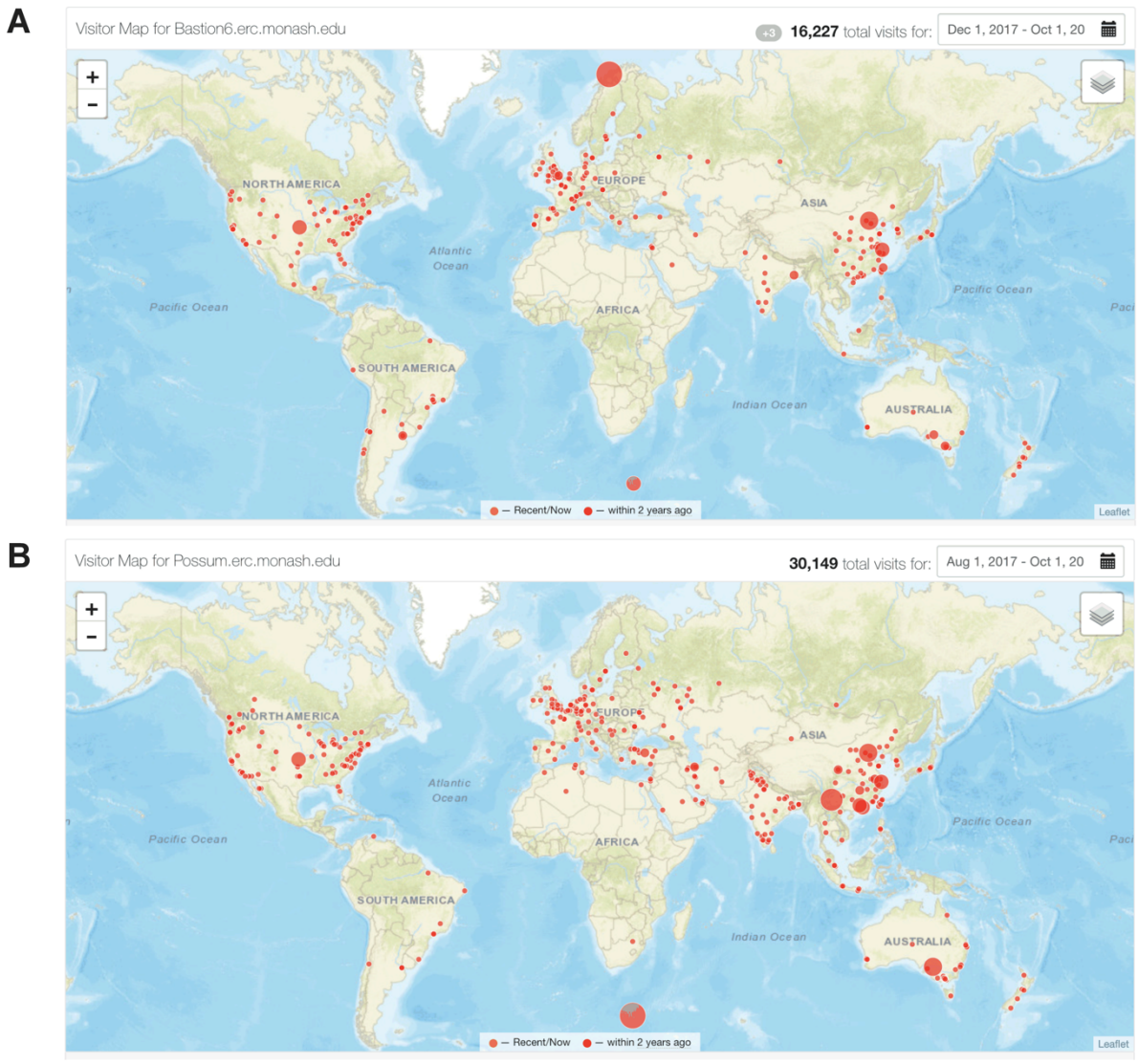


Fig 1.4. Impact of two representative toolkits developed in this PhD study. (A) As the first available type VI secreted substrate predictor, Bastion6 has attracted more than 16000 visits from more than 40 countries and regions; (B) As the only specialized PSSM profile-based feature generator, POSSUM has attracted more than 30000 visits from more than 55 countries and regions, and obtained 293 downloads of its standalone toolkit.

References

An, Y., Wang, J., Li, C., Leier, A., Marquez-Lago, T., Wilksch, J., Zhang, Y., Webb, G. I., Song, J., & Lithgow, T. (2018). Comprehensive assessment and performance improvement of effector protein predictors for bacterial secretion systems III, IV and VI. *Brief Bioinform, 19*(1), 148-161. doi:10.1093/bib/bbw100

An, Y., Wang, J., Li, C., Revote, J., Zhang, Y., Naderer, T., Hayashida, M., Akutsu, T., Webb, G. I., Lithgow, T., & Song, J. (2017). SecretEPDB: a comprehensive web-based resource for secreted effector proteins of the bacterial types III, IV and VI secretion systems. *Sci Rep, 7*, 41031. doi:10.1038/srep41031

Angermueller, C., Parnamaa, T., Parts, L., & Stegle, O. (2016). Deep learning for computational biology. *Mol Syst Biol, 12*(7), 878. doi:10.15252/msb.20156651

Arnold, R., Brandmaier, S., Kleine, F., Tischler, P., Heinz, E., Behrens, S., Niinikoski, A., Mewes, H. W., Horn, M., & Rattei, T. (2009). Sequence-based prediction of type III secreted proteins. *PLoS Pathog, 5*(4), e1000376. doi:10.1371/journal.ppat.1000376

Awada, W., Khoshgoftaar, T. M., Dittman, D., Wald, R., & Napolitano, A. (2012). *A review of the stability of feature selection techniques for bioinformatics data.* Paper presented at the Information Reuse and Integration (IRI), 2012 IEEE 13th International Conference on.

Barbosa, V. A. A., & Lery, L. M. S. (2019). Insights into Klebsiella pneumoniae type VI secretion system transcriptional regulation. *BMC genomics, 20*(1), 506. doi:10.1186/s12864-019-5885-9

Bartlett, P. L. (2002). An introduction to reinforcement learning theory: Value function methods. *Advanced Lectures on Machine Learning, 2600*, 184-202.

Bermingham, M. L., Pong-Wong, R., Spiliopoulou, A., Hayward, C., Rudan, I., Campbell, H., Wright, A. F., Wilson, J. F., Agakov, F., Navarro, P., & Haley, C. S. (2015). Application of high-dimensional feature selection: evaluation for genomic prediction in man. *Sci Rep, 5*, 10312. doi:10.1038/srep10312

Bi, D., Liu, L., Tai, C., Deng, Z., Rajakumar, K., & Ou, H. Y. (2013). SecReT4: a web-based bacterial type IV secretion system resource. *Nucleic Acids Res, 41*(Database issue), D660-665. doi:10.1093/nar/gks1248

Birtalan, S. C., Phillips, R. M., & Ghosh, P. (2002). Three-dimensional secretion signals in chaperone-effector complexes of bacterial pathogens. *Mol Cell, 9*(5), 971-980.

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

Brodmann, M., Dreier, R. F., Broz, P., & Basler, M. (2017). Francisella requires dynamic type VI secretion system and ClpB to deliver effectors for phagosomal escape. *Nat Commun, 8*, 15853. doi:10.1038/ncomms15853

Buchko, G. W., Niemann, G., Baker, E. S., Belov, M. E., Smith, R. D., Heffron, F., Adkins, J. N., & McDermott, J. E. (2010). A multi-pronged search for a common structural motif in the secretion signal of Salmonella enterica serovar Typhimurium type III effector proteins. *Mol Biosyst, 6*(12), 2448-2458. doi:10.1039/c0mb00097c

Burkinshaw, B. J., & Strynadka, N. C. (2014). Assembly and structure of the T3SS. *Biochim Biophys Acta, 1843*(8), 1649-1663. doi:10.1016/j.bbamcr.2014.01.035

Burstein, D., Amaro, F., Zusman, T., Lifshitz, Z., Cohen, O., Gilbert, J. A., Pupko, T., Shuman, H. A., & Segal, G. (2016). Genomic analysis of 38 Legionella species identifies large and diverse effector repertoires. *Nat Genet, 48*(2), 167-175. doi:10.1038/ng.3481

Burstein, D., Satanower, S., Simovitch, M., Belnik, Y., Zehavi, M., Yerushalmi, G., Ben-Aroya, S., Pupko, T., & Banin, E. (2015). Novel type III effectors in Pseudomonas aeruginosa. *MBio, 6*(2), e00161. doi:10.1128/mBio.00161-15

Burstein, D., Zusman, T., Degtyar, E., Viner, R., Segal, G., & Pupko, T. (2009). Genome-scale identification of Legionella pneumophila effectors using a machine learning approach. *PLoS Pathog, 5*(7), 6974-6974.

Camacho, D. M., Collins, K. M., Powers, R. K., Costello, J. C., & Collins, J. J. (2018). Next-Generation Machine Learning for Biological Networks. *Cell, 173*(7), 1581-1592. doi:10.1016/j.cell.2018.05.015

Cao, D. S., Xu, Q. S., & Liang, Y. Z. (2013). propy: a tool to generate various modes of Chou's PseAAC. *Bioinformatics, 29*(7), 960-962. doi:10.1093/bioinformatics/btt072

Cascales, E., & Christie, P. J. (2003). The versatile bacterial type IV secretion systems. *Nature Reviews Microbiology, 1*(2), 137-149.

Chen, T., & Guestrin, C. (2016). *Xgboost: A scalable tree boosting system.* Paper presented at the Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining.

Chen, W., Xing, P., & Zou, Q. (2017). Detecting N(6)-methyladenosine sites from RNA transcriptomes using ensemble Support Vector Machines. *Sci Rep, 7*, 40242. doi:10.1038/srep40242

Chen, X. W., & Jeong, J. C. (2009). Sequence-based prediction of protein interaction sites with an integrative method. *Bioinformatics, 25*(5), 585-591. doi:10.1093/bioinformatics/btp039

Chen, Y., Wong, J., Sun, G. W., Liu, Y., Tan, G. Y., & Gan, Y. H. (2011). Regulation of type VI secretion system during Burkholderia pseudomallei infection. *Infect Immun, 79*(8), 3064-3073. doi:10.1128/IAI.05148-11

Chou, K. C. (2011). Some remarks on protein attribute prediction and pseudo amino acid composition. *J Theor Biol, 273*(1), 236-247.

Chou, K. C., & Cai, Y. D. (2004). Prediction of protein subcellular locations by GO-FunD-PseAA predictor. *Biochem Biophys Res Commun, 320*(4), 1236-1239. doi:10.1016/j.bbrc.2004.06.073

Christie, P. J. (2016). The Mosaic Type IV Secretion Systems. *EcoSal Plus, 7*(1). doi:10.1128/ecosalplus.ESP-0020-2015

Cianfanelli, F. R., Monlezun, L., & Coulthurst, S. J. (2016). Aim, Load, Fire: The Type VI secretion system, a bacterial nanoweapon. *Trends Microbiol, 24*(1), 51-62.

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

Costa, T. R., Felisberto-Rodrigues, C., Meir, A., Prevost, M. S., Redzej, A., Trokter, M., & Waksman, G. (2015). Secretion systems in Gram-negative bacteria: structural and mechanistic insights. *Nat Rev Microbiol, 13*(6), 343-359. doi:10.1038/nrmicro3456

d'Enfert, C., Ryter, A., & Pugsley, A. P. (1987). Cloning and expression in Escherichia coli of the Klebsiella pneumoniae genes for production, surface localization and secretion of the lipoprotein pullulanase. *EMBO J, 6*(11), 3531-3538.

Dalbey, R. E., & Kuhn, A. (2012). Protein traffic in Gram-negative bacteria--how exported and secreted proteins find their way. *FEMS Microbiol Rev, 36*(6), 1023-1045. doi:10.1111/j.1574-6976.2012.00327.x

Dean, P., & Kenny, B. (2009). The effector repertoire of enteropathogenic E. coli: ganging up on the host cell. *Curr Opin Microbiol, 12*(1), 101-109. doi:10.1016/j.mib.2008.11.006

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint, arXiv:1810.04805*. doi:arXiv:1810.04805

Dhroso, A., Eidson, S., & Korkin, D. (2018). Genome-wide prediction of bacterial effector candidates across six secretion system types using a feature-based statistical framework. *Sci Rep, 8*(1), 17209. doi:10.1038/s41598-018-33874-1

Dong, X., Lu, X., & Zhang, Z. (2015). BEAN 2.0: an integrated web resource for the identification and functional analysis of type III secreted effectors. *Database (Oxford), 2015*, bav064. doi:10.1093/database/bav064

Dong, X., Zhang, Y. J., & Zhang, Z. (2013). Using weakly conserved motifs hidden in secretion signals to identify type-III effectors from bacterial pathogen genomes. *PLoS One, 8*(2), e56632. doi:10.1371/journal.pone.0056632

Durand, E., Cambillau, C., Cascales, E., & Journet, L. (2014). VgrG, Tae, Tle, and beyond: the versatile arsenal of Type VI secretion effectors. *Trends Microbiol, 22*(9), 498-507. doi:10.1016/j.tim.2014.06.004

Edgar, R. C. (2010). Search and clustering orders of magnitude faster than BLAST. *Bioinformatics, 26*(19), 2460-2461. doi:10.1093/bioinformatics/btq461

Eichinger, V., Nussbaumer, T., Platzer, A., Jehl, M.-A., Arnold, R., & Rattei, T. (2016). EffectiveDB—updates and novel features for a better annotation of bacterial secreted proteins and Type III, IV, VI secretion systems. *Nucleic Acids Res, 44*(D1), D669-D674.

Ernst, N. H., Reeves, A. Z., Ramseyer, J. E., & Lesser, C. F. (2018). High-Throughput Screening of Type III Secretion Determinants Reveals a Major Chaperone-Independent Pathway. *MBio, 9*(3). doi:10.1128/mBio.01050-18

Fan, E., Chauhan, N., Udatha, D. B., Leo, J. C., & Linke, D. (2016). Type V Secretion Systems in Bacteria. *Microbiol Spectr, 4*(1). doi:10.1128/microbiolspec.VMBF-0009-2015

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861-874. doi:10.1016/j.patrec.2005.10.010

Flaugnatti, N., Le, T. T., Canaan, S., Aschtgen, M. S., Nguyen, V. S., Blangy, S., Kellenberger, C., Roussel, A., Cambillau, C., Cascales, E., & Journet, L. (2016). A phospholipase A1 antibacterial Type VI secretion effector interacts directly with the C-terminal domain of the VgrG spike protein for delivery. *Mol Microbiol, 99*(6), 1099-1118. doi:10.1111/mmi.13292

Folders, J., Algra, J., Roelofs, M. S., van Loon, L. C., Tommassen, J., & Bitter, W. (2001). Characterization of Pseudomonas aeruginosa chitinase, a gradually secreted protein. *J Bacteriol, 183*(24), 7044-7052. doi:10.1128/JB.183.24.7044-7052.2001

Freedman, D. A. (2009). *Statistical models: theory and practice*: cambridge university press.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics, 29*(5), 1189-1232. doi:DOI 10.1214/aos/1013203451

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis, 38*(4), 367-378. doi: 10.1016/S0167-9473(01)00065-2

Friedman, N., Dan, G., & Goldszmidt, M. (1997). Bayesian Network Classifiers. *Machine learning, 29*(2-3), 131-163.

Galan, J. E., & Curtiss, R., 3rd. (1989). Cloning and molecular characterization of genes whose products allow Salmonella typhimurium to penetrate tissue culture cells. *Proc Natl Acad Sci U S A, 86*(16), 6383-6387.

Garg, A., & Gupta, D. (2008). VirulentPred: a SVM based prediction method for virulent proteins in bacterial pathogens. *BMC Bioinformatics, 9*, 62. doi:10.1186/1471-2105-9-62

Giles, C. L., Kuhn, G. M., & Williams, R. J. (1994). Dynamic Recurrent Neural Networks - Theory and Applications. *Ieee Transactions on Neural Networks, 5*(2), 153-156. doi:Doi 10.1109/Tnn.1994.8753425

Glaser, F., Rosenberg, Y., Kessel, A., Pupko, T., & Ben-Tal, N. (2005). The ConSurf-HSSP database: the mapping of evolutionary conservation among homologs onto PDB structures. *Proteins, 58*(3), 610-617. doi:10.1002/prot.20305

Goldberg, T., Rost, B., & Bromberg, Y. (2016). Computational prediction shines light on type III secretion origins. *Sci Rep, 6*, 34516. doi:10.1038/srep34516

Grinter, R., Hay, I. D., Song, J., Wang, J., Teng, D., Dhanesakaran, V., Wilksch, J. J., Davies, M. R., Littler, D., Beckham, S. A., Henderson, I. R., Strugnell, R. A., Dougan, G., & Lithgow, T. (2018). FusC, a member of the M16 protease family acquired by bacteria for iron piracy against plants. *PLoS Biol, 16*(8), e2006026. doi:10.1371/journal.pbio.2006026

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

Hall, M. A. (1999). *Correlation-based feature selection for machine learning.* The University of Waikato.

Hay, I. D., Belousoff, M. J., Dunstan, R. A., Bamert, R. S., & Lithgow, T. (2018). Structure and Membrane Topography of the Vibrio-Type Secretin Complex from the Type 2 Secretion System of Enteropathogenic Escherichia coli. *J Bacteriol, 200*(5). doi:10.1128/JB.00521-17

Hay, I. D., Belousoff, M. J., & Lithgow, T. (2017). Structural Basis of Type 2 Secretion System Engagement between the Inner and Outer Bacterial Membranes. *MBio, 8*(5). doi:10.1128/mBio.01344-17

Heinz, E., & Lithgow, T. (2014). A comprehensive analysis of the Omp85/TpsB protein superfamily structural diversity, taxonomic occurrence, and evolution. *Front Microbiol, 5*, 370. doi:10.3389/fmicb.2014.00370

Heinz, E., Stubenrauch, C. J., Grinter, R., Croft, N. P., Purcell, A. W., Strugnell, R. A., Dougan, G., & Lithgow, T. (2016). Conserved Features in the Structure, Mechanism, and Biogenesis of the Inverse Autotransporter Protein Family. *Genome Biol Evol, 8*(6), 1690-1705. doi:10.1093/gbe/evw112

Henderson, I. R., Navarro-Garcia, F., Desvaux, M., Fernandez, R. C., & Ala'Aldeen, D. (2004). Type V protein secretion pathway: the autotransporter story. *Microbiol Mol Biol Rev, 68*(4), 692-744. doi:10.1128/MMBR.68.4.692-744.2004

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780. doi:DOI 10.1162/neco.1997.9.8.1735

Hohlfeld, S., Pattis, I., Puls, J., Plano, G. V., Haas, R., & Fischer, W. (2006). A C-terminal translocation signal is necessary, but not sufficient for type IV secretion of the Helicobacter pylori CagA protein. *Mol Microbiol, 59*(5), 1624-1637. doi:10.1111/j.1365-2958.2006.05050.x

Hong, J., Luo, Y., Mou, M., Fu, J., Zhang, Y., Xue, W., Xie, T., Tao, L., Lou, Y., & Zhu, F. (2019). Convolutional neural network-based annotation of bacterial type IV secretion system effectors with enhanced accuracy and reduced false discovery. *Brief Bioinform*.

Huang, Y., Niu, B., Gao, Y., Fu, L., & Li, W. (2010). CD-HIT Suite: a web server for clustering and comparing biological sequences. *Bioinformatics, 26*(5), 680-682. doi:10.1093/bioinformatics/btq003

I, G., J, P.-A., & M, M. (2014). *Generative adversarial nets.* Paper presented at the Advances in Neural Information Processing Systems.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning : with applications in R*. New York: Springer.

Jamet, A., & Nassif, X. (2015). New players in the toxin field: polymorphic toxin systems in bacteria. *MBio, 6*(3), e00285-00215. doi:10.1128/mBio.00285-15

Jehl, M. A., Arnold, R., & Rattei, T. (2011). Effective--a database of predicted secreted bacterial proteins. *Nucleic Acids Res, 39*(Database issue), D591-595. doi:10.1093/nar/gkq1154

Jensen, F. V. (1996). *An introduction to Bayesian networks* (Vol. 210): UCL press London.
Jeong, K. C., Ghosal, D., Chang, Y. W., Jensen, G. J., & Vogel, J. P. (2017). Polar delivery of Legionella type IV secretion system substrates is essential for virulence. *Proc Natl Acad Sci U S A, 114*(30), 8077-8082. doi:10.1073/pnas.1621438114
Kanonenberg, K., Schwarz, C. K., & Schmitt, L. (2013). Type I secretion systems - a story of appendices. *Res Microbiol, 164*(6), 596-604. doi:10.1016/j.resmic.2013.03.011
Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 3149-3157.
Koskiniemi, S., Lamoureux, J. G., Nikolakakis, K. C., t'Kint de Roodenbeke, C., Kaplan, M. D., Low, D. A., & Hayes, C. S. (2013). Rhs proteins from diverse bacteria mediate intercellular competition. *Proc Natl Acad Sci U S A, 110*(17), 7032-7037. doi:10.1073/pnas.1300627110
Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the Acm, 60*(6), 84-90. doi:10.1145/3065386
Kuldau, G. A., De Vos, G., Owen, J., McCaffrey, G., & Zambryski, P. (1990). The virB operon of Agrobacterium tumefaciens pTiC58 encodes 11 open reading frames. *Mol Gen Genet, 221*(2), 256-266.
Larranaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J. A., Armananzas, R., Santafe, G., Perez, A., & Robles, V. (2006). Machine learning in bioinformatics. *Brief Bioinform, 7*(1), 86-112.
Lee, S. H., & Galan, J. E. (2004). Salmonella type III secretion-associated chaperones confer secretion-pathway specificity. *Mol Microbiol, 51*(2), 483-495. doi:10.1046/j.1365-2958.2003.03840.x
Li, J., Yao, Y., Xu, H. H., Hao, L., Deng, Z., Rajakumar, K., & Ou, H. Y. (2015). SecReT6: a web-based resource for type VI secretion systems found in bacteria. *Environ Microbiol, 17*(7), 2196-2202. doi:10.1111/1462-2920.12794
Li, Z. R., Lin, H. H., Han, L. Y., Jiang, L., Chen, X., & Chen, Y. Z. (2006). PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic Acids Res, 34*(Web Server issue), W32-37. doi:10.1093/nar/gkl305
Libbrecht, M. W., & Noble, W. S. (2015). Machine learning applications in genetics and genomics. *Nat Rev Genet, 16*(6), 321-332. doi:10.1038/nrg3920
Lien, Y. W., & Lai, E. M. (2017). Type VI Secretion Effectors: Methodologies and Biology. *Front Cell Infect Microbiol, 7*, 254. doi:10.3389/fcimb.2017.00254
Lilic, M., Vujanac, M., & Stebbins, C. E. (2006). A common structural motif in the binding of virulence factors to bacterial secretion chaperones. *Mol Cell, 21*(5), 653-664. doi:10.1016/j.molcel.2006.01.026
Lin, J., Zhang, W., Cheng, J., Yang, X., Zhu, K., Wang, Y., Wei, G., Qian, P. Y., Luo, Z. Q., & Shen, X. (2017). A Pseudomonas T6SS effector recruits PQS-containing outer membrane vesicles for iron acquisition. *Nat Commun, 8*, 14888. doi:10.1038/ncomms14888
Linhartova, I., Bumba, L., Masin, J., Basler, M., Osicka, R., Kamanova, J., Prochazkova, K., Adkins, I., Hejnova-Holubova, J., Sadilkova, L., Morova, J., & Sebo, P. (2010). RTX proteins: a highly diverse family secreted by a common mechanism. *FEMS Microbiol Rev, 34*(6), 1076-1112. doi:10.1111/j.1574-6976.2010.00231.x
Liu, B. (2017a). BioSeq-Analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches. *Brief Bioinform*. doi:10.1093/bib/bbx165

Liu, B., Liu, F., Fang, L., Wang, X., & Chou, K. C. (2015a). repDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects. *Bioinformatics, 31*(8), 1307-1309. doi:10.1093/bioinformatics/btu820

Liu, B., Liu, F., Fang, L., Wang, X., & Chou, K. C. (2016). repRNA: a web server for generating various feature vectors of RNA sequences. *Mol Genet Genomics, 291*(1), 473-481. doi:10.1007/s00438-015-1078-7

Liu, B., Liu, F., Wang, X., Chen, J., Fang, L., & Chou, K. C. (2015b). Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nucleic Acids Res, 43*(W1), W65-71. doi:10.1093/nar/gkv458

Liu, B., Wu, H., Zhang, D., Wang, X., & Chou, K. C. (2017b). Pse-Analysis: a python package for DNA/RNA and protein/ peptide sequence analysis based on pseudo components and kernel methods. *Oncotarget, 8*(8), 13338-13343. doi:10.18632/oncotarget.14524

Lloyd, S. A., Forsberg, A., Wolf-Watz, H., & Francis, M. S. (2001). Targeting exported substrates to the Yersinia TTSS: different functions for different signals? *Trends Microbiol, 9*(8), 367-371.

Löwer, M., & Schneider, G. (2009). Prediction of Type III Secretion Signals in Genomes of Gram-Negative Bacteria. *PLoS One, 4*(6), e5917.

Luo, J., Li, W., Liu, Z., Guo, Y., Pu, X., & Li, M. (2015). A sequence-based two-level method for the prediction of type I secreted RTX proteins. *Analyst, 140*(9), 3048-3056. doi:10.1039/c5an00311c

Ma, J., Pan, Z., Huang, J., Sun, M., Lu, C., & Yao, H. (2017a). The Hcp proteins fused with diverse extended-toxin domains represent a novel pattern of antibacterial effectors in type VI secretion systems. *Virulence*, 1-14. doi:10.1080/21505594.2017.1279374

Ma, J., Sun, M., Dong, W., Pan, Z., Lu, C., & Yao, H. (2017b). PAAR-Rhs proteins harbor various C-terminal toxins to diversify the antibacterial pathways of type VI secretion systems. *Environ Microbiol, 19*(1), 345-360. doi:10.1111/1462-2920.13621

Ma, L. S., Hachani, A., Lin, J. S., Filloux, A., & Lai, E. M. (2014). Agrobacterium tumefaciens deploys a superfamily of type VI secretion DNase effectors as weapons for interbacterial competition in planta. *Cell Host Microbe, 16*(1), 94-104. doi:10.1016/j.chom.2014.06.002

Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure, 405*(2), 442-451.

Mattoo, S., Lee, Y. M., & Dixon, J. E. (2007). Interactions of bacterial effector proteins with host proteins. *Curr Opin Immunol, 19*(4), 392-401. doi:10.1016/j.coi.2007.06.005

McDermott, J. E., Corrigan, A., Peterson, E., Oehmen, C., Niemann, G., Cambronne, E. D., Sharp, D., Adkins, J. N., Samudrala, R., & Heffron, F. (2011). Computational prediction of type III and IV secreted effectors in gram-negative bacteria. *Infect Immun, 79*(1), 23-32. doi:10.1128/IAI.00537-10

Meuskens, I., Saragliadis, A., Leo, J. C., & Linke, D. (2019). Type V Secretion Systems: An Overview of Passenger Domain Functions. *Front Microbiol, 10*, 1163. doi:10.3389/fmicb.2019.01163

Michiels, T., & Cornelis, G. R. (1991). Secretion of hybrid proteins by the Yersinia Yop export system. *J Bacteriol, 173*(5), 1677-1685. doi:10.1128/jb.173.5.1677-1685.1991

Min, S., Lee, B., & Yoon, S. (2017). Deep learning in bioinformatics. *Brief Bioinform, 18*(5), 851-869. doi:10.1093/bib/bbw068

Morgan, J. L. W., Acheson, J. F., & Zimmer, J. (2017). Structure of a Type-1 Secretion System ABC Transporter. *Structure, 25*(3), 522-529. doi:10.1016/j.str.2017.01.010

Mukhtar, M. S., Carvunis, A. R., Dreze, M., Epple, P., Steinbrenner, J., Moore, J., Tasan, M., Galli, M., Hao, T., Nishimura, M. T., Pevzner, S. J., Donovan, S. E., Ghamsari, L., Santhanam, B., Romero, V., Poulin, M. M., Gebreab, F., Gutierrez, B. J., Tam, S., Monachello, D., Boxem, M., Harbort, C. J., McDonald, N., Gai, L., Chen, H., He, Y., European Union Effectoromics, C., Vandenhaute, J., Roth, F. P., Hill, D. E., Ecker, J. R., Vidal, M., Beynon, J., Braun, P., & Dangl, J. L. (2011). Independently evolved virulence effectors converge onto hubs in a plant immune system network. *Science, 333*(6042), 596-601. doi:10.1126/science.1203659

Murdoch, S. L., Trunk, K., English, G., Fritsch, M. J., Pourkarimi, E., & Coulthurst, S. J. (2011). The opportunistic pathogen Serratia marcescens utilizes type VI secretion to target bacterial competitors. *J Bacteriol, 193*(21), 6057-6069. doi:10.1128/JB.05671-11

Nagai, H., Cambronne, E. D., Kagan, J. C., Amor, J. C., Kahn, R. A., & Roy, C. R. (2005). A C-terminal translocation signal required for Dot/Icm-dependent delivery of the Legionella RalF protein to host cells. *Proc Natl Acad Sci U S A, 102*(3), 826-831. doi:10.1073/pnas.0406239101

Navarro-Garcia, F., Ruiz-Perez, F., Cataldi, A., & Larzabal, M. (2019). Type VI Secretion System in Pathogenic Escherichia coli: Structure, Role in Virulence, and Acquisition. *Front Microbiol, 10*, 1965. doi:10.3389/fmicb.2019.01965

Nicolay, T., Vanderleyden, J., & Spaepen, S. (2015). Autotransporter-based cell surface display in Gram-negative bacteria. *Crit Rev Microbiol, 41*(1), 109-123. doi:10.3109/1040841X.2013.804032

Pallen, M. J., Beatson, S. A., & Bailey, C. M. (2005). Bioinformatics, genomics and evolution of non-flagellar type-III secretion systems: a Darwinian perspective. *FEMS Microbiol Rev, 29*(2), 201-229. doi:10.1016/j.femsre.2005.01.001

Patrick, M., Gray, M. D., Sandkvist, M., & Johnson, T. L. (2010). Type II Secretion in Escherichia coli. *EcoSal Plus, 4*(1). doi:10.1128/ecosalplus.4.3.4

Pearson, J. S., Zhang, Y., Newton, H. J., & Hartland, E. L. (2015). Post-modern pathogens: surprising activities of translocated effectors from E. coli and Legionella. *Curr Opin Microbiol, 23*, 73-79. doi:10.1016/j.mib.2014.11.005

Pendleton, J. N., Gorman, S. P., & Gilmore, B. F. (2013). Clinical relevance of the ESKAPE pathogens. *Expert Rev Anti Infect Ther, 11*(3), 297-308. doi:10.1586/eri.13.12

Peng, H., Long, F., & Ding, C. (2005). Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. *IEEE Transactions on Pattern Analysis & Machine Intelligence, 27*(8), 1226-1238.

Pukatzki, S., Ma, A. T., Sturtevant, D., Krastins, B., Sarracino, D., Nelson, W. C., Heidelberg, J. F., & Mekalanos, J. J. (2006). Identification of a conserved bacterial protein secretion system in Vibrio cholerae using the Dictyostelium host model system. *Proc Natl Acad Sci U S A, 103*(5), 1528-1533. doi:10.1073/pnas.0510322103

Pukatzki, S., McAuley, S. B., & Miyata, S. T. (2009). The type VI secretion system: translocation of effectors and effector-domains. *Curr Opin Microbiol, 12*(1), 11-17. doi:10.1016/j.mib.2008.11.010

Qiu, J., & Luo, Z. Q. (2013). Effector translocation by the Legionella Dot/Icm type IV secretion system. *Curr Top Microbiol Immunol, 376*, 103-115. doi:10.1007/82_2013_345

Rao, H. B., Zhu, F., Yang, G. B., Li, Z. R., & Chen, Y. Z. (2011). Update of PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic Acids Res, 39*(Web Server issue), W385-390. doi:10.1093/nar/gkr284

Rigard, M., Broms, J. E., Mosnier, A., Hologne, M., Martin, A., Lindgren, L., Punginelli, C., Lays, C., Walker, O., Charbit, A., Telouk, P., Conlan, W., Terradot, L., Sjostedt, A., & Henry, T. (2016). Francisella tularensis IglG Belongs to a Novel Family of PAAR-Like T6SS Proteins and Harbors a Unique N-terminal Extension Required for Virulence. *PLoS Pathog, 12*(9), e1005821. doi:10.1371/journal.ppat.1005821

Ripoll-Rozada, J., Zunzunegui, S., de la Cruz, F., Arechaga, I., & Cabezon, E. (2013). Functional interactions of VirB11 traffic ATPases with VirB4 and VirD4 molecular motors in type IV secretion systems. *J Bacteriol, 195*(18), 4195-4201. doi:10.1128/JB.00437-13

Russell, A. B., LeRoux, M., Hathazi, K., Agnello, D. M., Ishikawa, T., Wiggins, P. A., Wai, S. N., & Mougous, J. D. (2013). Diverse type VI secretion phospholipases are functionally plastic antibacterial effectors. *Nature, 496*(7446), 508-512.

Russell, A. B., Peterson, S. B., & Mougous, J. D. (2014). Type VI secretion system effectors: poisons with a purpose. *Nat Rev Microbiol, 12*(2), 137-148. doi:10.1038/nrmicro3185

Russmann, H., Kubori, T., Sauer, J., & Galan, J. E. (2002). Molecular and functional analysis of the type III secretion signal of the Salmonella enterica InvJ protein. *Mol Microbiol, 46*(3), 769-779. doi:10.1046/j.1365-2958.2002.03196.x

Saeys, Y., Inza, I., & Larrañaga, P. (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics, 23*(19), 2507-2517.

Salomon, D. (2016). MIX and match: mobile T6SS MIX-effectors enhance bacterial fitness. *Mob Genet Elements, 6*(1), e1123796. doi:10.1080/2159256X.2015.1123796

Salomon, D., Kinch, L. N., Trudgian, D. C., Guo, X., Klimko, J. A., Grishin, N. V., Mirzaei, H., & Orth, K. (2014). Marker for type VI secretion system effectors. *Proc Natl Acad Sci U S A, 111*(25), 9271-9276. doi:10.1073/pnas.1406110111

Salomon, D., Klimko, J. A., Trudgian, D. C., Kinch, L. N., Grishin, N. V., Mirzaei, H., & Orth, K. (2015). Type VI Secretion System Toxins Horizontally Shared between Marine Bacteria. *PLoS Pathog, 11*(8), e1005128. doi:10.1371/journal.ppat.1005128

Samudrala, R., Heffron, F., & McDermott, J. E. (2009). Accurate prediction of secreted substrates and identification of a conserved putative secretion signal for type III secretion systems. *PLoS Pathog, 5*(4), e1000375. doi:10.1371/journal.ppat.1000375

Santajit, S., & Indrawattana, N. (2016). Mechanisms of Antimicrobial Resistance in ESKAPE Pathogens. *Biomed Res Int, 2016*, 2475067. doi:10.1155/2016/2475067

Satchell, K. J. (2007). MARTX, multifunctional autoprocessing repeats-in-toxin toxins. *Infect Immun, 75*(11), 5079-5084. doi:10.1128/IAI.00525-07

Sato, Y., Takaya, A., & Yamamoto, T. (2011). Meta-analytic approach to the accurate prediction of secreted virulence effectors in gram-negative bacteria. *BMC Bioinformatics, 12*(1), 1.

Schroeder, G. N. (2017). The Toolbox for Uncovering the Functions of Legionella Dot/Icm Type IVb Secretion System Effectors: Current State and Future Directions. *Front Cell Infect Microbiol, 7*, 528. doi:10.3389/fcimb.2017.00528

Schwarz, C. K., Landsberg, C. D., Lenders, M. H., Smits, S. H., & Schmitt, L. (2012). Using an E. coli Type 1 secretion system to secrete the mammalian, intracellular protein IFABP in its active form. *J Biotechnol, 159*(3), 155-161. doi:10.1016/j.jbiotec.2012.02.005

Sen, R., Nayak, L., & De, R. K. (2019). PyPredT6: A python-based prediction tool for identification of Type VI effector proteins. *J Bioinform Comput Biol, 17*(3), 1950019. doi:10.1142/S0219720019500197

Shannon, C. E. (1948). A Mathematical Theory of Communication. *Bell System Technical Journal, 27*(3), 3 - 55.

Shneider, M. M., Buth, S. A., Ho, B. T., Basler, M., Mekalanos, J. J., & Leiman, P. G. (2013). PAAR-repeat proteins sharpen and diversify the type VI secretion system spike. *Nature, 500*(7462), 350-353. doi:10.1038/nature12453

Si, M., Zhao, C., Burkinshaw, B., Zhang, B., Wei, D., Wang, Y., Dong, T. G., & Shen, X. (2017). Manganese scavenging and oxidative stress response mediated by type VI secretion system in Burkholderia thailandensis. *Proc Natl Acad Sci U S A, 114*(11), E2233-E2242. doi:10.1073/pnas.1614902114

Slater, S. L., Sagfors, A. M., Pollard, D. J., Ruano-Gallego, D., & Frankel, G. (2018). The Type III Secretion System of Pathogenic Escherichia coli. *Curr Top Microbiol Immunol, 416*, 51-72. doi:10.1007/82_2018_116

Solomon, S. L., & Oliver, K. B. (2014). Antibiotic resistance threats in the United States: stepping back from the brink. *Am Fam Physician, 89*(12), 938-941.

Song, J., Wang, H., Wang, J., Leier, A., Marquez-Lago, T., Yang, B., Zhang, Z., Akutsu, T., Webb, G. I., & Daly, R. J. (2017). PhosphoPredict: A bioinformatics tool for prediction of human kinase-specific phosphorylation substrates and sites by integrating heterogeneous feature selection. *Sci Rep, 7*(1), 6862. doi:10.1038/s41598-017-07199-4

Sory, M. P., Boland, A., Lambermont, I., & Cornelis, G. R. (1995). Identification of the YopE and YopH domains required for secretion and internalization into the cytosol of macrophages, using the cyaA gene fusion approach. *Proc Natl Acad Sci U S A, 92*(26), 11998-12002. doi:10.1073/pnas.92.26.11998

Stavrinides, J., McCann, H. C., & Guttman, D. S. (2008). Host-pathogen interplay and the evolution of bacterial effectors. *Cell Microbiol, 10*(2), 285-292. doi:10.1111/j.1462-5822.2007.01078.x

Tay, D. M., Govindarajan, K. R., Khan, A. M., Ong, T. Y., Samad, H. M., Soh, W. W., Tong, M., Zhang, F., & Tan, T. W. (2010). T3SEdb: data warehousing of virulence effectors secreted by the bacterial Type III Secretion System. *BMC Bioinformatics, 11 Suppl 7*, S4. doi:10.1186/1471-2105-11-S7-S4

Tobe, T., Beatson, S. A., Taniguchi, H., Abe, H., Bailey, C. M., Fivian, A., Younis, R., Matthews, S., Marches, O., & Frankel, G. (2006). An extensive repertoire of type III secretion effectors in Escherichia coli O157 and the role of lambdoid phages in their dissemination. *Proceedings of the National Academy of Sciences, 103*(40), 14941-14946.

Vergunst, A. C., van Lier, M. C., den Dulk-Ras, A., Stüve, T. A. G., Ouwehand, A., & Hooykaas, P. J. (2005). Positive charge is an important feature of the C-terminal transport signal of the VirB/D4-translocated proteins of Agrobacterium. *Proc Natl Acad Sci U S A, 102*(3), 832-837.

Wan, S., Duan, Y., & Zou, Q. (2017). HPSLPred: An Ensemble Multi-Label Classifier for Human Protein Subcellular Location Prediction with Imbalanced Source. *Proteomics, 17*(17-18). doi:10.1002/pmic.201700262

Wandersman, C. (2013). Concluding remarks on the special issue dedicated to bacterial secretion systems: function and structural biology. *Res Microbiol, 164*(6), 683-687. doi:10.1016/j.resmic.2013.03.008

Wang, J., Li, J., Yang, B., Xie, R., Marquez-Lago, T. T., Leier, A., Hayashida, M., Akutsu, T., Zhang, Y., Chou, K. C., Selkrig, J., Zhou, T., Song, J., & Lithgow, T. (2019a). Bastion3: a two-layer ensemble predictor of type III secreted effectors. *Bioinformatics, 35*(12), 2017-2028. doi:10.1093/bioinformatics/bty914

Wang, J., Yang, B., An, Y., Marquez-Lago, T., Leier, A., Wilksch, J., Hong, Q., Zhang, Y., Hayashida, M., Akutsu, T., Webb, G. I., Strugnell, R. A., Song, J., & Lithgow, T. (2019b). Systematic analysis and prediction of type IV secreted effector proteins by

machine learning approaches. *Brief Bioinform, 20*(3), 931-951. doi:10.1093/bib/bbx164

Wang, J., Yang, B., Leier, A., Marquez-Lago, T. T., Hayashida, M., Rocker, A., Zhang, Y., Akutsu, T., Chou, K. C., Strugnell, R. A., Song, J., & Lithgow, T. (2018). Bastion6: a bioinformatics approach for accurate prediction of type VI secreted effectors. *Bioinformatics, 34*(15), 2546-2555. doi:10.1093/bioinformatics/bty155

Wang, J., Yang, B., Revote, J., Leier, A., Marquez-Lago, T. T., Webb, G., Song, J., Chou, K. C., & Lithgow, T. (2017). POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. *Bioinformatics, 33*(17), 2756-2758. doi:10.1093/bioinformatics/btx302

Wang, T., Si, M., Song, Y., Zhu, W., Gao, F., Wang, Y., Zhang, L., Zhang, W., Wei, G., Luo, Z. Q., & Shen, X. (2015). Type VI Secretion System Transports Zn2+ to Combat Multiple Stresses and Host Immunity. *PLoS Pathog, 11*(7), e1005020. doi:10.1371/journal.ppat.1005020

Wang, Y., Huang, H., Sun, M., Zhang, Q., & Guo, D. (2012). T3DB: an integrated database for bacterial type III secretion system. *BMC Bioinformatics, 13*(1), 66.

Wang, Y., Sun, M., Bao, H., & White, A. P. (2013a). T3_MM: a Markov model effectively classifies bacterial type III secretion signals. *PLoS One, 8*(3), e58173. doi:10.1371/journal.pone.0058173

Wang, Y., Sun, M., Bao, H., Zhang, Q., & Guo, D. (2013b). Effective identification of bacterial type III secretion signals using joint element features. *PLoS One, 8*(4), e59754. doi:10.1371/journal.pone.0059754

Wang, Y., Wei, X., Bao, H., & Liu, S. L. (2014). Prediction of bacterial type IV secreted effectors by C-terminal features. *BMC genomics, 15*(1), 50. doi:10.1186/1471-2164-15-50

Wang, Y., Zhang, Q., Sun, M. A., & Guo, D. (2011). High-accuracy prediction of bacterial type III secreted effectors based on position-specific amino acid composition profiles. *Bioinformatics, 27*(6), 777-784. doi:10.1093/bioinformatics/btr021

Welch, R. A. (2001). RTX toxin structure and function: A story of numerous anomalies and few analogies in toxin biology. *Pore-Forming Toxins, 257*, 85-111.

Welch, R. A., Dellinger, E. P., Minshew, B., & Falkow, S. (1981). Haemolysin contributes to virulence of extra-intestinal E. coli infections. *Nature, 294*(5842), 665-667.

Whitney, J. C., Beck, C. M., Goo, Y. A., Russell, A. B., Harding, B. N., De Leon, J. A., Cunningham, D. A., Tran, B. Q., Low, D. A., Goodlett, D. R., Hayes, C. S., & Mougous, J. D. (2014). Genetically distinct pathways guide effector export through the type VI secretion system. *Mol Microbiol, 92*(3), 529-542. doi:10.1111/mmi.12571

Wolpert, D. H. (1992). Stacked Generalization. *Neural Networks, 5*(2), 241-259. doi:Doi 10.1016/S0893-6080(05)80023-1

Xiao, N., Cao, D. S., Zhu, M. F., & Xu, Q. S. (2015). protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences. *Bioinformatics, 31*(11), 1857-1859. doi:10.1093/bioinformatics/btv042

Xiong, Y., Wang, Q., Yang, J., Zhu, X., & Wei, D. Q. (2018). PredT4SE-Stack: Prediction of Bacterial Type IV Secreted Effectors From Protein Sequences Using a Stacked Ensemble Method. *Front Microbiol, 9*, 2571. doi:10.3389/fmicb.2018.02571

Xue, L., Tang, B., Chen, W., & Luo, J. (2018). DeepT3: deep convolutional neural networks accurately identify Gram-Negative Bacterial Type III Secreted Effectors using the N-terminal sequence. *Bioinformatics*. doi:10.1093/bioinformatics/bty931

Yang, X., Guo, Y., Luo, J., Pu, X., & Li, M. (2013). Effective identification of Gram-negative bacterial type III secreted effectors using position-specific residue conservation profiles. *PLoS One, 8*(12), e84439. doi:10.1371/journal.pone.0084439

Yang, Y., Zhao, J., Morgan, R. L., Ma, W., & Jiang, T. (2010). Computational prediction of type III secreted proteins from gram-negative bacteria. *BMC Bioinformatics, 11 Suppl 1*, S47. doi:10.1186/1471-2105-11-S1-S47

Zeng, C., & Zou, L. (2017). An account of in silico identification tools of secreted effector proteins in bacteria and future challenges. *Brief Bioinform*. doi:10.1093/bib/bbx078

Zhang, P., Tao, L., Zeng, X., Qin, C., Chen, S. Y., Zhu, F., Yang, S. Y., Li, Z. R., Chen, W. P., & Chen, Y. Z. (2017). PROFEAT Update: A Protein Features Web Server with Added Facility to Compute Network Descriptors for Studying Omics-Derived Networks. *J Mol Biol, 429*(3), 416-425. doi:10.1016/j.jmb.2016.10.013

Zhang, Y., Xie, R., Wang, J., Leier, A., Marquez-Lago, T. T., Akutsu, T., Webb, G. I., Chou, K. C., & Song, J. (2018). Computational analysis and prediction of lysine malonylation sites by exploiting informative features in an integrative machine-learning framework. *Brief Bioinform*. doi:10.1093/bib/bby079

Zhang, Y., Yu, S., Xie, R., Li, J., Leier, A., Marquez-Lago, T. T., Akutsu, T., Smith, A. I., Ge, Z., Wang, J., Lithgow, T., & Song, J. (2019). PeNGaRoo, a combined gradient boosting and ensemble learning framework for predicting non-classical secreted proteins. *Bioinformatics*. doi:10.1093/bioinformatics/btz629

Zhao, Y., Zhang, X., Torres, V. V. L., Liu, H., Rocker, A., Zhang, Y., Wang, J., Chen, L., Bi, W., Lin, J., Strugnell, R. A., Zhang, S., Lithgow, T., Zhou, T., & Cao, J. (2019). An Outbreak of Carbapenem-Resistant and Hypervirulent Klebsiella pneumoniae in an Intensive Care Unit of a Major Teaching Hospital in Wenzhou, China. *Front Public Health, 7*, 229. doi:10.3389/fpubh.2019.00229

Zou, L., Nan, C., & Hu, F. (2013). Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles. *Bioinformatics, 29*(24), 3135-3142. doi:10.1093/bioinformatics/btt554

Zou, Q., Guo, J., Ju, Y., Wu, M., Zeng, X., & Hong, Z. (2015). Improving tRNAscan-SE Annotation Results via Ensemble Classifiers. *Mol Inform, 34*(11-12), 761-770. doi:10.1002/minf.201500031

CHAPTER 2: Computational prediction of single types of secreted substrates in Gram-negative bacteria

Bacterial secreted substrates adapt quickly to different hosts and survival strategies during evolution. Biologically, all of the substrates of a given protein secretion system are recognized by the system, meaning that there must be conserved features that can be read by the secretion system for it to choose its substrates. However, it is now clear that substrate proteins show low similarity in terms of primary structure *i.e.* sequence: without highly conserved patterns and motifs encoded in simple sequence. Whatever features are being read by the secretion machines, it is encoded in higher order structural elements such as secondary structure and/or tertiary structure information. Machine learning provides a universal framework and practical solution to predict various substrates from such "features", in a high-throughput and cost-effective manner. How secretion systems select their substrates for secretion remains unknown, but it is likely that they too use a complicated and combined strategy to recognize their substrates for translocation.

The aim of this chapter of work was to extract as much information as possible from protein sequences of the substrates in the form of different features to feed into different machine learning algorithms, to closely simulate the substrate selection by each secretion system. By exploring various ensemble strategies to integrate these features and combine a wide range of machine learning algorithms, I sought to uncover the hidden characteristics within the substrate sequences. Taking full use of the strengths and merits of the different features along with the power of machine learning algorithms, I can develop more accurate and robust substrate predictors.

This chapter presents three predictors that have been developed within a multi-layer framework to enable an accurate and streamlined prediction of three well studied types of substrates, including type III, IV and VI secreted substrates. Note that the more general term of secretion 'substrate' is used throughout the thesis, while the three published articles specifically refer to 'effectors', a common term for type III, IV and VI substrates. The three pieces of work are distributed into three subsections in chronological order as they were undertaken during my PhD candidature: Section 2.1 describes the Bastion4 predictor for the prediction of type IV secreted substrate proteins after systematic analysis of their characteristics based on multiple machine learning approaches. Section 2.2 describes the first

type VI substrate predictor Bastion6, and Section 2.3 describes the Bastion3 predictor for the prediction of type III secreted substrates, based on a two-layer ensemble strategy.

2.1

Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches

The supplementary information for this manuscript is listed in **Appendix 1**.

OXFORD

Briefings in Bioinformatics, 2017, 1–21

doi: 10.1093/bib/bbx164
Paper

Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches

Jiawei Wang,* Bingjiao Yang,* Yi An,* Tatiana Marquez-Lago, André Leier, Jonathan Wilksch, Qingyang Hong, Yang Zhang, Morihiro Hayashida, Tatsuya Akutsu, Geoffrey I. Webb, Richard A. Strugnell, Jiangning Song and Trevor Lithgow

Corresponding authors. Trevor Lithgow, Infection and Immunity Program, Biomedicine Discovery Institute and Department of Microbiology, Monash University, Melbourne, Victoria 3800, Australia. Tel.: +61-3-9902-9217; Fax: +61-3-9905-3726; E-mail: Trevor.Lithgow@monash.edu; Jiangning Song, Infection and Immunity Program, Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University, Melbourne, Victoria 3800, Australia. Tel.: +61-3-9902-9304; E-mail: Jiangning.Song@monash.edu
*These authors contributed equally to this work.

Jiawei Wang is currently a PhD candidate in the Biomedicine Discovery Institute and the Department of Microbiology at Monash University, Australia. He received his bachelor degree in software engineering from Tongji University and his master degree in computer science from Peking University, China. His research interests are computational biology, bioinformatics, machine learning and data mining.
Bingjiao Yang received his master degree at the National Engineering Research Center for Equipment and Technology of Cold Strip Rolling, College of Mechanical Engineering from Yanshan University, China. His research interests are bioinformatics, machine learning and data mining.
Yi An is currently a master student in the College of Information Engineering, Northwest A&F University, China. During her stay as a visiting student at the Biomedicine Discovery Institute and Department of Microbiology at Monash University, she undertook a bioinformatics project focused on computational analysis of bacterial secreted effector proteins. Her research interests include bioinformatics, data mining and Web-based information systems.
Tatiana Marquez-Lago is an Associate Professor in the Department of Genetics, University of Alabama at Birmingham (UAB) School of Medicine, USA. She is additionally affiliated with the UAB Comprehensive Cancer Center and the Informatics Institute. Her research interests include systems biology and biomedicine, gene expression and bioengineering, big data informatics, multiscale modeling and simulations. Her interdisciplinary laboratory studies stochastic gene expression, chromatin organization and microbiota/microbiome interactions in complex diseases.
André Leier is currently an Assistant Professor in the Department of Genetics and the Informatics Institute, University of Alabama at Birmingham (UAB) School of Medicine, USA. He is also an associate scientist in the UAB Comprehensive Cancer Center. He received his PhD in Computer Science (Dr rer. nat.), University of Dortmund, Germany. He conducted postdoctoral research at Memorial University of Newfoundland, Canada, the University of Queensland, Australia and ETH Zürich, Switzerland. His research interests are in Biomedical Informatics and Computational and Systems Biomedicine.
Jonathan Wilksch received his PhD degree in 2012 from The University of Melbourne, Australia. He is a Research Fellow in the Department of Microbiology and Immunology at the University of Melbourne, Australia. His research background and current interests include the mechanisms of bacterial pathogenesis, biofilm formation, gene regulation and host–pathogen interactions.
Qingyang Hong received his bachelor degree from Central South University, China, and his master degree in computer science from the University of Melbourne, Australia. His research interests are machine learning and data mining.
Yang Zhang received his PhD degree in Computer Science and Engineering in 2015 from Northwestern Polytechnical University, China. He is a Professor and Vice-Dean in the College of Information Engineering, Northwest A&F University, China. His research interests are big data analytics, machine learning and data mining.
Morihiro Hayashida received his PhD degree in Informatics in 2005 from Kyoto University, Japan. He is an assistant professor in Bioinformatics Center, Institute for Chemical Research, Kyoto University, Japan. His research interests include functional analysis of proteins and development of computational methods.
Tatsuya Akutsu received his Dr Eng. degree in Information Engineering in 1989 from University of Tokyo, Japan. Since 2001, he has been a professor in Bioinformatics Center, Institute for Chemical Research, Kyoto University, Japan. His research interests include bioinformatics and discrete algorithms.
Geoffrey I. Webb received his PhD degree in 1987 from La Trobe University, Australia. He is a professor in the Faculty of Information Technology and director of the Monash Centre for Data Science at Monash University. His research interests include machine learning, data mining, computational biology and user modeling.
Richard A. Strugnell undertook his PhD training at Monash University and Postdoctoral research at University of Birmingham and the Wellcome Research Laboratories in the UK, and at Monash University in Australia. He is currently Pro Vice-Chancellor (Graduate and International Research) and Professor in the Department of Microbiology and Immunology, Faculty of Medicine Dentistry and Health Sciences, the University of Melbourne. His laboratory is interested in how bacteria cause disease and what interventions can be made to stop this happening.
Jiangning Song is a senior research fellow and group leader in the Biomedicine Discovery Institute and the Department of Biochemistry and Molecular Biology, Monash University, Australia. He is affiliated with the Monash Centre for Data Science, Monash University. He is also an Associate Investigator at the ARC Centre of Excellence in Advanced Molecular Imaging, Monash University. His research interests include bioinformatics, systems biology, machine learning, functional genomics and enzyme engineering.
Trevor Lithgow received his PhD degree in 1992 from La Trobe University, Australia. He is an ARC Australian Laureate Fellow in the Biomedicine Discovery Institute and the Department of Microbiology at Monash University, Australia. His research interests particularly focus on molecular biology, cellular microbiology and bioinformatics. His laboratory develops and deploys multidisciplinary approaches to identify new protein transport machines in bacteria, understand the assembly of protein transport machines and dissect the effects of anti-microbial peptides on anti-biotic resistant 'superbugs'.
Submitted: 20 August 2017; **Received (in revised form):** 8 November 2017

Abstract

In the course of infecting their hosts, pathogenic bacteria secrete numerous effectors, namely, bacterial proteins that pervert host cell biology. Many Gram-negative bacteria, including context-dependent human pathogens, use a type IV secretion system (T4SS) to translocate effectors directly into the cytosol of host cells. Various type IV secreted effectors (T4SEs) have been experimentally validated to play crucial roles in virulence by manipulating host cell gene expression and other processes. Consequently, the identification of novel effector proteins is an important step in increasing our understanding of host–pathogen interactions and bacterial pathogenesis. Here, we train and compare six machine learning models, namely, Naïve Bayes (NB), *K*-nearest neighbor (KNN), logistic regression (LR), random forest (RF), support vector machines (SVMs) and multilayer perceptron (MLP), for the identification of T4SEs using 10 types of selected features and 5-fold cross-validation. Our study shows that: (1) including different but complementary features generally enhance the predictive performance of T4SEs; (2) ensemble models, obtained by integrating individual single-feature models, exhibit a significantly improved predictive performance and (3) the 'majority voting strategy' led to a more stable and accurate classification performance when applied to predicting an ensemble learning model with distinct single features. We further developed a new method to effectively predict T4SEs, Bastion4 (Bacterial secretion effector predictor for T4SS), and we show our ensemble classifier clearly outperforms two recent prediction tools. In summary, we developed a state-of-the-art T4SE predictor by conducting a comprehensive performance evaluation of different machine learning algorithms along with a detailed analysis of single- and multi-feature selections.

Key words: type IV secreted effector; bioinformatics; sequence analysis; comprehensive performance evaluation; machine learning; feature analysis

Introduction

Pathogenic bacteria are microorganisms that cause infections. During this process, bacteria invade a host organism where they multiply, producing and secreting effector proteins. Such effector proteins fulfill a range of functions critical for the virulence of the pathogen, that is the degree of damage that the bacterium causes to the host. In most cases, effector proteins are directly injected into host cells via dedicated secretion systems, enabling them to modulate or manipulate a wide range of cellular processes, including actin dynamics (e.g. Beps secreted by *Bartonella* spp.) [1–3], phagocytosis (e.g. various effectors of *Yersinia* and *Salmonella enterica*) [4, 5], endocytic trafficking (e.g. effectors of *Legionella pneumophila*) [6–8], apoptosis (e.g. *Shigella* effectors IpgD and OspG) [9, 10], immune response (YopJ from *Yersinia enterocolitica*) [4, 11] and secretion (e.g. *Escherichia coli* effector EspG) [12].

Currently, Gram-negative bacterial secretion systems are classified into six types (I-VI) [13]. Among them, type III and type IV secretion systems (T3SS and T4SS, respectively) and their associated effectors (T3SEs and T4SEs, respectively) have been widely studied, as they are critical for virulence of various human pathogens. For example, *S. enterica*, *Yersinia pestis* and *Pseudomonas syringae* use type III secretion systems [14], while *Brucella* spp., *Bartonella* spp., *Helicobacter pylori* and *L. pneumophila* use T4SSs [15]. Despite their clinical significance, a fundamental biological question remains: How does a given secretion system recognize a given effector protein as a substrate, which it must bind and secrete? These secretion systems are highly selective nanomachines, and do not inadvertently secrete non-effector proteins. Clearly, some element or elements of effector protein sequence and/or structure must dictate recognition by the cognate secretion system, but there is an outstanding need for an integrative understanding what these recognition elements are and how they determine substrate protein secretion. While specific wet-lab experimental studies can answer underlying questions for individual effector proteins, bioinformatics-based tools are needed to address the matter more efficiently and comprehensively.

Recently, machine learning algorithms were introduced to predict T4SEs [16–18]. For instance, Burstein *et al.* [16] developed a machine learning model for differentiating T4SEs from non-effectors in *L. pneumophila*. Their model used seven types of features including 'taxonomic distribution among bacteria and metazoa', 'sequence similarity to known effectors' and 'homology to known eukaryotic proteins', which the authors concluded from their analysis were the three best representative features [16]. To examine the classification performance of different algorithms, they used support vector machine (SVM), multilayer perceptron (MLP), Naïve Bayes (NB), Bayesian networks (BNs) and a Voting Algorithm, the latter of which was based on the former four classifiers. The study successfully predicted and experimentally verified 40 novel T4SEs from *Legionella*. In another recent work, Zou *et al.* [17] developed an SVM-based classifier called T4EffPred using four distinct feature types, including amino acid composition (AAC) and position-specific scoring matrix (PSSM), as well as feature combinations. T4EffPred could distinguish IVA and IVB effectors, which are the two main subtypes of T4SEs [17]; it has also been successfully applied to perform genome-wide predictions of effectors in the bacterium *Bartonella henselae*, where ∼50 putative T4SEs were found. In a third study, Wang *et al.* [18] presented a T4SE inter-species cross-prediction tool based on C-terminal features, such as AACs, motifs, secondary structures (SSs) and solvent accessibility (SA). The tool comprises three computational models that were trained using SVM-based machine learning (T4SEpre_psAac, trained using position-specific, sequence-based AACs; T4SEpre_bpbAac, trained using AACs based on bi-profile Bayes feature extraction combined with SVM; T4SEpre_Joint, trained using position-specific AACs, SSs and SA). When applied to the genome of *H. pylori*, 25 candidate T4SEs were identified by the authors. Also based on C-terminal Signals, Zou *et al.* [19] analyzed the performance of C-terminal sequence features such as AAC and position-specific amino acid composition (PSAac). They used multiple machine learning algorithms to train models of T4SEs with a majority vote strategy. Based on their findings, an SVM predictor of type IV-B

effectors trained with PSAac and AAC was developed and validated through a genome-scale prediction in *Coxiella burnetii*. Our previous work [20] comprehensively reviewed the currently available bioinformatics approaches for T4SE prediction, and offered an assessment of these approaches in terms of software utilities and prediction performance. A recent review from Zeng *et al.* [21] further discussed and highlighted some potential improvements of the prediction performance after benchmarking the available identification tools of secreted effector proteins in bacteria. The schematic figures in such article give a bird's-eye view of computational toolkits in the field of secreted effector predictions.

While previous work has demonstrated that machine learning approaches can successfully predict effector proteins, the features or combinations of features that are most appropriate for efficient T4SE prediction have not been systematically assessed. Here, we used 10 types of features and 6 different machine learning algorithms to train predictors with 390 T4SS effectors and 1112 non-effectors. We first compared the 10 types of features with their combinations on multiple performance assessments and found that, while combinations of features in a single model do not yield statistically significant improvements, the ensemble of multiple individual models trained with different single features significantly improved the overall performance. Our direct comparison of six representative models, namely, NB, *K*-nearest neighbor (KNN), logistic regression (LR), random forest (RF), SVM and MLP, shows that RF and SVM outperformed all others in terms of predictive and computational performance. In addition, the ensemble model that integrated all six machine learning methods further improved the prediction performance. With this valuable knowledge, we developed Bastion4, an online T4SE predictor that operates as an ensemble classifier based on six machine learning models, each of which consists of individual models trained with various types of selected features. Our subsequent analysis presented here shows that Bastion4 outperforms T4Effpred and T4SEpred based on independent tests. Bastion4 is available at http://bastion4.erc.monash.edu/.

Materials and methods

The Bastion4 methodology development (Figure 1) involved five major stages: Data set Curation, Feature Extraction, Feature Selection, Model Training and Validation and Prediction. Each of these major stages is described in the following sections.

Data sets collection

The input data set consisted of two parts: the training data set and the independent data set. We constructed the training data set by extracting known T4SEs from independent data sets described in the literature. Specifically, 347 T4SE sequences were extracted from the T4SEpre data set constructed by Wang *et al.* [18]. The pathogen *B. henselae* has two subtypes of T4SS (IVA and IVB), and 340 effectors including 30 IVA proteins and 310 IVB proteins were acquired from Zou *et al.* [17]. Finally, we added 120 proteins identified by Burstein *et al.* [16]. For the negative training set, we chose the entire set of 1132 non-effectors in Zou *et al.* [17]. After forming the preliminary data set, CD-HIT [22] was used to remove highly homologous sequences (defined as having 60% sequence identity) to reduce sequence redundancy, which may otherwise lead to a potential bias in the trained models. The final training data set contained 390 positive and 1112 negative sequences.

To evaluate the model performance in comparison with existing T4SE prediction tools, we generated an independent data set containing both positive and negative samples. For the former, 43 positive samples were acquired from the UniProt Database [23] and Meyer *et al.* [24], while for the latter, we used 150 samples from the data set of *Vibrio parahaemolyticus* serotype O3: K6 (strain RIMD 2210633) [25]. After removal of duplicate samples, which appear in our training set and the data sets used by the existing T4SE predictors, we obtained a final independent data set made up of 30 positive and 150 negative samples.

Feature extraction

The variety of features used in this work can be categorized into three main types: local sequence encoding, global sequence encoding and structural descriptor encoding. Extracted from the type-specific information available for any given protein, each feature is represented by a number of encoding vectors.

Local sequence encoding

Feature associated with local sequence encoding refers to distinguishable patterns in the protein sequence.

(1) Amino acid composition

AAC is represented as a 20-dimensional feature vector, in which each element characterizes the frequency of an amino acid type in the whole protein sequence [26].

Each element in this feature vector was calculated according to the following formula:

$$v_i = \frac{c_i}{len(seq)}, i = 1, \ \ldots, \ 20,$$

where c_i is the number of occurrences of amino acid i in the whole protein sequence, and $len(seq)$ is the length of the sequence. Finally, v_i represents the i-th element in the feature vector, which indicates the frequency of the amino acid i in the protein sequence.

(2) Dipeptide composition

A protein's dipeptide composition (DPC) is encoded in a 400-dimensional feature vector $\{fp_1, \ fp_2, \ \ldots, fp_{400}\}$, which represents the frequencies of all 400 possible amino acid pairs in the protein sequence. Each element fp_i is obtained using the following formula:

$$fp_k = \frac{p_i}{len(seq) - 1}, i = 1, \ 2, \ldots, \ 400,$$

where p_i denotes the number of occurrences of the i-th amino acid pair [17], and $len(seq) - 1$ refers to the total number of dipeptides in the whole sequence.

(3) Composition of *k*-spaced amino acid pairs

As a widely used feature type in sequence analysis [27, 28], the composition of k-spaced amino acid pairs (CKSAAPs) is in effect a generalization of the DPC. Two amino acids form a k-spaced amino acid pair if they have k amino acids in-between them. In this sense, amino acid pairs in the DPC can be viewed as 0-spaced amino acid pairs in the CKSAAP. For CKSAAP, all pairs with space $\leq k$ are considered. Thus, CKSAAP outputs a $400 \times (k+1)$-dimensional feature vector for a given protein sequence. We use $k = 5$, and, consequently, a 2400-dimensional vector is constructed.

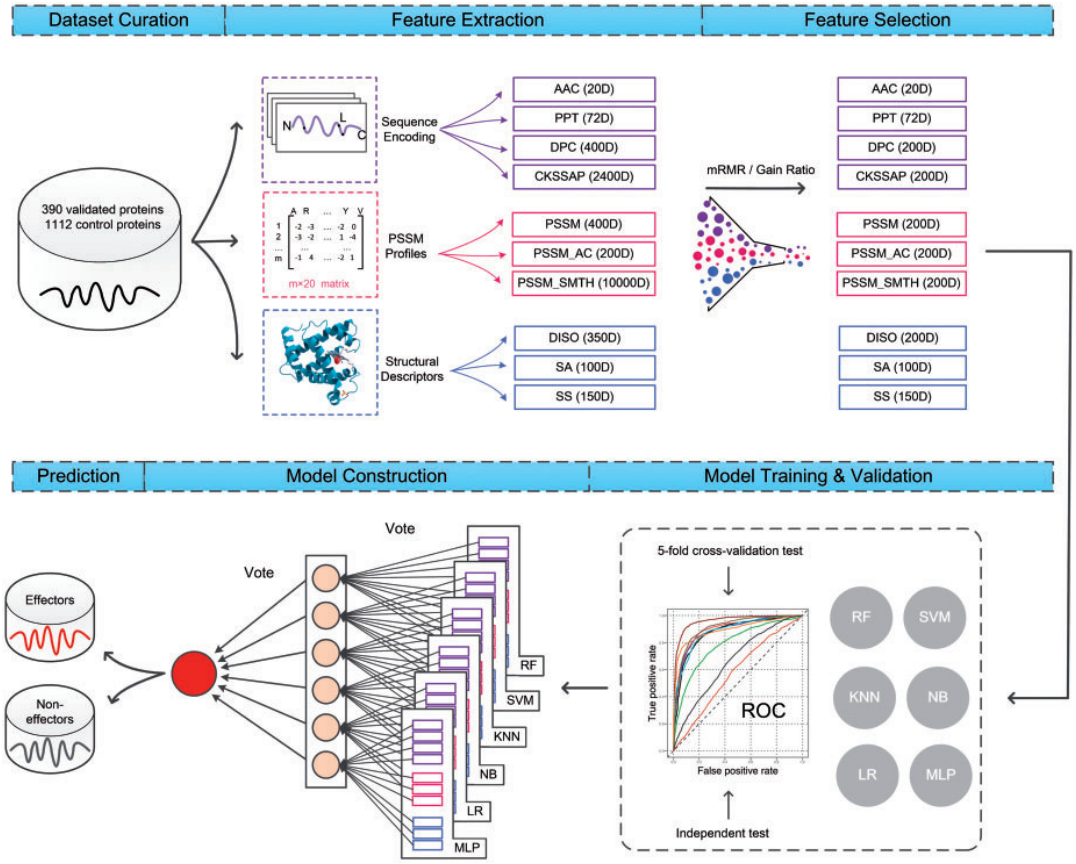


Figure 1. Overview of the proposed methodology for predicting T4SEs. First, a large number of protein sequences are collected, forming the input data set. Then, 10 types of features are extracted that characterize those proteins in different ways. Using the mRMR/Gain Ratio technique, a subset of features is selected to optimize the following model training. Next, the performance of trained models is evaluated by a 5-fold cross-validation test and an independent test. Finally, by applying a voting mechanism to various models, an ensemble classifier is formed, which separates the input into putative effectors and non-effectors.

(4) Property composition

The property composition (PPT) [29] maps amino acids to three distinct amino acid alphabets, namely, the classical amino acid alphabet, the amino acid property alphabet and the hydrophobic/hydrophilic alphabet. Each amino acid corresponds to a certain property class. When an amino acid fits to multiple property classes, it was categorized into the most specific (smallest) class. For each property class, di- and tripeptides were measured in terms of frequency. Moreover, only the features that occur more than one time in both positive and negative data sets were selected to avoid over-fitting. Consequently, a 72-dimensional feature vector is formed for each protein sequence.

Global sequence encoding

PSSMs have proved beneficial for incorporating evolutionary information in machine learning methods [17, 30–35]. Here, we generated PSSM profiles by running PSI-BLAST against the nonredundant database of NCBI with parameters $j = 3$ and $h = 0.001$. There are two types of methods for exploiting patterns from PSSM profiles, which are explained below.

(5) PSSM profiles with auto covariance transformation

A PSSM is a $L \times 20$ matrix, where L is the length of the corresponding protein sequence. The (i, j)-th element of the matrix denotes the probability of amino acid j to appear at the i-th position of the protein sequence. The PSSM encoding converts the PSSM profile into a 20×20 matrix by summing up all rows of the same amino acid residue [34], thereby forming a 400-dimensional vector as part of the input for model training.

Based on the original $L \times 20$ matrix, the PSSM_AC encoding uses the auto covariance (AC) transformation to further measure the correlation between two properties [17, 36] by using the following formula:

$$AC(j, lg) = \sum_{i=1}^{L-lg} (S_{i,j} - \bar{S}_j)(S_{i+lg,j} - \bar{S}_j)/(L - lg),$$

where j refers to one of the 20 amino acids, L denotes the length of the whole protein sequence, $S_{i,j}$ denotes the PSSM score of amino acid j at position i and $\bar{S}_j$ is the average score for amino acid j along the whole sequence:

$$\bar{S}_j = \sum_{i=1}^{L} S_{i,j}/L.$$

Consequently, the number of AC components amounts to $20 \times LG$, where lg runs from $1, 2, \ldots, LG$, with $LG < L$. Here, we set $LG = 10$ as previously used in Zou *et al.* [17], yielding a 200-dimensional feature vector.

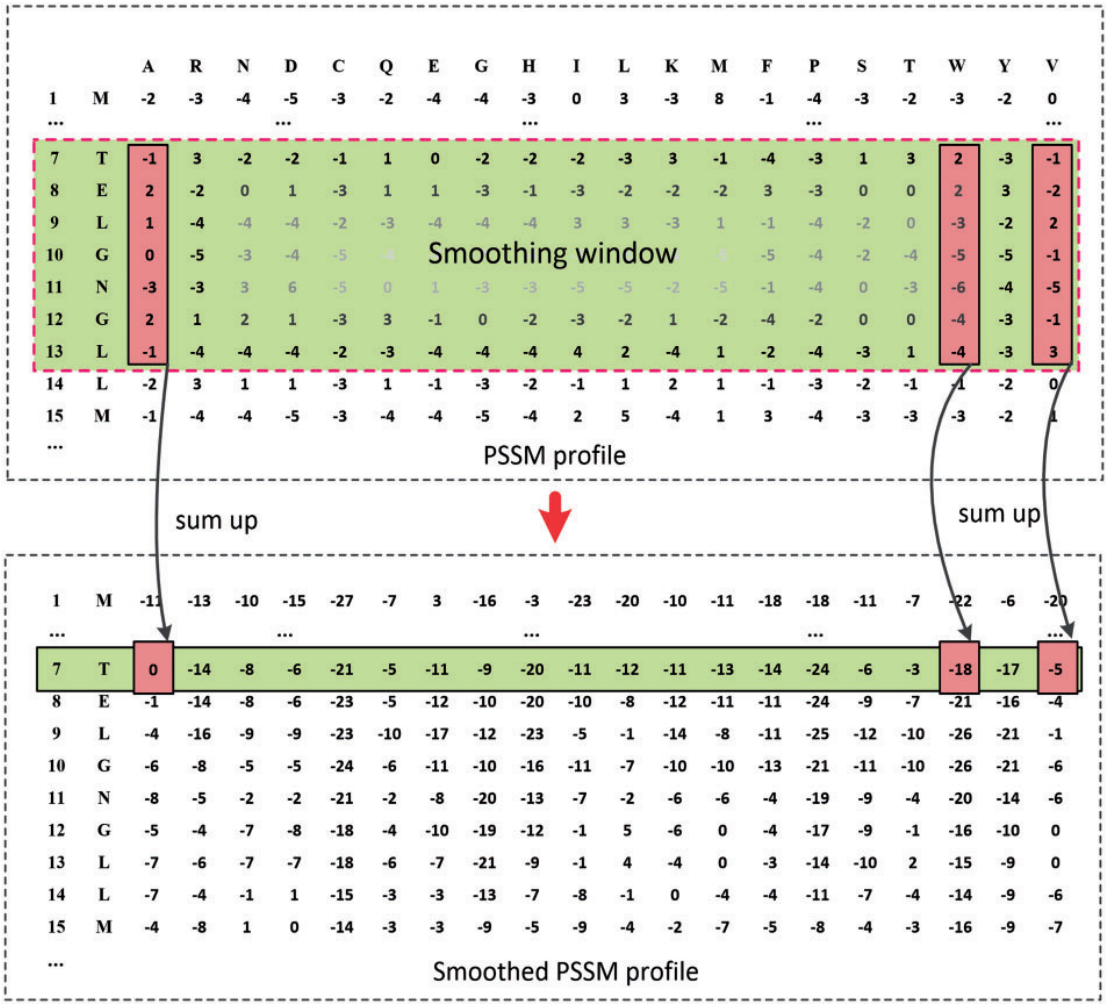


Figure 2. Example of a PSSM_SMTH profile using a smoothing window of size 7. The PSSM profile shows the evolutionary information extracted from the PSSM file, which is generated by PSI-BLAST. When the size of the smoothed window is set to 7, the values of the 7th row in the PSSM_SMTH profile are equal to the sum of the corresponding values from the 7th row to the 13th row of the PSSM file.

(6) Smoothed PSSM encoding

A transformation of the standard PSSM profile, the Smoothed PSSM (PSSM_SMTH) encoding, has been previously used to predict RNA-binding sites of proteins [37] and drug-binding residues [38]. Assuming the size of a smoothing window is w and v_i represents the ith row vector of the PSSM, each row vector of the PSSM_SMTH can be constructed by summation of the current row vector and the following $w-1$ row vectors (Figure 2):

$$v_{smoothed\_i} = v_i + \cdots + v_{i+(w-1)}.$$

For this method, we use values of w ranging from 1 to 10. Therefore, 10 PSSM_SMTH profiles are obtained. For each PSSM_SMTH profile, rows corresponding to the first 50 amino acids starting from the protein's N-terminus are considered to form a vector with dimension $50 \times 20 = 1000$. As a result, a 10 000-dimensional vector is constructed.

To extract the PSSM_AC encoding and the 10 PSSM_SMTH encodings, we used the POSSUM server, which is a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles [39].

Structural descriptor encoding

Protein structural information has been widely used to improve the prediction performance in a number of bioinformatics applications [40–45], but has not been comprehensively analyzed for the prediction of T4SEs. In our machine learning framework described here, we extract SS, SA and natively disordered region information for T4SE sequences and use them as features for model training.

(7) Predicted SS

Protein SS is a widely used attribute in bioinformatics predictors. As the SS is known for only a relatively small number of proteins, we instead predicted protein SSs from amino acid sequences using SSpro [46]. Specifically, for each residue of the query sequence, SSpro predicts one of three types of SS: alpha-helix, beta-strand or coil. Here, we represent these types of predicted SS by using a 3-bit encoding and encode the first 50

residues of the queried sequence, thereby forming a vector of length $3 \times 50 = 150$.

(8) Predicted SA
SA is another important feature for prediction. The SSpro program can be used to predict SA from protein sequence data. For each residue in a sequence SSpro predicts, it being in one of the two possible states 'exposed' or 'buried'. Therefore, we use a 2-bit encoding to represent predicted SA and encode the first 50 residues of the queried sequence, forming a vector of length $2 \times 50 = 100$.

(9) Predicted natively disordered region
Disordered (DISO) protein regions lack fixed tertiary structure, being either fully or partially unfolded [47]. Contrary to initial concerns that these regions were functionally 'useless', recent studies indicate that such regions are commonly involved in many biological functions [47]. Here, we predict the native disorder information using DISOPRED2 [48], which provides a quantitative real-valued score ranging from 0 to 1, which represents the probability of a residue being disordered. For this structural descriptor, we used seven different sizes of smoothing windows as previously suggested [49] ($w = 1, 7, 11, 21, 27, 31, 41$) to encode the first 50 residues of the queried sequence, resulting in a feature vector of length $7 \times 50 = 350$.

Feature normalization

After feature extraction, we found that some features have values ranging between 0 and 0.01, while others have values ranging from 1 to 1000. However, features that can frequently assume larger numeric values are also more likely to have a larger impact on the prediction as compared with features with ranges of smaller numeric values. Thus, to improve the prediction accuracy and avoid having a particular feature dominating the prediction because of it assuming larger numerical values, we normalize values of different features so that all values fall into the same numeric interval [50].

Here, we use the following formula to normalize all feature values to the numeric interval [0, 1]:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}},$$

where x, x_{min}, x_{max} denote the original value, the minimum value and the maximum value in the feature vector, respectively, and x' denotes the output value of x after scaling. If the numbers in a feature vector are equal to each other, i.e. $x_{max} - x_{min} = 0$, we assign the value 0.

Feature selection

Feature selection plays an important role in machine learning. Biological data sets are usually characterized by a large number of initial features, making it a formidable task to deal with oversized feature sets; some of the typical problems include slow algorithm speed and a low predictor performance. Thus, the objective of feature selection is 3-fold: improving the prediction performance of the predictors, providing faster and more cost-effective predictors and providing a better understanding of the underlying process that generated the data [51].

Gain ratio
The gain ratio algorithm is a powerful method based on information theory [52]. In this binary classification problem, we assume the probability of a positive sample to be P and the probability of a negative sample to be $1 - P$. The entropy of the classification can be denoted as:

$$H(C) = -P\log_2 P - (1-P)\log_2(1-P),$$

where C denotes the positive class label. The conditional entropy of the feature F_j can be calculated as follows:

$$H(C|F_j) = \sum_{j=1}^{m} P_{F=F_j} H(C|F = F_j),$$

where m denotes the total number of features. Therefore, we can express the formula of gain ratio as:

$$GR(F_j) = \frac{H(C) - H(C|F_j)}{H(C)}.$$

mRMR
The mRMR algorithm is based on mutual information [53]. It was originally proposed by Peng *et al.* [53] and can be downloaded from http://penglab.janelia.org/proj/mRMR/. The mRMR algorithm has been widely used in a number of feature-selection tasks in many research areas [54–59], including protease cleavage sites prediction, acetylation site prediction and other posttranslational modification site predictions.

Model training

Naive Bayes
NB is a commonly used statistical classifier that is generally adopted to calculate the conditional probability without assuming any dependence between features. It has been successfully applied in many disciplines of science, and performs consistently well even when considering relatively few attributes [60]. NB operates based on the Bayes' theorem:

$$p(C|F_1, F_2, \ldots, F_n) = \frac{p(C)}{p(F_1, F_2, \ldots, F_n)} \prod_{i=1}^{n} p(F_i|C),$$

where C represents the binary class variable, and F denotes the input feature vector of the classifier.

K-nearest neighbor
KNN is a simple but powerful classification method, which predicts a new candidate by evaluating the distance functions to k nearest known neighbors. It has been successfully used in many bioinformatics endeavors such as the prediction of protein function [61], protein subcellular localization [62] and membrane protein architecture [63]. According to the KNN algorithm, a new instance is classified by a majority vote of its top KNNs. The instance is then assigned to the most common class among the top KNNs.

The choice of parameter k is important, and has a direct effect in the performance and outcome of a KNN classifier. In this work, k was optimized so as to minimize the classification error for values $k = 1, 2, \ldots \left| max\ \{\sqrt{\text{featureNum}},\ \text{featureNum}/2\} \right|$, where featureNum is the number of features used during model training.

Logistic regression
As a widely used algorithm [64, 65], LR results from a linear regression using the following equation:

$$p(y) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}},$$

where $p(y)$ refers to the expected probability of dependent variables, and β_0, β_1 are constants.

As the values of LR range from 0 to 1, it is a useful technique for handling classification problems, especially in situations where only the probability of occurrence of the response is concerned.

Random forest

The RF algorithm is a classification algorithm developed by Leo Breiman [66] using an ensemble of classification trees. It has been widely used and implemented as the RF package in R [67]. RF is one of the most powerful algorithms in machine learning [68]. In RF, two key parameters are the number of the trees, *M*, and the number of features selected randomly, *mtry*.

Here, we selected $M = 1000$, and optimized the parameter *mtry* over the set of integers between 1 and $\left\lfloor max\ \{\sqrt{\text{featureNum}},\ \text{featureNum}/2\} \right\rfloor$ to minimize the classification error. Here, featureNum is the total number of features.

Support vector machine

SVM is a powerful machine learning algorithm and is commonly used to deal with binary classification problems. SVM has been widely applied to solve many classification and regression problems in bioinformatics and computational biology [17, 18, 26, 27] and, particularly, SVM with a Gaussian radial basis kernel is widely used for nonlinear classification problems. There are two parameters that affect the performance of the nonlinear SVM model: Cost (*C*), which controls the cost of misclassification during data training, and Gamma (γ), which is the free parameter of the Gaussian radial basis function.

In this study, we adopt the radial basis kernel for SVM model training by using the e1071 package [69] in R language. We use the grid search method to identify the optimal parameters $C \in \{2^{-6}, 2^{-5}, \ldots 1, \ldots, 2^5, 2^6$ and $\gamma \in \{2^{-6}, 2^{-5}, \ldots 1, \ldots, 2^5, 2^6$. Accordingly, our number of grid points is $13 \times 13 = 169$. Based on the training data, the SVM is optimized by finding the optimal values for C and γ that minimize the classification error by performing 10-fold cross-validation.

Neural networks

A neural network is a nonlinear statistical classifier that is able to detect complex relationships between dependent and independent variables [70]. One type of neural network is called MLP. An MLP is characterized by multiple layers, that is there can be one or more nonlinear layers (hidden layers) between the input and the output layers. An increase in the number of hidden layers facilitates neural network models to solve increasingly nonlinear problems.

Using RSNNS [71], an R implementation of SNNS [72], we train an MLP classifier with two hidden layers. The numbers of nodes in the first and second hidden layers are set to 64 and 32, respectively, while the maximum number of iterations to learn is set to 1000.

Randomized 5-fold cross-validation test

Cross-validation is a common method for estimating the performance of a classification model. In this study, the benchmark data set is randomly partitioned into five equal-sized subsets, and tests are repeated five times. For each cross-validation test, one subset is used as testing data, while the remaining four subsets form the training set are used to train the classifier. Hence, each subset is used once for testing and four times for training. The five numerical results obtained from these tests are averaged to obtain a single value that represents the performance of the classification model.

Independent test

In this study, we compare the performance of our models with three previously published classifiers: T4Effpred [17] and two variant models of T4SEpred (i.e. T4SEpred_bpbAac and T4SEpred_psAac) [18]. As noted earlier, we constructed an independent test data set, which is completely different from the training data sets of these three models. Performance comparison is conducted on this independent data set.

Performance assessment

Six performance measures, namely, Sensitivity (SN), Specificity (SP), Precision (PRE), Accuracy (ACC), *F*-value and Matthew's correlation coefficient (MCC) [73], are used to evaluate the overall predictive performance of classification models. These measures are defined as:

$$\text{SN} = \frac{TP}{TP + FN}$$

$$\text{SP} = \frac{TN}{TN + FP}$$

$$\text{PRE} = \frac{TP}{TP + FP}$$

$$\text{ACC} = \frac{TP + TN}{TP + FP + TN + FN}$$

$$F - \text{score} = 2 \times \frac{TP}{2TP + FP + FN}$$

$$MCC = \frac{(TP \times TN) - (FN \times FP)}{\sqrt{(TP + FN) \times (TN + FP) \times (TP + FP) \times (TN + FN)}},$$

where TP, TN, FP and FN represent the numbers of true positives, true negatives, false positives and false negatives, respectively.

Additionally, the receiver operating characteristic (ROC) curve, which is a plot of the true-positive rate versus the false-positive rate, is depicted to visually measure the comprehensive performance of different classifiers. The area under the curve (AUC) is also provided in each of the ROC plots, to quantify the respective performance.

Results and discussion

Sequence analysis

We analyzed the amino acid occurrences (including those over-represented and underrepresented) on each position of T4SS effectors. We examined the first 50 N-terminal and 50 C-terminal positions of sequences of T4SEs [18, 20], non-effectors and control proteins with the pLogo program [74], and studied the differences among the three groups of proteins with respect to their amino acid preferences (Figure 3).

For the N-terminus, remarkable consensus was found in T4SE sequences, while amino acid residues tended to be more disordered in non-T4SE and control sequences. Specifically, the

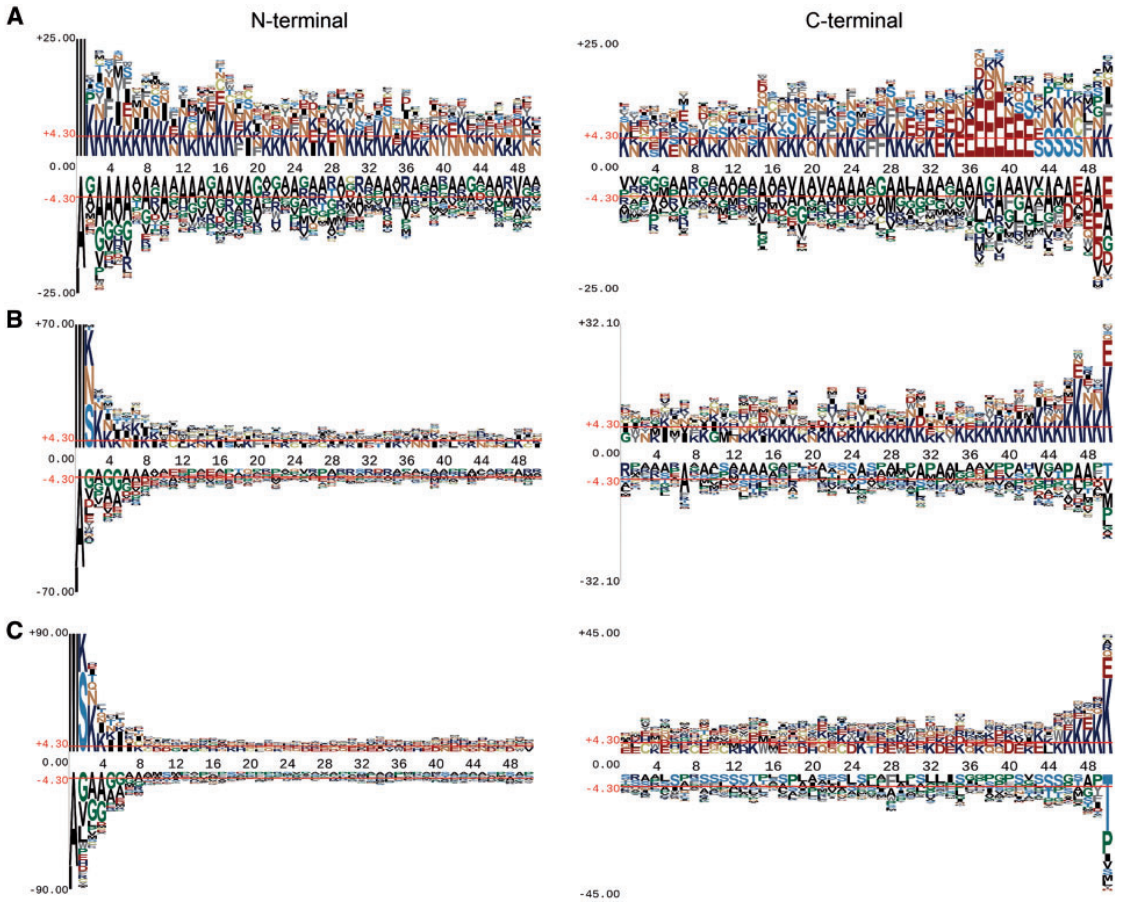


Figure 3. Position-specific amino acid sequence profiles of T4SEs and non-effectors for N- and C-terminal 50 positions. Images were generated by pLogo. The vertical axis denotes the log-odds binomial probability, while the horizontal one represents the N-terminal position number. The red horizontal bars on the images denote the statistical significant thresholds ($P = 0.05$) following a Bonferroni correction. (**A**), (**B**) and (**C**) illustrate sequence logo representations for T4SEs, non-effectors and control effectors (i.e. cytoplasmic proteins), respectively.

N-terminal sequences of T4SEs showed a significant overrepresentation of lysine and asparagine residues, with glycine and alanine largely absent. Likewise, the C-terminal sequences showed an enrichment for glutamate residues at positions 35–42 for the T4SEs (i.e. in residues located at −16 to −8 positions relative to the C-terminus). There was no significant motif pattern in the C-terminal sequences of non-T4SEs or the control sequences. Such characteristic features distinguish T4SEs from non-effectors, and are useful for explaining protein features that might be captured in machine learning models. Previous work on several specific T4SEs has shown that the C-terminal segment of the proteins incorporates at least part of the signal for engagement by the T4SS [75].

As shown in Table 1, *L. pneumophila* has 291 T4SEs, thereby accounting for the largest proportion (74.6%) of T4SEs. To address whether this biases the outcomes of putative signal sequence motifs, we analyzed sequences from *L. pneumophila* and *C. burnetii*, respectively (Figure 4). The enrichment of glutamate ('E') residues is clear in sequences from *L. pneumophila*. While sequences from *C. burnetii* commonly have glutamic residues, these have a much reduced preference. Biologically, this could indicate two distinct targeting signals, with the one composed of glutamic residues being the predominant form in species, such as *L. pneumophila*, but with this glutamic acid-rich signal used by fewer of the T4SEs in species like *C. burnetii*. Computationally, this finding reveals that there is no common motif in T4SS effectors across multiple species, which further supports the need to look at many features to develop globally effective machine learning models.

Performance evaluation using randomized 5-fold cross-validation tests

For each of the 10 feature encodings, all six classifiers were trained and validated to predict T4SEs based on a randomized 5-fold cross-validation test. As negative samples, 390 protein sequences were randomly selected from the non-type IV effector data set, to generate a balanced training data set with a 1:1 ratio of positive to negative samples. All experiments were repeated five times. The results are documented in Table 2, Figures 5 and 6, and discussed below.

Performance evaluation of various classifiers

For most of the feature encodings, RF and SVM predictors clearly outperformed the other classifiers in terms of ACC, *F*-score and MCC (Table 2, Figures 5 and 6). This observation is

Table 1. The components of various species in T4SEs

Species	Number
Agrobacterium rhizogenes	4
Agrobacterium tumefaciens str. C58	2
Agrobacterium tumefaciens	4
Anaplasma marginale str. St. Maries	3
Anaplasma phagocytophilum HZ	2
Bartonella grahamii as4aup	1
Bartonella henselae str. Houston-1	5
Bordetella pertussis Tohama I	4
Brucella melitensis biovar Abortus 2308	6
Brucella melitensis bv. 1 str. 16M	2
Coxiella burnetii CbuG_Q212	**1**
Coxiella burnetii CbuK_Q154	**3**
Coxiella burnetii Dugway 5J108-111	**7**
Coxiella burnetii RSA 331	**15**
Coxiella burnetii RSA 493	**34**
Ehrlichia chaffeensis str. Arkansas	1
Helicobacter pylori 26695	1
Legionella pneumophila subsp. *pneumophila* str. Philadelphia 1	**35**
Ochrobactrum anthropi ATCC 49188	1
Legionella pneumophila subsp. *pneumophila* (strain Philadelphia 1/ATCC 33152/DSM 7513)	**256**
Unknown	3
Total	390

The top two species with the largest numbers of known T4SEs are highlighted in bold.

consistent with and supports the conclusion drawn by Fernández-Delgado *et al.* [68], who found that RF and SVM are most likely the best classifiers among all compared 17 machine learning algorithms based on 112 different data sets. Among all the classifiers corresponding to various feature encoding methods, RF classifiers achieved the highest *F*-score (0.905) and MCC (0.811) when PSSM was used for training.

To make a fair performance comparison of a variety of different classifiers, the trade-off between SN and SP was taken into consideration. The difference between SN and SP for RF models, in most cases, is lower than for other models. This implies that the RF classifier provides a better trade-off between SN and SP, and achieves a more comprehensive and stable performance on the prediction of T4SEs. As an ensemble classifier, RF can even fit training data that suffers heavily from noisy, high-dimensional and highly correlated features without over-fitting [76].

To evaluate the computational efficiency of various classifiers, we compared the computational time for each classifier, using 200-dimensional PSSM features (selected by GainRatio) for model training. The total computational time for each classifier included parameter tuning time (Tuning time) and randomized 5-fold cross-validation time (CV time). As can be seen in Figure 7A, SVM and MLP were most time-consuming among all methods in terms of the total computational process, which consists of parameter tuning and model training. Parameter tuning for SVM was computationally costliest (Figure 7C), highlighting difficulties associated with optimizing parameters for SVM models. In contrast, training MLP model (without performing parameter optimization in advance, which is another extremely complex task)-associated cross-validations are most time-consuming. Finally, when compared with SVM, RF achieved a better trade-off with remarkably less tuning time and only slightly longer CV time (Figure 7B and C).

Performance evaluation of various feature encoding schemes

Among all feature encoding schemes, the most powerful one is PSSM (Table 2), achieving the highest AUC values for five of six classifiers when compared with other feature encodings (Figure 6). The local sequence encoding and global sequence encoding (except for PSSM) achieved similar performances, while the structural descriptor encoding showed a poor performance (Figure 6). CKSAAP performed worse than DPC for most classifiers (Table 2 and Figure 6): a possible explanation is that DPC might recognize the most valuable patterns in protein sequences, while CKSAAP may introduce redundant and noisy information that reduce the performance of T4SE prediction.

We explored the contribution of all features and three distinctive groups of them (AAC group, PSSM group and structure group) in two ways: feature ensemble and feature combination. For feature ensemble, we trained single-feature models and then integrated these as an ensemble model. For feature combination, features were first combined into a vector to train a model.

As shown, for each machine learning method, models trained based on all features and the three distinctive groups using feature ensemble (Figure 6 and Table 2) outperformed those trained using feature combination (Supplementary Figure S1 and Supplementary Table S1). When compared with single feature-based models, feature ensemble models achieved more stable performance across various machine learning methods (Figure 6 and Table 2).

Performance evaluation of feature selection methods

To remove redundant features and properly characterize feature importance, we conducted feature selection experiments (Figure 8). For different feature encodings, models trained using GainRatio-selected features (such as the top 50, 100, 150, 200, 250, 300 and 350 features) generally resulted in a comparable or, in some cases, better performance compared with models trained using all original features (Figure 8A). This finding indicates that the most discriminative features from the original set could be extracted to form a subset that preserved the original semantics of the variables. Owing to the removal of noisy features, a selected feature set is also likely to be better modeled and interpreted by machine learning methods [77]. It is also advantageous to use selected feature sets, which can help significantly reduce the computational time during model training. This is especially so for feature encodings with a large number of features (such as PSSM_SMTH). By using the mRMR feature selection, we obtained similar results as with GainRatio (Figure 8B). It is noteworthy that mRMR failed to recognize an informative feature set for PSSM_SMTH encoding, leading to a decreased performance after feature selection as compared with the full original feature set. A side-by-side performance comparison of GainRatio and mRMR revealed that, overall, GainRatio achieved a more stable performance (Figure 8C).

Performance comparison of models trained using individual feature types versus feature combinations

Although previous studies have used a combination of features to train prediction models [17, 18, 28], our experiments indicate that simply combining features did not help in further enhancing the model performance. Classifiers trained with different combinations of feature types did not show improved performances, compared with the model trained using PSSM feature encoding only (Figure 9). There are possible reasons for this. As the PSSM features dominate others for T4SE prediction [17] (also refer to Figure 6, Table 2), the performance of a feature

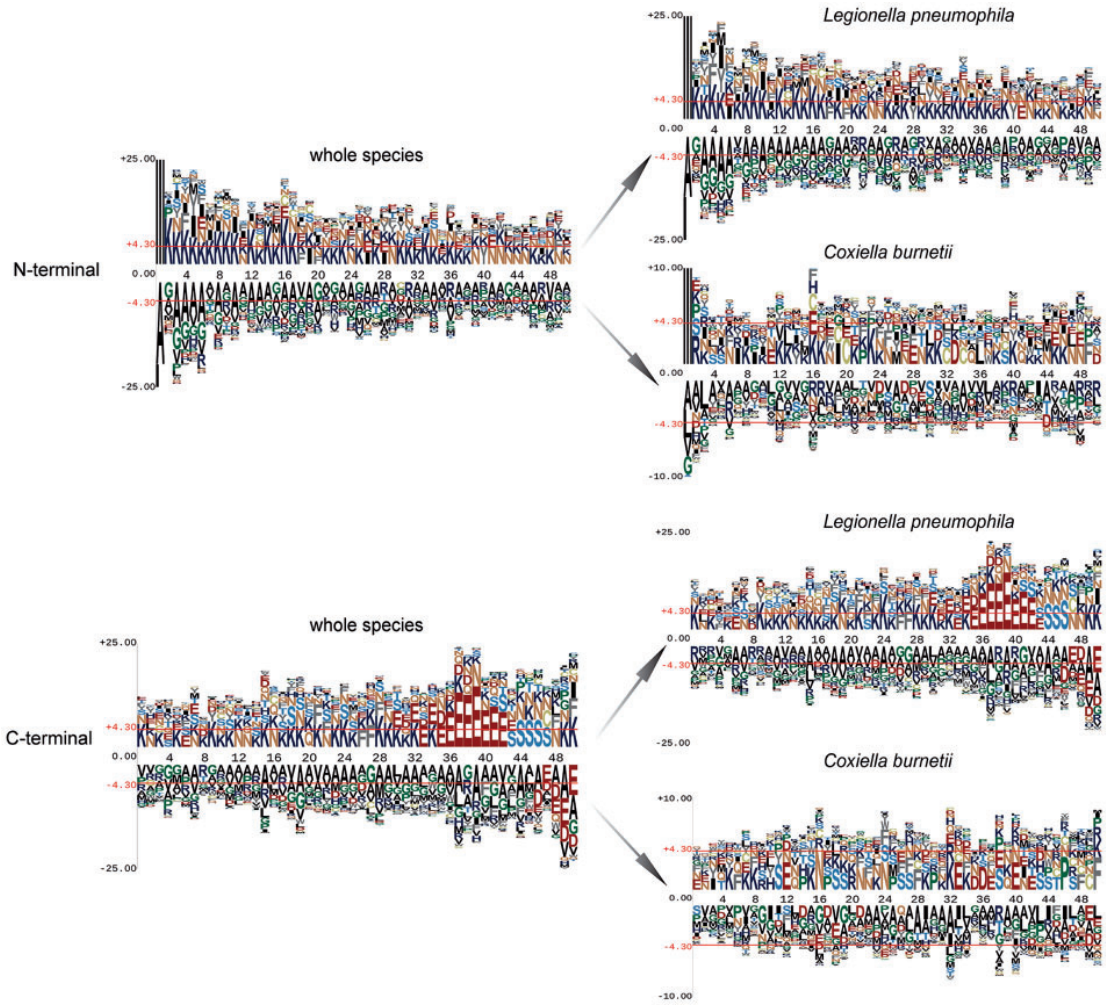


Figure 4. Position-specific amino acid sequence profiles of *L. pneumophila* effectors and *C. burnetii* effectors for both N- and C-terminal 50 positions.

combination model could largely depend on the proportion of PSSM features among the combined features. Other features, when directly combined with the PSSM features, may not contribute to the performance improvement and/or could even result in a decreased performance.

These observations support training single-feature models and subsequently assembling them into ensemble models, instead of merging all features into a vector to train a model.

A majority voting strategy based on ensemble learning models further improves the prediction performance

We first assessed the performance of various classifiers (single-feature encoding-based models and ensemble models) using RF by performing independent tests. All experiments were conducted five times. Each time, 30 negative randomly chosen samples were used to form the balanced independent data set along with the positive samples. The performance results are shown in Table 3 and Figure 10. The predictive performance of models trained by single-feature encodings showed a highly consistent trend with respect to the performance evaluation based on 5-fold cross-validation, further confirming the effectiveness of local and global sequence encodings (Table 3 and Figure 10).

Ensemble models based on selections of single-feature-encoding models were assessed in combination with majority voting, to determine whether this could further improve the predictive performance. Table 3 reports only on a few representative ensemble models selected after comprehensively examining the behaviors of all possible combinations of single-feature models. Several important conclusions were drawn from these results. First, ensemble models achieved a better and more robust performance as compared with single encoding-based models. In particular, the majority voting scheme {1, 3, 5, 6, 8, 10} achieved the overall best performance, with a maximum accuracy of 95.7%, an *F*-score of 0.959 and an MCC value of 0.918 (Table 3). Second, combinations of similar-feature-group encoding-based models did not lead to visible performance improvement. This has been observed in the case of ensemble classifiers {1, 2, 3} (AAC feature group), {5, 6, 7} (PSSM feature group) and {8, 9, 10} (structural feature group). On the other hand, ensembles of models that were trained on different feature groups resulted in clear

Table 2. The performance of various classifiers based on the 5-fold cross-validation tests

Feature	Method	PRE	SN	SP	F-score	ACC	MCC
AAC	RF	0.836 ± 0.009	0.825 ± 0.005	0.839 ± 0.012	0.829 ± 0.006	0.831 ± 0.007	0.663 ± 0.014
	SVM	0.856 ± 0.007	**0.845 ± 0.011**	0.859 ± 0.009	**0.849 ± 0.007**	**0.851 ± 0.007**	**0.703 ± 0.014**
	LR	0.816 ± 0.006	0.834 ± 0.005	0.813 ± 0.007	0.824 ± 0.003	0.823 ± 0.004	0.647 ± 0.009
	NB	0.792 ± 0.005	0.837 ± 0.004	0.782 ± 0.005	0.813 ± 0.004	0.809 ± 0.003	0.619 ± 0.007
	KNN	0.827 ± 0.005	0.838 ± 0.009	0.826 ± 0.006	0.831 ± 0.005	0.831 ± 0.003	0.664 ± 0.008
	MLP	**0.864 ± 0.010**	0.727 ± 0.008	**0.886 ± 0.011**	0.788 ± 0.007	0.805 ± 0.007	0.620 ± 0.013
PPT	RF	0.816 ± 0.006	0.816 ± 0.014	0.817 ± 0.005	0.815 ± 0.010	0.816 ± 0.008	0.633 ± 0.017
	SVM	0.818 ± 0.009	**0.828 ± 0.007**	0.817 ± 0.011	**0.822 ± 0.005**	**0.822 ± 0.005**	**0.645 ± 0.010**
	LR	0.803 ± 0.007	0.788 ± 0.003	0.808 ± 0.008	0.794 ± 0.004	0.797 ± 0.004	0.596 ± 0.008
	NB	0.715 ± 0.006	0.348 ± 0.003	0.860 ± 0.004	0.464 ± 0.004	0.603 ± 0.002	0.243 ± 0.007
	KNN	0.808 ± 0.008	0.745 ± 0.008	0.824 ± 0.008	0.773 ± 0.010	0.783 ± 0.009	0.570 ± 0.016
	MLP	**0.843 ± 0.016**	0.689 ± 0.035	**0.872 ± 0.019**	0.755 ± 0.020	0.779 ± 0.014	0.571 ± 0.027
DPC	RF	0.811 ± 0.015	0.810 ± 0.006	0.812 ± 0.017	0.809 ± 0.010	0.810 ± 0.011	0.621 ± 0.023
	SVM	**0.837 ± 0.007**	0.805 ± 0.010	**0.844 ± 0.010**	0.819 ± 0.005	**0.823 ± 0.004**	**0.648 ± 0.007**
	LR	0.812 ± 0.003	0.839 ± 0.005	0.806 ± 0.002	**0.824 ± 0.003**	0.822 ± 0.002	0.645 ± 0.005
	NB	0.793 ± 0.002	**0.840 ± 0.003**	0.782 ± 0.004	0.815 ± 0.002	0.811 ± 0.003	0.623 ± 0.006
	KNN	0.797 ± 0.004	0.820 ± 0.006	0.793 ± 0.003	0.807 ± 0.005	0.806 ± 0.003	0.612 ± 0.006
	MLP	0.813 ± 0.015	0.681 ± 0.012	0.843 ± 0.014	0.739 ± 0.013	0.761 ± 0.012	0.531 ± 0.023
CKSAAP	RF	0.840 ± 0.004	0.813 ± 0.009	0.846 ± 0.005	**0.825 ± 0.003**	**0.829 ± 0.002**	**0.659 ± 0.006**
	SVM	**0.877 ± 0.005**	0.726 ± 0.009	**0.900 ± 0.006**	0.793 ± 0.004	0.812 ± 0.002	0.635 ± 0.005
	LR	0.737 ± 0.009	0.742 ± 0.012	0.736 ± 0.012	0.738 ± 0.009	0.738 ± 0.009	0.477 ± 0.018
	NB	0.819 ± 0.003	0.831 ± 0.004	0.817 ± 0.004	0.824 ± 0.003	0.823 ± 0.003	0.648 ± 0.006
	KNN	0.763 ± 0.008	**0.860 ± 0.007**	0.732 ± 0.010	0.808 ± 0.006	0.796 ± 0.006	0.598 ± 0.011
	MLP	0.831 ± 0.008	0.733 ± 0.006	0.852 ± 0.007	0.779 ± 0.005	0.792 ± 0.005	0.589 ± 0.010
PSSM	RF	0.909 ± 0.004	0.900 ± 0.005	0.911 ± 0.003	0.904 ± 0.004	**0.905 ± 0.003**	**0.811 ± 0.007**
	SVM	**0.933 ± 0.001**	0.861 ± 0.008	**0.939 ± 0.003**	0.895 ± 0.004	0.900 ± 0.003	0.803 ± 0.006
	LR	0.808 ± 0.007	0.851 ± 0.016	0.797 ± 0.011	0.828 ± 0.008	0.824 ± 0.006	0.649 ± 0.012
	NB	0.888 ± 0.004	0.887 ± 0.003	0.889 ± 0.003	0.887 ± 0.004	0.888 ± 0.003	0.776 ± 0.006
	KNN	0.899 ± 0.003	**0.911 ± 0.003**	0.898 ± 0.003	**0.904 ± 0.003**	0.904 ± 0.003	0.809 ± 0.005
	MLP	0.935 ± 0.013	0.859 ± 0.010	0.943 ± 0.010	0.895 ± 0.009	0.902 ± 0.008	0.806 ± 0.016
PSSM_AC	RF	**0.906 ± 0.006**	0.771 ± 0.009	**0.921 ± 0.005**	**0.832 ± 0.007**	**0.846 ± 0.006**	**0.699 ± 0.012**
	SVM	0.897 ± 0.012	0.765 ± 0.022	0.914 ± 0.012	0.825 ± 0.015	0.839 ± 0.012	0.686 ± 0.022
	LR	0.720 ± 0.011	0.757 ± 0.012	0.705 ± 0.015	0.736 ± 0.008	0.730 ± 0.008	0.463 ± 0.017
	NB	0.610 ± 0.001	**0.867 ± 0.003**	0.447 ± 0.003	0.715 ± 0.002	0.656 ± 0.002	0.346 ± 0.006
	KNN	0.833 ± 0.004	0.816 ± 0.004	0.836 ± 0.004	0.823 ± 0.002	0.825 ± 0.002	0.652 ± 0.006
	MLP	0.896 ± 0.021	0.690 ± 0.009	0.921 ± 0.018	0.777 ± 0.007	0.805 ± 0.007	0.628 ± 0.018
PSSM_SMTH	RF	0.859 ± 0.006	0.825 ± 0.007	0.865 ± 0.006	**0.840 ± 0.005**	**0.844 ± 0.005**	**0.691 ± 0.011**
	SVM	0.873 ± 0.007	0.790 ± 0.014	0.886 ± 0.004	0.828 ± 0.010	0.837 ± 0.008	0.679 ± 0.017
	LR	0.733 ± 0.017	0.734 ± 0.014	0.730 ± 0.026	0.732 ± 0.008	0.732 ± 0.011	0.466 ± 0.020
	NB	0.658 ± 0.003	**0.870 ± 0.002**	0.548 ± 0.006	0.748 ± 0.001	0.708 ± 0.002	0.441 ± 0.006
	KNN	0.804 ± 0.004	0.784 ± 0.005	0.809 ± 0.007	0.793 ± 0.003	0.796 ± 0.005	0.594 ± 0.010
	MLP	**0.886 ± 0.016**	0.756 ± 0.022	**0.909 ± 0.013**	0.815 ± 0.018	0.835 ± 0.016	0.675 ± 0.030
DISO	RF	0.714 ± 0.011	0.733 ± 0.015	0.708 ± 0.011	0.722 ± 0.012	0.719 ± 0.011	0.441 ± 0.022
	SVM	**0.736 ± 0.016**	0.726 ± 0.020	0.739 ± 0.020	**0.728 ± 0.015**	**0.732 ± 0.014**	**0.466 ± 0.027**
	LR	0.604 ± 0.008	0.607 ± 0.018	0.602 ± 0.020	0.603 ± 0.009	0.603 ± 0.007	0.209 ± 0.016
	NB	0.631 ± 0.026	0.657 ± 0.033	0.625 ± 0.009	0.637 ± 0.033	0.640 ± 0.016	0.283 ± 0.037
	KNN	0.695 ± 0.005	**0.746 ± 0.008**	0.674 ± 0.010	0.718 ± 0.004	0.709 ± 0.004	0.422 ± 0.006
	MLP	0.733 ± 0.016	0.570 ± 0.032	**0.791 ± 0.016**	0.639 ± 0.022	0.680 ± 0.014	0.371 ± 0.028
SA	RF	0.611 ± 0.005	0.642 ± 0.010	0.590 ± 0.005	0.623 ± 0.008	**0.613 ± 0.006**	**0.232 ± 0.013**
	SVM	0.604 ± 0.010	0.606 ± 0.022	0.600 ± 0.022	0.601 ± 0.013	0.600 ± 0.010	0.206 ± 0.018
	LR	0.585 ± 0.014	0.591 ± 0.015	0.581 ± 0.016	0.585 ± 0.012	0.583 ± 0.012	0.172 ± 0.026
	NB	0.543 ± 0.006	**0.911 ± 0.011**	0.207 ± 0.007	**0.672 ± 0.006**	0.560 ± 0.007	0.179 ± 0.015
	KNN	**0.633 ± 0.014**	0.498 ± 0.007	**0.711 ± 0.019**	0.555 ± 0.008	0.603 ± 0.010	0.214 ± 0.020
	MLP	0.576 ± 0.019	0.449 ± 0.036	0.671 ± 0.017	0.502 ± 0.030	0.560 ± 0.014	0.123 ± 0.032
SS	RF	0.560 ± 0.022	0.535 ± 0.030	0.579 ± 0.016	0.544 ± 0.025	0.555 ± 0.022	**0.115 ± 0.046**
	SVM	**0.562 ± 0.021**	0.463 ± 0.043	0.634 ± 0.023	0.492 ± 0.034	0.540 ± 0.021	0.102 ± 0.037
	LR	0.536 ± 0.017	0.542 ± 0.022	0.531 ± 0.018	0.537 ± 0.019	0.536 ± 0.018	0.073 ± 0.037
	NB	0.543 ± 0.007	**0.673 ± 0.018**	0.432 ± 0.010	**0.597 ± 0.012**	**0.555 ± 0.007**	0.111 ± 0.015
	KNN	0.530 ± 0.017	0.493 ± 0.018	0.564 ± 0.020	0.505 ± 0.017	0.524 ± 0.016	0.057 ± 0.032
	MLP	0.535 ± 0.024	0.361 ± 0.029	**0.688 ± 0.032**	0.428 ± 0.025	0.525 ± 0.018	0.052 ± 0.037
Group 1	RF	0.835 ± 0.004	0.825 ± 0.003	0.838 ± 0.005	0.829 ± 0.002	0.831 ± 0.003	0.663 ± 0.006
	SVM	0.850 ± 0.008	0.833 ± 0.004	0.854 ± 0.010	**0.840 ± 0.004**	**0.842 ± 0.005**	**0.687 ± 0.011**

(continued)

Table 2. Continued

Feature	Method	PRE	SN	SP	F-score	ACC	MCC
	LR	0.828 ± 0.009	0.829 ± 0.017	0.829 ± 0.011	0.827 ± 0.010	0.828 ± 0.009	0.658 ± 0.018
	NB	0.831 ± 0.001	0.820 ± 0.004	0.835 ± 0.003	0.824 ± 0.002	0.826 ± 0.003	0.654 ± 0.005
	KNN	0.805 ± 0.002	**0.849 ± 0.004**	0.796 ± 0.003	0.825 ± 0.002	0.821 ± 0.001	0.645 ± 0.003
	MLP	**0.882 ± 0.005**	0.743 ± 0.017	**0.902 ± 0.007**	0.805 ± 0.009	0.821 ± 0.007	0.652 ± 0.013
Group 2	RF	0.930 ± 0.003	0.865 ± 0.004	0.935 ± 0.003	0.895 ± 0.003	0.899 ± 0.003	0.802 ± 0.006
	SVM	0.938 ± 0.003	0.856 ± 0.012	0.945 ± 0.003	0.895 ± 0.007	0.900 ± 0.005	**0.804 ± 0.010**
	LR	0.827 ± 0.012	0.852 ± 0.009	0.822 ± 0.013	0.838 ± 0.007	0.836 ± 0.006	0.674 ± 0.012
	NB	0.679 ± 0.003	0.881 ± 0.002	0.584 ± 0.006	0.765 ± 0.001	0.731 ± 0.002	0.487 ± 0.006
	KNN	0.905 ± 0.006	**0.894 ± 0.004**	0.907 ± 0.007	**0.899 ± 0.005**	**0.900 ± 0.004**	0.800 ± 0.008
	MLP	**0.964 ± 0.007**	0.789 ± 0.052	**0.972 ± 0.007**	0.864 ± 0.033	0.879 ± 0.026	0.775 ± 0.044
Group 3	RF	0.730 ± 0.012	0.737 ± 0.014	0.728 ± 0.015	0.731 ± 0.011	0.730 ± 0.010	0.465 ± 0.019
	SVM	**0.742 ± 0.011**	0.736 ± 0.017	0.744 ± 0.014	**0.737 ± 0.013**	**0.738 ± 0.012**	**0.481 ± 0.020**
	LR	0.622 ± 0.006	0.629 ± 0.017	0.617 ± 0.009	0.623 ± 0.010	0.621 ± 0.007	0.246 ± 0.016
	NB	0.582 ± 0.010	**0.829 ± 0.016**	0.393 ± 0.014	0.679 ± 0.010	0.611 ± 0.010	0.250 ± 0.028
	KNN	0.718 ± 0.006	0.703 ± 0.009	0.725 ± 0.010	0.708 ± 0.006	0.712 ± 0.006	0.427 ± 0.011
	MLP	0.684 ± 0.009	0.445 ± 0.021	**0.794 ± 0.014**	0.536 ± 0.015	0.619 ± 0.007	0.255 ± 0.014
All features	RF	0.912 ± 0.005	0.860 ± 0.006	0.919 ± 0.004	0.885 ± 0.005	0.889 ± 0.005	0.779 ± 0.008
	SVM	0.931 ± 0.004	0.864 ± 0.009	0.937 ± 0.003	**0.896 ± 0.007**	**0.900 ± 0.006**	**0.803 ± 0.010**
	LR	0.887 ± 0.006	0.873 ± 0.010	0.890 ± 0.006	0.878 ± 0.007	0.880 ± 0.006	0.762 ± 0.012
	NB	0.809 ± 0.005	0.885 ± 0.003	0.792 ± 0.007	0.844 ± 0.002	0.838 ± 0.003	0.680 ± 0.007
	KNN	0.900 ± 0.006	**0.887 ± 0.006**	0.904 ± 0.006	0.893 ± 0.003	0.894 ± 0.002	0.790 ± 0.005
	MLP	**0.943 ± 0.009**	0.715 ± 0.039	**0.956 ± 0.007**	0.806 ± 0.027	0.833 ± 0.017	0.692 ± 0.026

Note: The values were expressed as mean ± standard error. Except for AAC (20 D) and PPT (72 D), all the feature vectors were 200-dimensional, and their selection was performed using GainRatio. Group 1 denotes the AAC group (AAC, DPC, CKSAAP and PPT); Group 2 denotes the PSSM group (PSSM, PSSM_AC and PSSM_SMTH); Group 3 denotes the structure group (SA, SS and DISO), while all features include all the 10 feature types and are used as a whole group. For each group, individual models were trained with the corresponding group and then integrated as an ensemble model using the majority vote scheme. For each performance measure, the best performance value across different machine learning methods within a feature group is highlighted in bold for clarification. These highlights also apply to Tables 3, 4 and 6.

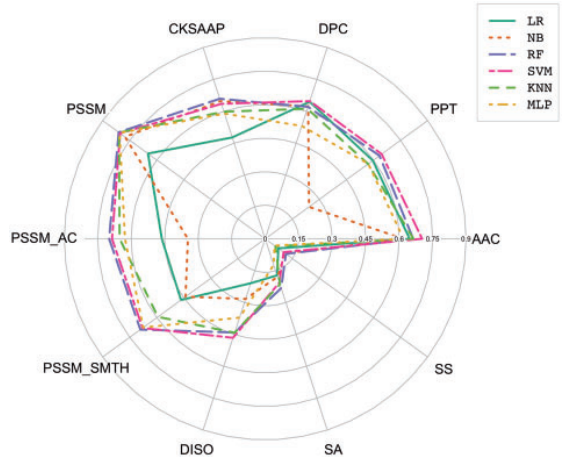


Figure 5. Prediction performance of different machine learning models trained with various feature encodings in terms of MCC on the 5-fold cross-validation test.

performance improvements, e.g. ensemble classifiers {1, 2, 3, 4, 5, 6, 7}, {1, 2, 3, 4, 8, 9, 10}, {5, 6, 7, 8, 9, 10} and {1–10}. This is in agreement with the result in [17] by exploring the vote of various feature-based models (including two sequence-based models and two PSSM-based models). The ensemble classifier {1, 2, 3, 4, 5, 6, 7, 8} is an excellent example portraying the advantages of ensemble learning. Comparing it with the ensemble classifier {1, 2, 3, 4, 5, 6, 7} showed that the DISO feature-based model still contributes to an improved performance of the ensemble classifier, while it only gives a moderate performance when used as a single-feature model.

For each of machine learning methods (i.e. SVM, KNN, NB, LR and MLP), we trained an ensemble model by integrating eight top single-feature-based models (i.e. AAC, PPT, DPC, CKSAAP, PSSM, PSSM_AC, PSSM_SMTH and DISO). By further integrating ensemble models with the majority vote scheme, we studied the prediction performance of these single machine learning-based models and their ensemble models using the independent test. As shown in Table 4, the RF- and SVM-based models outperformed other method-based models, while the ensemble model of these two models ({a, b}) further improved the prediction performance. The ensemble model integrating all six method-based models ({a, b, c, d, e, f}) achieved the best performance in terms of *F*-value, ACC and MCC, consistent with the observations reported in [19]. Based on these findings, we constructed Bastion4 with a default setting: all six machine learning methods were integrated, and for each of them the eight top single-feature-based models were generated for assembling.

Performance evaluation of specific training data sets

To investigate whether the diversity of positive samples affects the performance of the predictors, we trained another two predictors using a part of the training data set. In more detail, from the positive samples in the training data set, 291 *Legionella* samples were chosen to construct a new balanced independent data set together with randomly selected negative samples from *V. parahaemolyticus* serotype O3: K6. The remaining 99 positive samples and an equal number of negative samples from the original training set were used to form a new training data set. Based on this new data set, we used eight feature types (i.e. AAC, PPT, DPC, CKSAAP, PSSM, PSSM_AC, PSSM_SMTH and DISO) to train individual models and aggregated their outputs to form an ensemble model for each of the six machine learning methods.

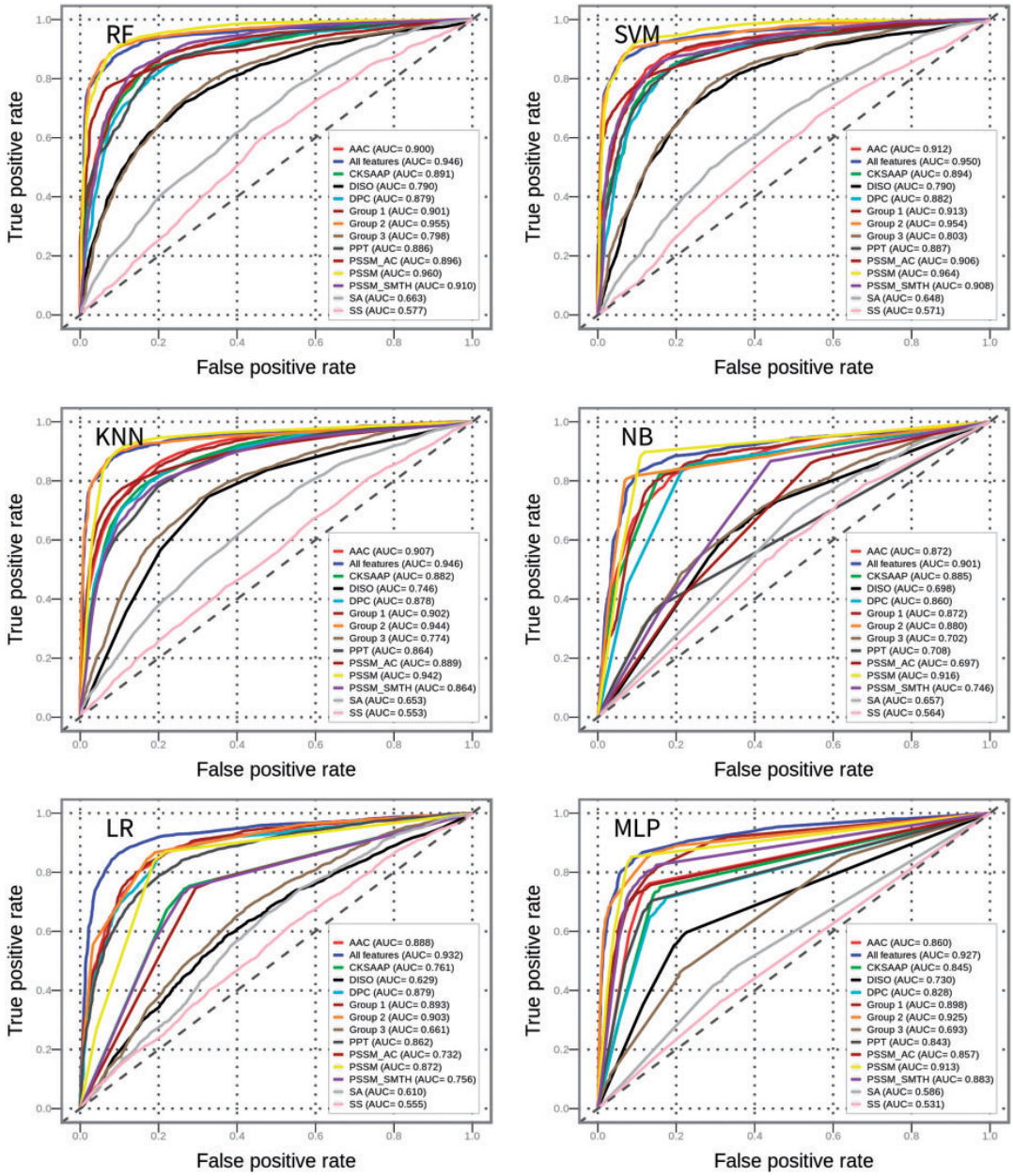


Figure 6. ROC curves of RF, SVM, NB, KNN, LR and MLP predictors of T4SEs with different feature encodings. Group 1 denotes the AAC group (AAC, DPC, CKSAAP and PPT); Group 2 denotes the PSSM group (PSSM, PSSM_AC and PSSM_SMTH); Group 3 denotes the structure group (SA, SS and DISO), while all features include all the 10 feature types and are used as a whole group. For each group, individual models were trained with the corresponding group and then integrated as an ensemble model using the majority vote scheme.

We further integrated all these single method-based models with the majority vote scheme to construct a new predictor (labeled 'Predictor_without_Legionella'). The new independent data set (containing all 291 *Legionella* samples as positives) was used to analyze the predictive performance. We applied the same procedures to construct a predictor (labeled 'Predictor_without_Coxiella') and analyzed its performance on the new independent data set (containing all 60 *Coxiella* samples as positives). In addition, eight single models trained using the full training data set were assembled as a reference predictor (labelled 'Predictor_with_Full_Dataset'). The overall performance of the three predictors was assessed based on their respective independent test data sets and is listed in Table 5.

The Predictor_with_Full_Dataset outperformed the Predictor _without_*Coxiella* and the Predictor_without_*Legionella* in terms of *F*-value, ACC and MCC (Table 5). These results indicate that the increase of samples diversity can improve the performance of predictors. Owing to limited training data, the Predictor_

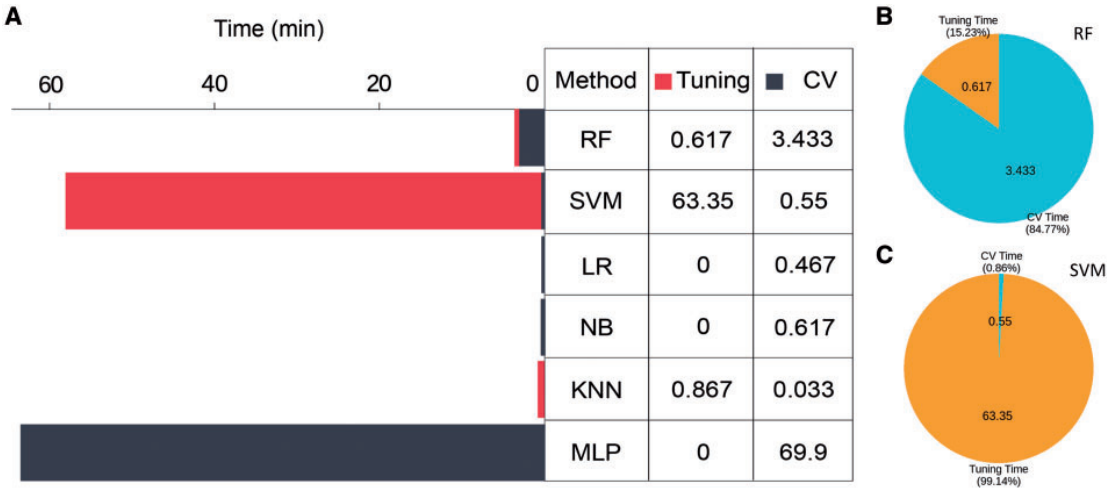


Figure 7. (**A**) Computational time of various classifiers when using the PSSM feature for training (after applying feature selection to form a 200-dimensional vector using GainRatio). Parameter tuning time and CV time are counted into the overall computational time for the classifiers. For classifiers without parameter optimization (LR, NB and MLP), the tuning time is 0. (**B**) and (**C**) represent the proportions of parameter tuning time and CV time of RF and SVM, respectively.

without_*Legionella* failed to achieve a competitive performance. Note that the high SN value of the Predictor_without_*Coxiella* suggests that it is well able to identify the *Coxiella* T4SEs even without using such data set for model training. This hints at an underlying similarity between *Legionella*, *Coxiella* and other T4SEs. Here, we used an unsupervised learning approach to investigate the potential similarity further. We encoded all T4SEs using PSSM encoding, partitioned them into three groups using the *k*-means clustering algorithm [78] and then performed dimension-reduction using *t*-SNE [79] to map to the 2D space for better visualization (Figure 11A). From Figure 11B–D, we can see that each of the identified three clusters is a mixture of *Legionella*, *Coxiella* and other T4SEs. This supports the idea that, because of their similarity, these types of T4SEs are not separable. The observation that *Legionella* samples dominate all three clusters can be attributed to their abundance in the original three classes of positive samples (Figure 11E).

While there are similarities between *Legionella*'s, *Coxiella*'s and other T4SEs, it is noteworthy that the performance of Predictor_without_*Legionella* was less than that of the Predictor_without_*Coxiella*. To explore why this is so, we used Clustal Omega [80] to do multiple sequence alignment on the T4SE data set. Based on the alignment results, a phylogenetic tree of T4SS effectors was generated (Supplementary Figure S2) using iTOL [81]. From inspection of Supplementary Figure S2, we found that T4SEs in *Legionella*, *Coxiella* and other species were often mixed, while some T4SEs in *Legionella* clustered alone (marked in light green). This finding indicated that some T4SEs in *Legionella* differ from those in other species, shedding light on why Predictor_without_*Legionella* could not distinguish some of T4SEs in *Legionella* species.

Performance comparison with existing tools

There are currently two tools [17, 18] available for T4SE prediction. Three classifiers (T4SEpred_bpbAac, T4SEpred_psAac and T4SEpre_Joint) were implemented in Wang *et al.* [18], while a second tool (T4Effpred) with multiple options was developed in Zou *et al.* [17]. Accordingly, we compared their performance on the independent test data set (Table 6).

All options of T4Effpred were explored but, for the sake of brevity, Table 6 only presents the best-performing model from among different T4Effpred variant models [17]: an ensemble model based on a 3-in-4 voting scheme. For the same reason, only the results of T4SEpred_bpbAac and T4SEpred_psAac are listed in Table 6. In Bastion4, default settings were used to construct the predictor. As can be seen from Table 6, Bastion4 achieved an overall accuracy of 95.3% with an *F*-value of 0.954 and an MCC of 0.907. This is the best performance among all compared predictors. T4Effpred achieved the second-best performance, also using an ensemble classifier based on multiple feature encodings. Moreover, we observed that both T4SEpred_bpbAac and T4SEpred_psAac performed poorly on the independent test data set.

The poorer performance of T4SEpred_bpbAac and T4SEpred_psAac is intriguing, especially considering important features of T4SE proteins that might be biologically important. The implementation of the two predictors did not extract features from the PSSM profiles, which are regarded as the primary features [18], and these have proved to be powerful for T4SE prediction both in our current work and in the work by Zou *et al.* [17]. Coupled with this, in T4SEpred_bpbAac and T4Sepred_psAac, only the 50 C-terminal amino acids, rather than whole protein sequences, were used to extract features [18]. As also shown in this study, pronounced sequence signals are present at the C-terminus of *L. pneumophila* effectors, but are not universal and diagnostic of all T4Ses. Our results presented here demonstrate that the complete sequences contain important information that is relevant for accurate T4SE prediction and, presumably, for their recognition by the T4SS.

Genome-wide prediction of T4SEs in *Klebsiella pneumoniae*

Klebsiella pneumoniae is emerging as a devastating pathogen of humans [82]. The T4SS of this pathogen has only been recently noted [83, 84], and effector proteins and T4SEs have not been identified to our knowledge. We took this opportunity to predict T4SEs with Bastion4 using our default settings, and to identify these on physical genome maps of three clinically relevant

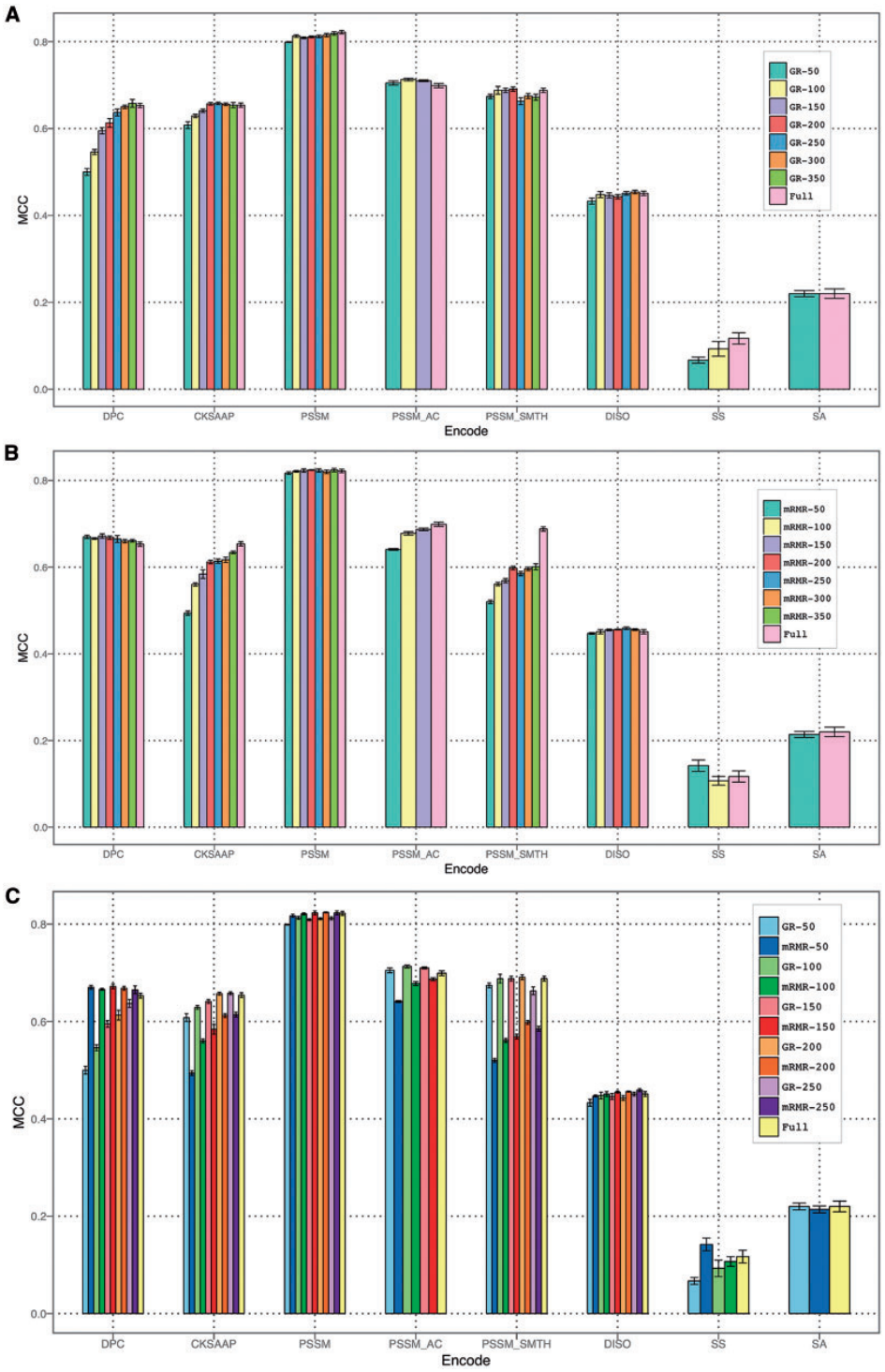


Figure 8. Feature selection by using GainRatio and mRMR methods. The error bars indicate the SDs of MCC values over five different randomized 5-fold cross-validation tests. (**A**) Performance of various feature encodings with different numbers of top features selected by GainRatio; (**B**) performance of various feature encodings with different numbers of top features selected by mRMR; (**C**) side-by-side performance comparison of various feature encodings with different numbers of top features selected by GainRatio versus mRMR.

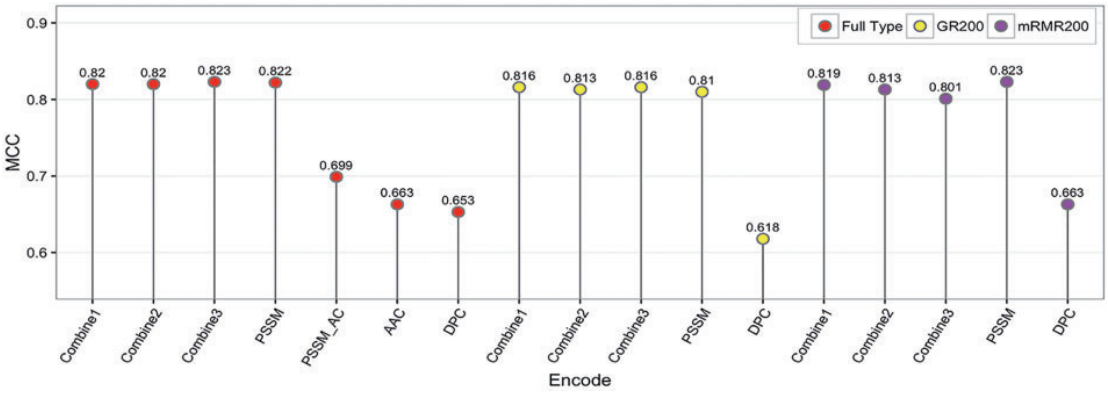


Figure 9. Performance comparison of models trained using single features versus combined features, based on 5-fold cross-validation using the training data set. Combine1 denotes the composition of PSSM and PSSM_AC; Combine2 denotes the composition of PSSM, PSSM_AC and AAC; Combine3 denotes the composition of PSSM, PSSM_AC, AAC and DPC.

Table 3. The performance of various classifiers based on the independent tests

Model[a]	Voting[b]	PRE	SN	SP	*F*-value	ACC	MCC
1. AAC	–	0.826 ± 0.044	**1.000 ± 0.000**	0.787 ± 0.069	**0.904 ± 0.027**	**0.893 ± 0.035**	**0.806 ± 0.057**
2. PPT	–	0.787 ± 0.057	0.967 ± 0.000	0.733 ± 0.091	0.867 ± 0.035	0.850 ± 0.046	0.721 ± 0.077
3. DPC	–	0.791 ± 0.039	0.900 ± 0.000	0.760 ± 0.055	0.842 ± 0.022	0.830 ± 0.027	0.667 ± 0.050
4. CKSAAP	–	0.839 ± 0.014	0.933 ± 0.000	0.820 ± 0.018	0.883 ± 0.008	0.877 ± 0.009	0.758 ± 0.017
5. PSSM	–	0.821 ± 0.033	**1.000 ± 0.000**	0.780 ± 0.051	0.901 ± 0.020	0.890 ± 0.025	0.800 ± 0.042
6. PSSM_AC	–	**0.882 ± 0.049**	0.833 ± 0.000	**0.887 ± 0.051**	0.857 ± 0.023	0.860 ± 0.025	0.722 ± 0.053
7. PSSM_SMTH	–	0.811 ± 0.080	0.800 ± 0.000	0.807 ± 0.095	0.804 ± 0.039	0.803 ± 0.048	0.609 ± 0.097
8. DISO	–	0.778 ± 0.061	0.800 ± 0.000	0.767 ± 0.082	0.788 ± 0.032	0.783 ± 0.041	0.568 ± 0.080
9. SA	–	0.645 ± 0.059	0.667 ± 0.000	0.627 ± 0.095	0.655 ± 0.030	0.647 ± 0.048	0.294 ± 0.095
10. SS	–	0.665 ± 0.065	0.700 ± 0.000	0.640 ± 0.092	0.681 ± 0.032	0.670 ± 0.046	0.342 ± 0.093
{1, 2, 3, 4}	3-in-4	0.854 ± 0.025	0.967 ± 0.000	0.833 ± 0.033	0.906 ± 0.014	0.900 ± 0.017	0.807 ± 0.030
{5, 6, 7}	2-in-3	0.880 ± 0.048	0.867 ± 0.000	0.880 ± 0.051	0.873 ± 0.023	0.873 ± 0.025	0.748 ± 0.052
{8, 9, 10}	2-in-3	0.788 ± 0.092	0.800 ± 0.000	0.773 ± 0.123	0.792 ± 0.047	0.787 ± 0.062	0.576 ± 0.121
{1, 2, 3, 4, 5, 6, 7}	4-in-7	0.854 ± 0.030	0.967 ± 0.000	0.833 ± 0.041	0.907 ± 0.017	0.900 ± 0.020	0.808 ± 0.036
{1, 2, 3, 4, 8, 9, 10}	4-in-7	0.850 ± 0.045	0.967 ± 0.000	0.827 ± 0.060	0.904 ± 0.025	0.897 ± 0.030	0.802 ± 0.054
{5, 6, 7, 8, 9, 10}	4-in-6	0.903 ± 0.058	0.900 ± 0.000	0.900 ± 0.067	0.901 ± 0.029	0.900 ± 0.033	0.801 ± 0.066
{1, 2, 3, 4, 5, 6, 7, 8}	5-in-8	0.918 ± 0.025	0.967 ± 0.000	**0.913 ± 0.030**	0.942 ± 0.014	0.940 ± 0.015	0.882 ± 0.028
{1-10}	6-in-10	0.908 ± 0.045	0.967 ± 0.000	0.900 ± 0.053	0.936 ± 0.024	0.933 ± 0.026	0.869 ± 0.050
{1, 3, 5, 6, 8, 10}	4-in-6	**0.922 ± 0.042**	**1.000 ± 0.000**	**0.913** ± 0.051	**0.959 ± 0.023**	**0.957 ± 0.025**	**0.918 ± 0.046**

Note: [a]Each term in this column refers to a single encoding-based model or an ensemble model of different single encoding-based models (e.g. 1. AAC means the model trained with AAC encoding features, while {5, 6, 7} stands for the ensemble model of number 5, 6 and 7 classifiers).
[b]The majority voting scheme was used for voting in ensemble models.

strains: *K. penumoniae* AJ218, B5055 and MGH 78578. Studies with other bacteria have identified the physical location of genomic regions encoding T4SEs [85–87], and the genes encoding certain T4SEs were found to be clustered within specific genomic regions with an observed bias in G + C content, leading to models, whereby T4SE genes are acquired by lateral gene transfer between different bacterial species [16, 88, 89].

Circular maps [90] of extant genome sequence data were generated (Figure 12) to graphically depict the relationships between genome properties and the distribution of predicted effectors in these genomes [91]. The G + C content of the tentative T4SEs in each of the three genomes is significantly lower than expected from the overall G + C contents (Table 7), all with significant *P*-values according to the Welch's *t*-test. The Venn diagram in Figure 12D illustrates the distributions of both predicted strain-specific and common effectors across these three bacterial genomes. While they share some common effectors (four common effectors shared across the three strains), AJ218, B5055 and MGH 78578 had 42, 33 and 33 strain-specific effectors possibly because of relatively recent horizontal gene transfer events [89]. This is consistent with our knowledge that genes encoding effector proteins are often shared via lateral gene transfer from other species. In the *K. penumoniae* B5055 genome, there is a cluster of predicted T4SEs genes that sit spatially in the nearby genes encoding the components of the T4SS nanomachine. Taken together, the genome-wide predictions of T4SEs provide a basis to explore their genome-level distributions, and to build a compendium of novel putative T4SEs that can be characterized by genetic and biochemical experiments.

Availability of online Web servers

As an implementation of the methodology presented here, we developed Bastion4, an online Web server for characterizing protein

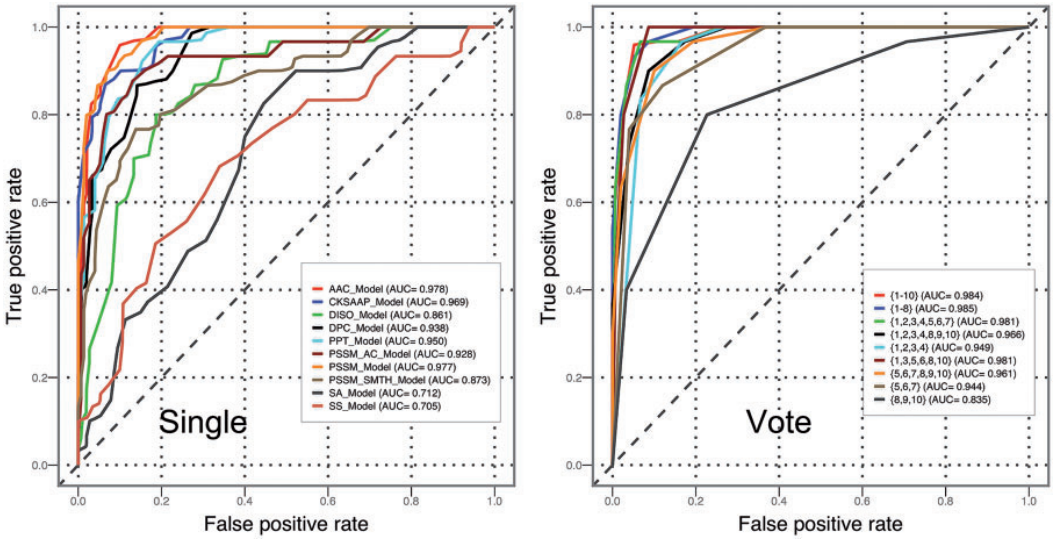


Figure 10. The ROC curve of both single encoding-based models and ensemble models based on the independent test.

Table 4. The performance of various machine learning methods based on the independent tests

Method[a]	**Voting**[b]	PRE	SN	SP	F-value	ACC	MCC
a. RF	–	0.918 ± 0.025	0.967 ± 0.000	0.913 ± 0.030	0.942 ± 0.014	0.940 ± 0.015	0.882 ± 0.028
b. SVM	–	0.940 ± 0.014	0.933 ± 0.000	0.940 ± 0.015	0.936 ± 0.007	0.937 ± 0.007	0.873 ± 0.015
c. KNN	–	0.880 ± 0.055	**1.000 ± 0.000**	0.860 ± 0.072	0.935 ± 0.031	0.930 ± 0.036	0.870 ± 0.064
d. NB	–	0.834 ± 0.033	0.933 ± 0.000	0.813 ± 0.045	0.881 ± 0.019	0.873 ± 0.022	0.753 ± 0.041
e. LR	–	0.875 ± 0.056	**1.000 ± 0.000**	0.853 ± 0.069	0.932 ± 0.031	0.927 ± 0.035	0.864 ± 0.063
f. MLP	–	0.906 ± 0.004	0.960 ± 0.043	0.900 ± 0.000	0.932 ± 0.023	0.930 ± 0.022	0.862 ± 0.046
{a, b}	2-in-2	**0.966 ± 0.024**	0.933 ± 0.000	**0.967 ± 0.024**	0.949 ± 0.011	0.950 ± 0.012	0.901 ± 0.024
{a, b, c}	2-in-3	0.918 ± 0.025	0.967 ± 0.000	0.913 ± 0.030	0.942 ± 0.014	0.940 ± 0.015	0.882 ± 0.028
{a, b, c, d}	3-in-4	0.918 ± 0.025	0.967 ± 0.000	0.913 ± 0.030	0.942 ± 0.014	0.940 ± 0.015	0.882 ± 0.028
{a, b, c, d, e}	3-in-5	0.907 ± 0.028	0.967 ± 0.000	0.900 ± 0.033	0.936 ± 0.015	0.933 ± 0.017	0.869 ± 0.031
{a, b, c, d, e, f}	4-in-6	0.942 ± 0.025	0.967 ± 0.000	0.940 ± 0.028	**0.954 ± 0.013**	**0.953 ± 0.014**	**0.907 ± 0.027**

Note: [a]Each term in this column refers to a single method-based model or an ensemble model that integrates different single method-based models (e.g. 'a. RF' means the model is trained based on the RF method, while '{a, b, c}' stands for the ensemble model that integrates a, b and c models).
[b]The majority voting scheme is used for voting in ensemble models.

Table 5. Performance comparison between Predictor_without_*Coxiella*, Predictor_without_*Legionella* and Predictor_with_Full_Dataset based on the independent test

Classifier	PRE	SN	SP	F-value	ACC	MCC
Predictor_with_Full_Dataset	0.942	0.967	0.940	0.954	0.953	0.907
Predictor_without_*Coxiella*	1.000	0.733	1.000	0.846	0.867	0.761
Predictor_without_*Legionella*	0.841	0.691	0.869	0.758	0.780	0.569

sequences of interest. Bastion4 is freely accessible at http://bastion4.erc.monash.edu/. The Bastion4 Web server was programmed using the Perl CGI and J2EE framework, and configured on the cloud computing facility provided by the Monash University e-Research Centre. Users can submit multiple protein sequences in raw or FASTA format to the online Web server. The computational time of the Bastion4 server to process a submitted sequence depends not only on the length of the submitted sequence but also considerably on the choice of the selected models.

Conclusion

Identifying effector proteins is necessary to understand host–pathogen interactions and bacterial pathogenesis. Here, we have systematically assessed the use and performance of different protein sequence and protein structure-related features and their combinations along with various machine learning approaches for T4SE prediction. Our main findings are (1) of the six machine learning classifiers (NB, KNN, LR, RF, SVM and MLP), RF and SVM

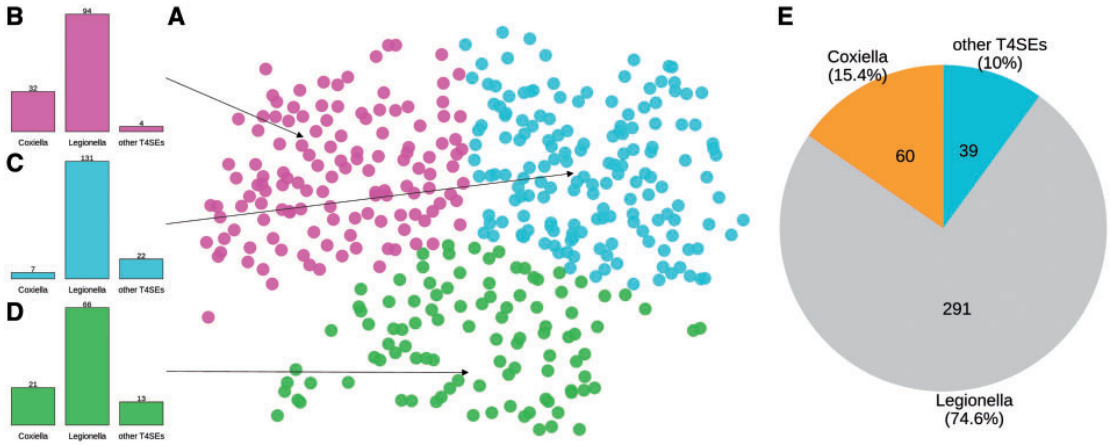


Figure 11. (**A**) Representation of the positive samples from *Coxiella*, *Legionella* and other T4SEs. The representation of each sample (which constituted a 400-dimensional space generated by the PSSM encoding scheme) was reduced to two dimensions by using t-SNE. Samples were clustered into three groups using *K*-means algorithm, and these three clusters were indicated by colors. (**B–D**) Detailed components of the three clusters. Each cluster contains samples from all three T4SE classes, namely, *Coxiella*, *Legionella* and others. (**E**) Detailed numbers and proportions of original three classes of samples.

Table 6. Performance comparison between our ensemble classifier and other existing predictors based on the independent test

Classifier	PRE	SN	SP	F-value	ACC	MCC
Bastion4	0.942 ± 0.025	0.967 ± 0.000	0.940 ± 0.028	**0.954 ± 0.013**	**0.953 ± 0.014**	**0.907 ± 0.027**
T4Effpred	0.940 ± 0.020	0.833 ± 0.000	0.947 ± 0.018	0.883 ± 0.009	0.890 ± 0.009	0.785 ± 0.020
T4SEpred_bpbAac	0.959 ± 0.060	0.433 ± 0.000	0.980 ± 0.030	0.597 ± 0.012	0.707 ± 0.015	0.495 ± 0.046
T4SEpred_psAac	**0.983 ± 0.037**	0.367 ± 0.000	**0.993 ± 0.015**	0.534 ± 0.006	0.680 ± 0.007	0.462 ± 0.026

Table 7. Statistical analysis of the G + C contents between the putative T4SEs and non-T4SEs in the *K. pneumoniae* strain AJ218, B5055 and MGH 78578 genomes

Strain type	Mean of G + C content (%)		*P*-value by Welch's *t*-test
	Putative T4SEs	Non-T4SEs	
AJ218	43.33	57.73	3.755e-16
B5055	44.99	57.55	3.51e-11
MGH 78578	45.45	57.99	4.314e-10

Note: For each species, the G + C content (%) of each sequence of putative T4SEs was calculated to form a percentage set. Similarly, the G + C content (% of each non-T4SE sequence was calculated to form another percentage set. Note that the percentage set of non-T4SE sequences was significantly larger than that of the putative T4SEs percentage set. Based on the two sets, the mean values of the G + C content (%) of both putative T4SEs and non-T4SEs were calculated. The Welch's *t*-test was performed and *P*-value calculated to assess the statistical significance.

performed best according to the performance measures ACC, F-value and MCC based on 5-fold cross-validation, while RF achieved a good trade-off between the predictive performance and computational time; (2) of the 10 different features, PSSM achieved the highest performance values for all classifiers, emphasizing the importance of global sequence encoding with PSSMs; (3) ensemble models performed better than single-feature-based models; (4) when applied to the predictions of an ensemble model, the diversity in the selected features resulted in a more stable and accurate classification performance. These findings led to the development of Bastion4, a tool that reflects the state of the art in effector prediction for T4SEs. Together with the compendium of predicted tentative T4SEs of the three bacterial genomes, we anticipate Bastion4 to be extensively used for T4SE prediction and, in conjunction with comparative genomics of bacterial pathogens, to improve our understanding of host–pathogen interactions.

Key Points

- In this work, we systematically train and compare six commonly used machine learning models for accurate and efficient identification of T4SEs using 10 different types of selected features.
- Our study shows that (1) including different but complementary features generally enhance the predictive performance of T4SEs; (2) ensemble models obtained by integrating individual single-feature models exhibit a significantly improved predictive performance. The majority voting strategy enables the ensemble models to achieve the most stable and accurate classification performance.
- We propose and built a new ensemble model, Bastion4, to further improve the performance in predicting effector proteins of the T4SS. Independent tests demonstrate that the ensemble models outperform all current predictors of types IV secretion systems. We make Bastion4 publicly accessible at http://bastion4.erc.monash.edu/.
- Genome-wide prediction of T4SEs provides important insights into the distribution of T4SEs in three bacterial pathogens. We provide a valuable compendium of novel T4SEs that can be further validated by genetic and biochemical experiments.

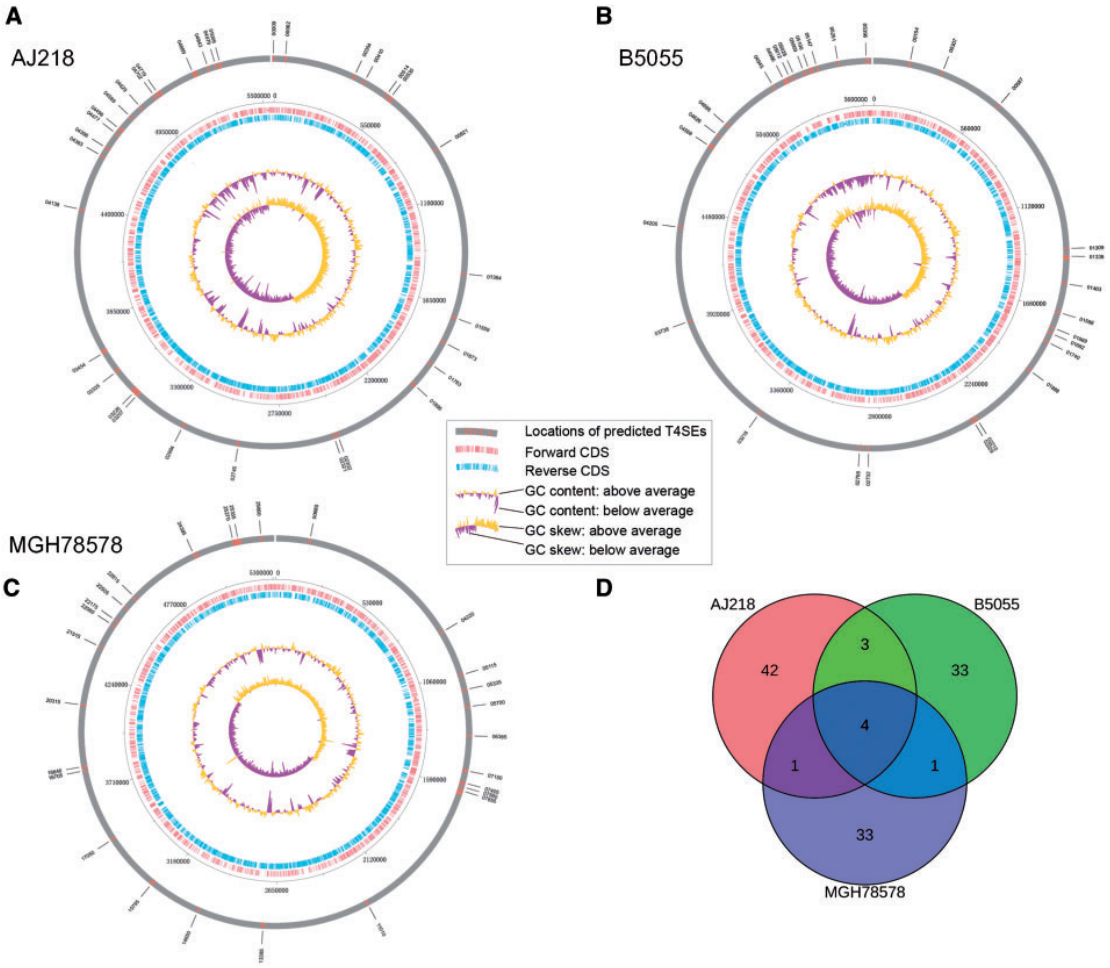


Figure 12. (**A**–**C**) Circular maps of representation of the genomes of *K. pneumoniae* AJ218, B5055 and MGH 78578 strains and (**D**) Venn diagram of the distributions of predicted effectors in these three bacterial genomes. In the (**A**–**C**), the tracks from the outside to the inside represent: (1) locations of predicted T4SEs; (2) forward coding DNA sequence (CDS); (3) reverse CDS; (4) GC content (yellow denotes that G + C content is higher than the average; purple denotes that G + C content is lower than the average); and (5) GC skew [(GC)/(G + C)].

Supplementary Data

Supplementary data are available online at http://bib.oxfordjournals.org/.

Funding

This work was supported by grants from the National Health and Medical Research Council of Australia (NHMRC) (grant number 1092262), the Australian Research Council (ARC) (grant numbers LP110200333 and DP120104460) and the National Institute of Allergy and Infectious Diseases of the National Institute of Health (grant number R01 AI111965). G.I.W. is a recipient of the Discovery Outstanding Research Award (DORA) of the ARC. T.L. is an ARC Australian Laureate Fellow (grant number FL130100038).

References

1. Eicher SC, Christoph D. Bartonella entry mechanisms into mammalian host cells. *Cell Microbiol* 2012;**14**(8):1166–73.
2. Rhomberg TA, Truttmann MC, Guye P. A translocated protein of *Bartonella hensela*: interferes with endocytic uptake of individual bacteria and triggers uptake of large bacterial aggregates via the invasome. *Cell Microbiol* 2009;**11**(6): 927–45.
3. Truttmann MC, Rhomberg TA, Dehio C. Combined action of the type IV secretion effector proteins BepC and BepF promotes invasome formation of *Bartonella henselae* on endothelial and epithelial cells. *Cell Microbiol* 2011;**13**(2):284–99.
4. Navarro L, Alto NM, Dixon JE. Functions of the Yersinia effector proteins in inhibiting host immune responses. *Curr Opin Microbiol* 2005;**8**(1):21–7.
5. Mcghie EJ, Brawn LC, Hume PJ, *et al*. Salmonella takes control: effector-driven manipulation of the host. *Curr Opin Microbiol* 2009;**12**(1):117–24.

6. O'Brien KM, Lindsay EL, Starai VJ. The *Legionella pneumophila* effector protein, LegC7, alters yeast endosomal trafficking. *PLoS One* 2015;**10**:
7. Ku B, Lee KH, Park WS, *et al*. VipD of *Legionella pneumophila* targets activated Rab5 and Rab22 to interfere with endosomal trafficking in macrophages. *PLoS Pathog* 2012;**8**(12):e1003082.
8. Hubber A, Roy CR. Modulation of host cell function by *Legionella pneumophila* type IV effectors. *Annu Rev Cell Dev Biol* 2010;**26**(1):261–83.
9. Clark CS, Maurelli AT. *Shigella flexneri* inhibits staurosporine-induced apoptosis in epithelial cells. *Infect Immun* 2007;**75**(5): 2531–9.
10. Ashida H, Kim M, Sasakawa C. Manipulation of the host cell death pathway by *Shigella*. *Cell Microbiol* 2014;**16**(12):1757–66.
11. Trosky JE, Liverman AD, Orth K. *Yersinia* outer proteins: Yops. *Cell Microbiol* 2008;**10**(3):557–65.
12. Dong N, Zhu Y, Lu Q, *et al*. Structurally distinct bacterial TBC-like GAPs link Arf GTPase to Rab1 inactivation to counteract host defenses. *Cell* 2012;**150**(5):1029–41.
13. Green ER, Mecsas J. Bacterial secretion systems: an overview. *Microbiol Spectr* 2016;**4**.
14. Gophna U, Ron EZ, Dan G. Bacterial type III secretion systems are ancient and evolved by multiple horizontal-transfer events. *Gene* 2003;**312**:151–63.
15. Burns DL. Type IV transporters of pathogenic bacteria. *Curr Opin Microbiol* 2003;**6**(1):29–34.
16. Burstein D, Zusman T, Degtyar E, *et al*. Genome-scale identification of Legionella pneumophila effectors using a machine learning approach. *PLoS Pathog* 2009;**5**(7):6974.
17. Zou L, Nan C, Hu F. Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles. *Bioinformatics* 2013;**29**(24):3135–42.
18. Wang Y, Wei X, Bao H, *et al*. Prediction of bacterial type IV secreted effectors by C-terminal features. *BMC Genomics* 2014;**15**(1):1–14.
19. Zou L, Chen K. Computational prediction of bacterial type IV-B effectors using C-terminal signals and machine learning algorithms. In: *2016 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, Chiang Mai, Thailand, 2016. IEEE, pp. 1–5.
20. An Y, Wang J, Li C, *et al*. Comprehensive assessment and performance improvement of effector protein predictors for bacterial secretion systems III, IV and VI. *Brief Bioinform* 2016, doi:10.1093/bib/bbw100.
21. Zeng C, Zou L. An account of in silico identification tools of secreted effector proteins in bacteria and future challenges. *Brief Bioinform* 2017, doi:10.1093/bib/bbx078.
22. Huang Y, Niu B, Gao Y, *et al*. CD-HIT Suite: a web server for clustering and comparing biological sequences. *Bioinformatics* 2010;**26**(5):680–2.
23. UniProt Consortium. The universal protein resource (uniprot). *Nucleic Acids Res* 2010;**38**:D142–8.
24. Meyer DF, Noroy C, Moumène A, *et al*. Searching algorithm for type IV secretion system effectors 1.0: a tool for predicting type IV effectors and exploring their genomic context. *Nucleic Acids Res* 2013;**41**:9218–29.
25. Makino K, Oshima K, Kurokawa K, *et al*. Genome sequence of *Vibrio parahaemolyticus*: a pathogenic mechanism distinct from that of V. *cholerae*. *Lancet* 2003;**361**(9359):743–9.
26. Li K, Xu C, Jian H, *et al*. Prediction and identification of the effectors of heterotrimeric G proteins in rice (*Oryza sativa L.*). *Brief Bioinform* 2017;**18**:270–8.
27. Wang XB, Wu LY, Wang YC, *et al*. Prediction of palmitoylation sites using the composition of k-spaced amino acid pairs. *Protein Eng Des Sel* 2009;**22**(11):707–12.
28. Chen Z, Zhou Y, Song J, *et al*. hCKSAAP_UbSite: improved prediction of human ubiquitination sites by exploiting amino acid pattern and properties. *Biochim Biophys Acta* 2013;**1834**(8):1461–7.
29. Grynberg M, Godzik A. Sequence-based prediction of type III secreted proteins. *PLoS Pathog* 2009;**5**:e1000376.
30. Delorenzi M, Speed T. An HMM model for coiled-coil domains and a comparison with PSSM-based predictions. *Bioinformatics* 2002;**18**(4):617–25.
31. Kaur H, Raghava GPS. Prediction of alpha-turns in proteins using PSI-BLAST profiles and secondary structure information. *Proteins Struct Func Bioinform* 2004;**55**(1):83–90.
32. Kaur H, Raghava SPG. A neural network method for prediction of beta-turn types in proteins using evolutionary information. *Bioinformatics* 2004;**20**(16):2751–8.
33. Xie D, Li A, Wang M, *et al*. LOCSVMPSI: a web server for subcellular localization of eukaryotic proteins using SVM and profile of PSI-BLAST. *Nucleic Acids Res* 2005;**33**:105–10.
34. Liu T, Zheng X, Wang J. Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile. *Biochimie* 2010;**92**(10):1330–4.
35. Chen SA, Ou YY, Lee TY, *et al*. Prediction of transporter targets using efficient RBF networks with PSSM profiles and biochemical properties. *Bioinformatics* 2011;**27**(15):2062–7.
36. Dong Q, Zhou S, Guan J. A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. *Bioinformatics* 2009;**25**(20):2655–62.
37. Cheng-Wei C, Emily Chia-Yu S, Jenn-Kang H, *et al*. Predicting RNA-binding sites of proteins using support vector machines and evolutionary information. *BMC Bioinformatics* 2008;**12**:1–19.
38. Li J, Zhang Y, Qin W, *et al*. Using the improved position specific scoring matrix and ensemble learning method to predict drug-binding residues from protein sequences. *Nat Sci* 2012;**04**(05):304.
39. Wang J, Yang B, Revote J, *et al*. POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. *Bioinformatics* 2017;**33**:2756–8.
40. Chen K, Kurgan L. PFRES: protein fold classification by using evolutionary information and predicted secondary structure. *Bioinformatics* 2007;**23**(21):2843–50.
41. Gnad F, Ren S, Cox J, *et al*. PHOSIDA (phosphorylation site database): management, structural and evolutionary investigation, and prediction of phosphosites. *Genome Biol* 2007; **8**(11):561–70.
42. Song J, Yuan Z, Tan H, *et al*. Predicting disulfide connectivity from protein sequence using multiple sequence feature vectors and secondary structure. *Bioinformatics* 2007;**23**(23):3147–54.
43. Li T, Du P, Xu N, Uversky VN. Identifying human kinase-specific protein phosphorylation sites by integrating heterogeneous information from various sources. *PLoS One* 2010; **5**(11):419–53.
44. Mizianty MJ, Stach W, Chen K, *et al*. Improved sequence-based prediction of disordered regions with multilayer fusion of multiple information sources. *Bioinformatics* 2010;**26**(18):i489–96.
45. Song J, Tan H, Shen H, *et al*. Cascleave: towards more accurate prediction of caspase substrate cleavage sites. *Bioinformatics* 2010;**26**(6):752–60.
46. Magnan CN, Baldi P. SSpro/ACCpro 5: almost perfect prediction of protein secondary structure and relative solvent accessibility using profiles, machine learning, and structural similarity. *Bioinformatics* 2014;**30**(18):2592–7.
47. Dunker AK, Obradovic Z. The protein trinity-linking function and disorder. *Nat Biotechnol* 2001;**19**(9):805–6.
48. Ward JJ, Sodhi JS, Mcguffin LJ, *et al*. Prediction and functional analysis of native disorder in proteins from the three kingdoms of life. *J Mol Biol* 2004;**337**(3):635–45.

49. Radivojac P, Vacic V, Haynes C, *et al*. Identification, analysis, and prediction of protein ubiquitination sites. *Proteins Struct Funct Bioinform* 2010;**78**(2):365–80.
50. Aksoy S, Haralick RM. Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern Recog Lett* 2001;**22**(5):563–82.
51. Guyon I, Elisseeff A. An introduction to variable and feature selection. *J Mach Learn Res* 2002;**3**:1157–82.
52. Shannon CE. A mathematical theory of communication: the bell system technical journal. *Bell Syst Tech J* 1948;**27**(3):3–55.
53. Peng H, Long F, Ding C. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans Pattern Anal Mach Intell* 2005;**27**:1226–38.
54. Yi Z, Ding C, Tao L. Gene selection algorithm by combining reliefF and mRMR. *BMC Genomics* 2008;**9(Suppl 2)**:453–8.
55. Li BQ, Hu LL, Niu S, *et al*. Predict and analyze S-nitrosylation modification sites with the mRMR and IFS approaches. *J Proteomics* 2012;**75**(5):1654–65.
56. Jing W, Zhang D, Jing L. PREAL: prediction of allergenic protein by maximum Relevance Minimum Redundancy (mRMR) feature selection. *BMC Syst Biol* 2013;**7(Suppl 5)**:5028–38.
57. Li Y, Wang M, Wang H, *et al*. Accurate in silico identification of species-specific acetylation sites by integrating protein sequence-derived and functional features. *Sci Rep* 2015;**4**(1): 5765–5.
58. Wang H, Wang M, Tan H, *et al*. PredPPCrys: accurate prediction of sequence cloning, protein production, purification and crystallization propensity from protein sequences using multi-step heterogeneous feature fusion and selection. *PLoS One* 2014;**9**(8):e105902.
59. Wang M, Zhao XM, Tan H, *et al*. Cascleave 2.0, a new approach for predicting caspase and granzyme cleavage targets. *Bioinformatics* 2014;**30**:71–80.
60. Friedman N, Dan G, Goldszmidt M. Bayesian network classifiers. *Mach Learn* 1997;**29**(2/3):131–63.
61. Liang L, Djuric N, Guo Y, *et al*. MS- k NN: protein function prediction by integrating multiple data sources. *BMC Bioinformatics* 2013;**14(Suppl 3)**:61–4.
62. Chou KC, Shen HB. Predicting eukaryotic protein subcellular location by fusing optimized evidence-theoretic K-Nearest Neighbor classifiers. *J Proteome Res* 2006;**5**(8):1888–97.
63. Shen H, Chou KC. Using optimized evidence-theoretic K-nearest neighbor classifier and pseudo-amino acid composition to predict membrane protein types. *Biochem Biophys Res Commun* 2005;**334**(1):288–92.
64. Kim S-J, Koh K, Lustig M, *et al*. An interior-point method for large-scale l 1-regularized logistic regression. *IEEE J Sel Topics Sign Proces* 2007;**1**(4):1519–55.
65. Zardo P, Collie A. Predicting research use in a public health policy environment: results of a logistic regression analysis. *Implement Sci* 2014;**9**(1):1–10.
66. Breiman L. Random forest. *Mach Learn* 2001;**45**(1):5–32.
67. Liaw A, Wiener M. Classification and regression by randomforest. *R News* 2001;**23**.
68. Fern N-DM, Cernadas E, *et al*. Do we need hundreds of classifiers to solve real world classification problems? *J Mach Learn Res* 2014;**15**:3133–81.
69. Meyer D, Dimitriadou E, Hornik K, *et al*. Misc Functions of the Department of Statistics (e1071), TU Wien. R Package version 1.6-1, 2009.
70. Tu JV. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *J Clin Epidemiol* 1996;**49**(11):1225–31.
71. Bergmeir C, Benitez JM. Neural networks in R using the stuttgart neural network simulator: RSNNS. *J Stat Softw* 2012;**46**(7):1–26.
72. Petron E. Stuttgart neural network simulator: exploring connectionism and machine learning with SNNS. *Linux J* 1999;**1999**.
73. Matthews BW. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim Biophys Acta* 1975;**405**(2):442–51.
74. O'Shea JP, Church GM, Schwartz D, *et al*. pLogo: a probabilistic approach to visualizing sequence motifs. *Nat Methods* 2013; **10**:1211–12.
75. Jeong KC, Sutherland MC, Vogel JP. Novel export control of a Legionella Dot/Icm substrate is mediated by dual, independent signal sequences. *Mol Microbiol* 2015;**96**(1):175–88.
76. Shah AD, Bartlett JW, Carpenter J, *et al*. Comparison of random forest and parametric imputation models for imputing missing data using MICE: a CALIBER study. *Am J Epidemiol* 2014;**179**:179–74.
77. Saeys Y, Inza I, Larrañaga P. A review of feature selection techniques in bioinformatics. *Bioinformatics* 2007;**23**(19):2507–17.
78. Forgy EW. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics* 1965;**21**: 768–9.
79. Maaten L, Hinton G. Visualizing data using t-SNE. *J Mach Learn Res* 2008;**9**:2579–605.
80. Li W, Cowley A, Uludag M, *et al*. The EMBL-EBI bioinformatics web and programmatic tools framework. *Nucleic Acids Res* 2015;**43**(W1):W580–4.
81. Letunic I, Bork P. Interactive tree of life (iTOL) v3: an online tool for the display and annotation of phylogenetic and other trees. *Nucleic Acids Res* 2016;**44**(W1):W242–5. 24
82. Schroeder GN, Petty NK, Mousnier A, *et al*. Legionella pneumophila strain 130b possesses a unique combination of type IV secretion systems and novel Dot/Icm secretion system effector proteins. *J Bacteriol* 2010;**192**(22):6001–16.
83. Darby A, Lertpiriyapong K, Sarkar U, *et al*. Cytotoxic and pathogenic properties of Klebsiella oxytoca isolated from laboratory animals. *PLoS One* 2014;**9**(7):e100542.
84. Fodah RA, Scott JB, Tam HH, *et al*. Correlation of *Klebsiella pneumoniae* comparative genetic analyses with virulence profiles in a murine respiratory disease model. *PLoS One* 2014; **9**(9):e107394.
85. Luo Z-Q, Isberg RR. Multiple substrates of the *Legionella pneumophila* Dot/Icm system identified by interbacterial protein transfer. *Proc Natl Acad Sci USA* 2004;**101**(3):841–6.
86. Zusman T, Degtyar E, Segal G. Identification of a hypervariable region containing new *Legionella pneumophila* Icm/Dot translocated substrates by using the conserved icmQ regulatory signature. *Infect Immun* 2008;**76**(10):4581–91.
87. Bardill JP, Miller JL, Vogel JP. IcmS-dependent translocation of SdeA into macrophages by the *Legionella pneumophila* type IV secretion system. *Mol Microbiol* 2005;**56**(1):90–103.
88. Juhas M, Crook DW, Hood DW. Type IV secretion systems: tools of bacterial horizontal gene transfer and virulence. *Cell Microbiol* 2008;**10**(12):2377–86.
89. Burstein D, Amaro F, Zusman T, *et al*. Genomic analysis of 38 Legionella species identifies large and diverse effector repertoires. *Nat Genet* 2016;**48**(2):167–75.
90. Carver T, Thomson N, Bleasby A, *et al*. DNAPlotter: circular and linear interactive genome visualization. *Bioinformatics* 2009;**25**(1):119–20.
91. Krzywinski M, Schein J, Birol I, *et al*. Circos: an information aesthetic for comparative genomics. *Genome Res* 2009;**19**(9): 1639–45.

2.2

Bastion6: a bioinformatics approach for accurate prediction of type VI secreted effectors

The supplementary information for this manuscript is listed in **Appendix 2**.

Bioinformatics, 34(15), 2018, 2546–2555
doi: 10.1093/bioinformatics/bty155
Advance Access Publication Date: 14 March 2018
Original Paper

OXFORD

Genome analysis

Bastion6: a bioinformatics approach for accurate prediction of type VI secreted effectors

Jiawei Wang[1], Bingjiao Yang[2], André Leier[3], Tatiana T. Marquez-Lago[3], Morihiro Hayashida[4], Andrea Rocker[1], Yanju Zhang[2], Tatsuya Akutsu[5], Kuo-Chen Chou[6,7,8], Richard A. Strugnell[9], Jiangning Song[10,11,12,*] and Trevor Lithgow[1,*]

[1]Biomedicine Discovery Institute and Department of Microbiology, Monash University, Clayton, VIC 3800, Australia, [2]Bioinformatics Group, School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China, [3]Department of Genetics, School of Medicine, University of Alabama at Birmingham, Birmingham, AL, USA, [4]National Institute of Technology, Matsue College, Matsue, Shimane 690-8518, Japan, [5]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Kyoto 611-0011, Japan, [6]Gordon Life Science Institute, Boston, MA 02478, USA, [7]Center for Informational Biology, University of Electronic Science and Technology of China, Chengdu 610054, China, [8]Center of Excellence in Genomic Medicine Research (CEGMR), King Abdulaziz University, Jeddah 21589, Saudi Arabia, [9]Department of Microbiology and Immunology and Peter Doherty Institute for Infection and Immunity, The University of Melbourne, Parkville, VIC, Australia, [10]Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, [11]Monash Centre for Data Science, Faculty of Information Technolog and [12]ARC Centre of Excellence for Advanced Molecular Imaging, Monash University, Clayton, VIC 3800, Australia

*To whom correspondence should be addressed.
Associate Editor: John Hancock

Received on October 23, 2017; revised on February 26, 2018; editorial decision on March 8, 2018; accepted on March 9, 2018

Abstract

Motivation: Many Gram-negative bacteria use type VI secretion systems (T6SS) to export effector proteins into adjacent target cells. These secreted effectors (T6SEs) play vital roles in the competitive survival in bacterial populations, as well as pathogenesis of bacteria. Although various computational analyses have been previously applied to identify effectors secreted by certain bacterial species, there is no universal method available to accurately predict T6SS effector proteins from the growing tide of bacterial genome sequence data.

Results: We extracted a wide range of features from T6SE protein sequences and comprehensively analyzed the prediction performance of these features through unsupervised and supervised learning. By integrating these features, we subsequently developed a two-layer SVM-based ensemble model with fine-grain optimized parameters, to identify potential T6SEs. We further validated the predictive model using an independent dataset, which showed that the proposed model achieved an impressive performance in terms of ACC (0.943), F-value (0.946), MCC (0.892) and AUC (0.976). To demonstrate applicability, we employed this method to correctly identify two very recently validated T6SE proteins, which represent challenging prediction targets because they significantly differed from previously known T6SEs in terms of their sequence similarity and cellular function. Furthermore, a genome-wide prediction across 12 bacterial species, involving in total 54 212 protein sequences, was carried out to distinguish 94 putative T6SE candidates. We envisage both this information and our publicly accessible web server will facilitate future discoveries of novel T6SEs.

Availability and implementation: http://bastion6.erc.monash.edu/

Contact: jiangning.song@monash.edu or trevor.lithgow@monash.edu
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Gram-negative bacteria secrete proteins for a variety of cell survival purposes, and recently a sophisticated nanomachine called the type VI secretion system (T6SS) has been shown to function in delivering effector proteins (termed T6SEs) into neighboring cells that may be either eukaryotic or prokaryotic (Ho *et al.*, 2014; Mougous, 2006; Vettiger and Basler, 2016). In this way, the T6SS can be employed for host cell subversion and pathogenesis, and also to eliminate bacterial competitors. Multiple gene clusters have been discovered that encode components of the T6SS machinery, and are widespread among Gram-negative bacteria (Boyer *et al.*, 2009). Each T6SS has multiple conserved mechanisms for recruiting its associated effectors for secretion. In each case, effector recruitment involves direct or indirect association with the hemolysin co-regulated protein (Hcp) and valine-glycine repeat G (VrgG) or proline-alanine-alanine-arginine (PAAR) proteins of the T6SS, which are expelled together during the translocation events (Cianfanelli *et al.*, 2016).

Experimental methods for the discovery of T6SEs have primarily been discovery-driven, knowledge/hypothesis-based methodologies: specific analysis of T6SS-associated genes, proteomics-based methods and screens of mutant libraries (Lien and Lai, 2017). In addition, sequence-based analyses have been developed for predicting potential effector candidates from genome sequence. For instance, variant members of the VgrG and Hcp protein families with additional C-terminal domains are promising T6SE candidates (Cianfanelli *et al.*, 2016; Jamet and Nassif, 2015; Ma *et al.*, 2017a; Pukatzki *et al.*, 2009) with some characterized as T6SEs (Blondel *et al.*, 2009; Brooks *et al.*, 2013; Dong *et al.*, 2013; Flaugnatti *et al.*, 2016; Ma *et al.*, 2017a; Pukatzki *et al.*, 2007). Also, there is evidence of genetic linkage between the known T6SS chaperones, such as DUF4123 of Tap-1/TEC (Liang *et al.*, 2015) and DUF2169 (Bondage *et al.*, 2016; Liang *et al.*, 2015), and their cognate T6SE. More recently, conserved domains have been used to identify T6SEs: Rhs/YD repeat (Koskiniemi *et al.*, 2013; Ma *et al.*, 2017b; Murdoch *et al.*, 2011; Whitney *et al.*, 2014), PAAR (Ma *et al.*, 2014; Rigard *et al.*, 2016; Whitney *et al.*, 2014), TTR (Flaugnatti *et al.*, 2016; Shneider *et al.*, 2013) and MIX motifs (Salomon, 2016; Salomon *et al.*, 2014, 2015) have all been used as tools to identify tentative T6SEs. While these bioinformatics approaches have identified some T6SEs they are limited to, and highly dependent on, the existing knowledge of biochemical features and transport mechanisms of T6SEs.

We sought to develop a universal machine learning based method to accurately predict T6SS effector proteins. We extracted a wide variety of features from T6SEs based on their sequence profile, evolutionary information and physicochemical property, and comprehensively analyzed the prediction performance of these features using unsupervised and supervised learning. A set of SVM-based models was then developed for these features, assembled as a two-layer integrative to identify potential T6SEs, effectively and robustly. This ensemble model was further tested using (i) an independent dataset of 20 newly discovered T6SEs, and (ii) by assessment of two newly discovered and experimentally validated T6SEs. The results show that our proposed model achieved a much better performance in terms of ACC (0.943), F-value (0.946), MCC (0.892) and AUC (0.976) when compared with single feature based models, one-layer ensemble models and two motif-based searching methods. Additionally, by accurately recognizing new experimentally validated T6SEs, the proposed model demonstrated its effectivity and robustness toward identification of potential T6SEs. Furthermore, with our genome-wide prediction across 12 bacterial species, involving a total of 54 212 encoded protein sequences, we were able to identify 94 putative T6SE candidates. Lastly, we developed an online bioinformatics server, termed Bastion6 (Bacterial secretion effector predictor for type VI secretion system), to provide a user-friendly T6SE prediction service. To the best of our knowledge, Bastion6 is the first machine learning based predictor for T6SE prediction. We envisage this server will be widely used to facilitate discovery of novel T6SEs.

2 Materials and methods

An overview of the workflow of our Bastion6 methodology is illustrated in Figure 1. Briefly, three major stages are involved in the development of Bastion6: (i) sequence analysis based on the curated dataset; (ii) feature extraction, model training and construction and (iii) feature analysis, model parameterization and performance assessment using unsupervised analysis, supervised analysis and case study.

2.1 Data collection and preprocessing

To construct the training dataset, we extracted 178 known T6SE sequences from the SecretEPDB database (An *et al.*, 2017) and 1132 non-effectors from the literature (Zou *et al.*, 2013), and then removed highly homologous sequences at the threshold of 90% sequence identity due to limited positive samples. We finally obtained a training dataset containing 138 positive and 1112 negative protein sequences (Supplementary Fig. S1).

To further evaluate the performance of our proposed ensemble method, as compared with single feature based models and existing motif-based T6SE searching methods, we generated an independent dataset by extracting T6SEs from recently published works in the literature (Supplementary Table S1) and non-T6SEs from *Vibrio parahaemolyticus*. After highly homologous samples (with more than

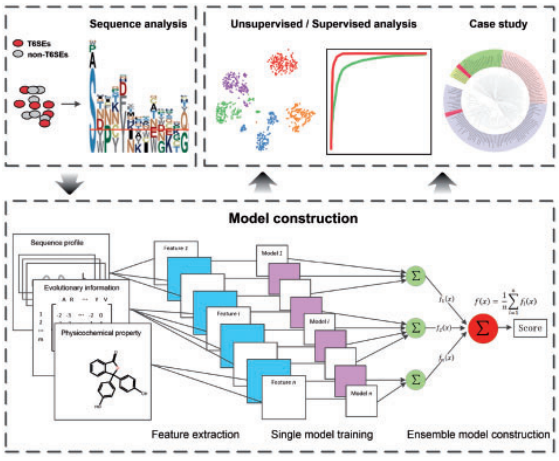


Fig. 1. Workflow of our developed Bastion6 approach

90% similarity) were removed from our training dataset, we obtained the final independent dataset with 20 positive and 200 negative samples. Aside, two very recently experimentally validated T6SEs (Lin *et al.*, 2017; Si *et al.*, 2017) were used as case studies to test the identifying capability of the proposed method.

2.2 Feature extraction

A protein's amino acid sequence contains important intrinsic information that dictates its properties. These include composition, permutation and combination modes of amino acids, orders of amino acids, similarities, homologies with other proteins, evolutionary information and physicochemical properties. While each type of feature may contribute to the characteristics of T6SEs, none of the features is predominant among all T6SEs, or indeed constitutes a sufficient and necessary determinant for a protein to be an effector. Thus, extracting features from a wide range of properties would better characterize T6SEs. In this work, we categorized this information into three groups: sequence profile, evolutionary information and physicochemical property.

2.2.1 Group 1: sequence-based features

Protein function is determined by the three-dimensional structure of the protein itself, which in turn depends on the primary structure, i.e. amino acid sequence (Anfinsen, 1972). Different proteins differ in the percentage compositions of amino acids, the modes of combination of amino acids, and the orders of amino acids. Accordingly, three types of sequence-derived features, including amino acid composition (AAC), dipeptide composition (DPC) and Quasi-Sequence-Order descriptors (QSO), were encoded to represent the above characteristics, respectively.

1. AAC is a widely used type of characterizing the occurrence frequencies of 20 amino acids in a sequence and can thus generate a 20-dimensional feature vector.
2. DPC describes the frequencies of dipeptides, each of which is made up of a pair of amino acids. It thus generates a 400-dimensional feature vector, which partially reflects the sequence order information and fragment information.
3. QSO (Chou, 2000) describes the sequence order effect based on the physicochemical distance between amino acids. The QSO descriptors of the sequence can be calculated as:

$$\begin{cases} X_r = \dfrac{f_r}{\sum_{r=1}^{20} f_r + \omega \sum_{d=1}^{maxlag} \tau_d}, \ r = 1, 2, \ldots, 20, \\ X_d = \dfrac{\omega \tau_{d-20}}{\sum_{r=1}^{20} f_r + \omega \sum_{d=1}^{maxlag} \tau_d}, \ d = 21, 22, \ldots, 20 + maxlag, \\ \tau_d = \sum_{i=1}^{N-d} (dist_{i,i+d})^2, d = 1, 2, \ldots, maxlag, \end{cases}$$

where f_r represents the normalized occurrence for amino acid r, $dist_{i,i+d}$ denotes the distance between the ith amino acid and the $(i+d)$th amino acid of the sequence, N counts the amino acids of the sequence, ω denotes the weighting factor and *maxlag* defines the maximum lag that should be no more than the length of the protein sequence. Accordingly, by applying these formulas into Schneider-Wrede physicochemical distance matrix (Schneider and Wrede, 1994) and another chemical distance matrix (Grantham, 1974), two feature vectors were obtained, each of which combines X_r and X_d in $20 + maxlag$ dimensions, with default values $\omega = 0.1$ and $maxlag = 30$.

2.2.2 Group 2: evolutionary information-based features

An increasing number of studies have shown that including evolutionary information is more informative than just sequence information alone (An *et al.*, 2018; Wang *et al.*, 2017a; Zou *et al.*, 2013). Accordingly, such information can serve as a basis for additional feature encodings (Wang *et al.*, 2017b):

1. The Blocks substitution matrix (BLOSUM) is a substitution matrix used to score local alignments between evolutionarily divergent protein sequences. Due to its usefulness it has been applied in many previous bioinformatics studies (Capra and Singh, 2008; Jones, 1999; Jones and Cozzetto, 2015; Wen *et al.*, 2016). In this work, we encoded a protein sequence by mapping its amino acids onto the BLOSUM62 matrix to retrieve the residue similarity values. Accordingly, we obtained a 175-dimentional feature vector.
2. A position-specific scoring matrix (PSSM) is a $L \times 20$ matrix, where L is the length of its corresponding protein sequence. The (i, j)th element of the matrix denotes the probability of amino acid j to appear at the ith position of the protein sequence (Wang *et al.*, 2017a). By borrowing the idea of a DPC encoding algorithm and applying it to a PSSM, DPC-PSSM is designed to partially express the local sequence-order effect (Liu *et al.*, 2010). As a result, DPC-PSSM is represented by a 400-dimentional feature vector, which utilizes the evolutionary information and, moreover, reflects the sequence-order information. DPC-PSSM can be calculated as:

$$\begin{cases} Y = \left(y_{1,1}, \ldots, y_{1,20}, y_{2,1}, \ldots, y_{2,20}, \ldots, y_{20,1}, \ldots, y_{20,20}\right)^T \\ y_{i,j} = \dfrac{1}{L-1} \sum_{k=1}^{L-1} p_{k,i} \times p_{k+1,j} \ (1 \le i, j \le 20) \end{cases}$$

where $p_{k,i}$ denotes the element at kth row and ith column of PSSM, and L denotes the row counts of the PSSM, which is equal to the length of the corresponding protein sequence.

3. S-FPSSM is designed to extract evolutionary information delicately based on the matrix transformation of the original PSSM (Zahiri *et al.*, 2013). The 'filtered' matrix FPSSM is produced from PSSM in a preprocessing step during which all negative elements of the PSSM are set to zero and all positive elements greater than an expected value δ (with a default value of 7) are set to δ. Consequently, all elements in FPSSM are in the range from 0 to δ. This step can help eliminate the negative elements' influence on the positive ones when adding two elements during matrix transformation. Based on the FPSSM, the resulting feature vector $S = (S_1^{(1)}, \ldots, S_{20}^{(1)}, \ldots, S_1^{(20)}, \ldots, S_{20}^{(20)})$ can be defined as follows:

$$s_j^{(i)} = \sum_{k=1}^{L} fp_{k,j} \times \delta_{k,i}$$

subject to

$$\begin{cases} \delta_{k,i} = 1, \ r_k = a_i \\ \delta_{k,i} = 0, \ r_k \neq a_i \end{cases} \ i, j = 1, \ldots, 20$$

where L denotes the total number of rows of the FPSSM, $fp_{k,i}$ denotes the element in the kth row and ith column of FPSSM, r_k

denotes the kth residue in the sequence, and a_i denotes the ith amino acid of 20 primary amino acids.

4. Pse-PSSM was originally proposed by Chou *et al.* and many empirical studies demonstrated its usefulness in protein sequence analysis (Chou and Shen, 2007). It is a reliable feature encoding method for extracting evolutionary information based on the PSSM transformation, and dimension normalization of the resulting feature vector. Pse-PSSM can be described using the following formulae:

$$mean_i = \frac{\sum_{k=1}^{20} E_{i,k}}{20},\ i = 1, 2, \ldots, L$$

$$STD_i = \sqrt{\frac{\sum_{u=1}^{20} (E_{i,u} - mean_i)}{20}},\ i = 1, 2, \ldots, L$$

$$T_{i,j} = \frac{E_{i,j} - mean_i}{STD_i},\ i = 1, 2, \ldots, L$$

$$H_j^{\alpha} = \frac{1}{L-\alpha}\sum_{i=1}^{L-\alpha}(T_{i,j} - T_{i+\alpha,j})^2$$

$$\bar{T_j} = \frac{1}{L}\sum_{i=1}^{L} T_{i,j}$$

$$T' = [\overline{T_1}, \ldots, \overline{T_{20}}]$$

$$H' = [H_1^{\alpha}, \ldots, H_{20}^{\alpha}]$$

$$P_{PsePSSM}^{\alpha} = [T', H']$$

where $E_{i,k}$ denotes the element in the ith row and kth column of the original PSSM, and L denotes the length of the protein sequence. Consequently, Pse-PSSM can be represented as a 40-dimentional feature vector, which reflects the relationship between an amino acid and its following αth amino acid in the sequence. In this work, we used the default value $\alpha = 1$.

2.2.3 Group 3: physicochemical features

We included two types of physicochemical properties [i.e. composition, transition and distribution (CTD)], composition among CTD (termed as CTDC) and transition among CTD (termed as CTDT) (Xiao *et al.*, 2015), which were previously designed to describe the global composition of amino acid properties in protein sequence (Dubchak *et al.*, 1995).

1. There are seven types of physicochemical properties in this work. For each property, 20 primary amino acids are categorized into 3 different classes, according to their attributes (Table 1). Thus, CTDC is represented as a 21-dimentional feature vector, obtained from a protein sequence, as follows:

$$C_A = \frac{n_A}{N},\ A = 1, 2, 3$$

where n_A denotes the number of amino acid type (class) A, and N denotes the sequence length.

2. CTDT is a representation of the frequency with which a type A residue is followed by a type B residue, or vice versa. Accordingly, CTDT is a 21-dimentional feature vector and can be calculated as follows:

$$\begin{cases} T_{AB} = \dfrac{n_{AB} + n_{BA}}{N-1} \\ T_{BC} = \dfrac{n_{BC} + n_{CB}}{N-1} \\ T_{CA} = \dfrac{n_{CA} + n_{AC}}{N-1} \end{cases}$$

where n_{AB} denotes the number of dipeptide AB in the sequence, and N denotes the length of the sequence.

2.3 Integrative model construction

To address the imbalanced classification problem, we constructed N ($N = 100$ in our setting) SVM classifiers and trained each of them with a different subset of the training dataset (Chen and Jeong, 2009). More specifically, to construct an individual classifier, all the positive samples and an equal number of negative samples randomly selected from the training dataset were combined as training samples. For each SVM classifier, we adopted the Gaussian radial basis kernel and performed a grid search to optimize the two parameters, Cost (C) and *Gamma* (γ), in the search space $\{2^{-10}, \ldots, 2^{10}\}$. Thus, for each feature, an ensemble SVM classifier (termed as single feature-based model) was generated by averaging the prediction scores of all the N SVM classifiers. In this way, the imbalanced classification problem is transformed and replaced by multiple balanced data classification problems.

Different features correspond to different properties of proteins and thus can be viewed as capturing distinct protein characteristics from various perspectives, thereby resulting in different data distributions (Chen and Jeong, 2009). Incorporating such knowledge may help improve the prediction performance, as compared to models that have been trained using a single feature only. For each group of features, the prediction scores of single feature based models are averaged to obtain a one-layer ensemble model. Lastly, prediction scores of these one-layer ensemble models (corresponding to different feature groups) are averaged to form an integrative two-layer ensemble model for the final prediction (Fig. 1).

2.4 Performance evaluation

To measure the performance of the proposed method, we carried out an unsupervised analysis, a supervised analysis (including 5-fold cross-validation and independent tests) and case studies. Five

Table 1. Classification of 20 standard amino acid types according to seven specific types of physicochemical properties

	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

performance measures including SN, SP, ACC, F-value and MCC were used. These are defined as follows:

$$SN = \frac{TP}{TP + FN}$$

$$SP = \frac{TN}{TN + FP}$$

$$ACC = \frac{TP + TN}{TP + FP + TN + FN}$$

$$F - value = 2 \times \frac{TP}{2TP + FP + FN}$$

$$MCC = \frac{(TP \times TN) - (FN \times FP)}{\sqrt{(TP + FN) \times (TN + FP) \times (TP + FP) \times (TN + FN)}}$$

where *TP*, *TN*, *FP* and *FN* denote the numbers of true positives, true negatives, false positives and false negatives, respectively.

3 Experimental results

3.1 Sequence analysis

One of the current tools for T6SE discovery is a motif-based search called MIX (marker for type six effectors) focused on N-terminal sequence similarities found in a sample of T6SEs from *Vibrio parahaemolyticus* (Salomon *et al.*, 2014), and together with other analysis has suggested common features may be present more broadly in the N- and C- terminal sequences of T6SEs (Lien and Lai, 2017). To test this hypothesis, a sequence analysis was conducted to characterize the amino acid occurrences on the first 50N-terminal and 50C-terminal positions of T6SEs (Supplementary Fig. S2A). The calculated amino acid frequencies show no indication for a strongly conserved sequence motif at either end of the proteins. Indeed, the only discernible position with a high conservation level (bit count twice as high as the second highest stack) is found at position 1 of the N-terminal sequences. However, a similarly high conservation is also found for non-effector proteins (Supplementary Fig. S2B), which can be distinguished at that position only by the relative abundance of lysine (K) and phenylalanine (F) residues and a depletion in proline (P) and arginine (R) residues. The C-terminal amino acids of both T6SEs and non-effectors show a distinctively even conservation distribution, indicating that none of the positions plays a major role in recognition. A more than twofold increase compared to the average stack height is only observed for the very last position of non-effector proteins, which is enriched in lysine (K) and glutamate (E), but depleted in leucine (L).

3.2 Unsupervised analysis

To intuitively visualize the effect of different feature encodings on the classification performance, we conducted an unsupervised analysis based on a randomly selected balanced dataset (due to the impossibility of visualizing all *N* balanced datasets) and demonstrated the value of such analysis to ascertain whether the extracted features can be used to effectively discriminate the T6SEs from non-effectors (Hulsman *et al.*, 2014). For each feature encoding, we mapped all the samples (including both positives and negatives) onto the 2D space (Fig. 2), so that the differences in the characterization of these samples would be represented by their mutual distances in space. Although the samples from both classes are not evenly distributed across the 2D map, the embedding didn't show a clear division into distinct subgroups. To further investigate this, the data was processed using *K*-means clustering. In this way, the samples in the picture were colored by their clustering labels, and shaped by the true labels. The classified distribution of the data samples in each cluster is shown as the bar chart in Figure 2 (with detailed results listed in Supplementary Table S2).

DPC-PSSM outperformed all other feature encoding methods: using DPC-PSSM, non-T6SEs dominated in Cluster 1 (accounting for 99.1%) while T6SEs dominated in Cluster 2 (accounting for 84.6%). The apparently higher division and low mixture rate of two classes of samples in each cluster strongly demonstrated the ability of this encoding scheme to recognize the T6SEs from non-effectors. Following DPC-PSSM, DPC, AAC and Pse-PSSM achieved a good, comparable performance, with a moderate mixture rate within each cluster. The good performances of these four encoding methods illustrate that evolutionary information-based and sequence-based features contribute the most to T6SE classification.

Note that although T6SEs dominated in Cluster 2 (96.9%) for BLOSUM, there was a considerable number of T6SEs and non-T6SEs aggregated together in Cluster 1 (43.9% of T6SEs and 56.1% of non-T6SEs). Moreover, there was an imbalance between Cluster1 (containing 244 samples) and Cluster 2 (containing 32 samples) which could potentially impact the classification outcome.

3.3 Supervised analysis

We further evaluated the effect of each feature encoding in a supervised setting, enabling us to quantitatively assess them by using a set of standard measures on 5-cross validation and independent tests. All 5-fold cross validation tests in this work were conducted based on *N* ($N = 100$ in our setting) balanced training datasets, and the performance was averaged over these *N* balanced datasets.

3.3.1 Performance evaluation using 5-fold cross-validation tests

For each feature encoding method, an SVM classifier was trained with optimally-tuned parameters and validated based on the training dataset by performing randomized 5-fold cross-validation tests. The averaged results are shown in Table 2 and Figure 3A.

As can be seen, PSSM-based features achieved the overall best performance in terms of ACC (>0.91), F-value (>0.91), MCC (>0.83) and AUC (>0.96) (Table 2 and Fig. 3A). This suggested that PSSM-based features were the most informative for T6SE classification, and its related features were considered as essential for building accurate models. These observations agree well with previous bioinformatics studies (An *et al.*, 2018; Wang *et al.*, 2017a; Zou *et al.*, 2013). DPC-PSSM was shown to be the most powerful feature encoding method, which consistently achieved the highest values of SN (0.950), ACC (0.938), F-value (0.940), MCC (0.878) and AUC (0.983). These results are in accordance with those in our unsupervised analysis. Similarly, following the PSSM-based feature encoding, AAC achieved the second-best performance reflected by the ACC (0.873), F-value (0.872), MCC (0.748) and AUC (0.943). The poorer performance of BLOSUM indicates that the substitution matrix was less informative when compared with the PSSM, although the former is more accessible and can be directly calculated. The same holds for CTDT, which yielded only a moderate performance, despite it providing a novel perspective on the feature extraction of protein sequences. These results suggest that BLOSUM and CTDT can be used as complementary encoding schemes in conjunction with the essential PSSM features.

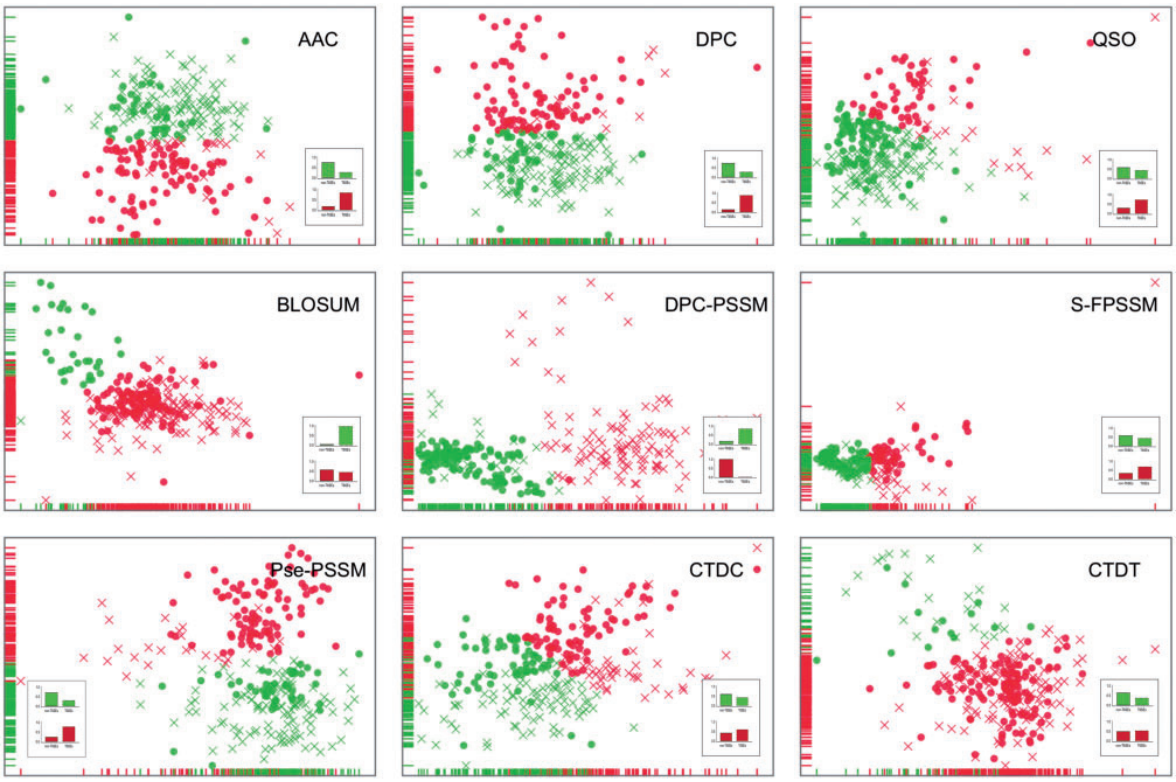


Fig. 2. Representation and clustering of data samples of T6SEs and non-T6SEs based on nine different types of feature encodings. For each encoding, the representation of data samples is presented in two dimensions after dimensionality reduction using principal component analysis (PCA). Samples were then clustered into two groups using the *K*-means algorithm; each cluster (represented by one color) consists of two types of samples (i.e. T6SEs and non-T6SEs) with two different shapes, in which circle and multiplication signs represent T6SEs and non-T6SEs, respectively. The classified distribution of T6SEs (right-hand bar) vs. non-T6SEs (left-hand bar) in each cluster is shown as the inset bar chart

Table 2. The performance of SVM classifiers using different sequence encoding methods based on 5-fold cross-validation tests

	Encoding	SN	SP	ACC	F-value	MCC
Group 1	AAC	0.871±0.022	0.875±0.028	0.873±0.020	0.872±0.020	0.748±0.041
	DPC	0.837±0.020	0.852±0.027	0.843±0.020	0.841±0.020	0.689±0.039
	QSO	0.843±0.020	0.863±0.027	0.851±0.018	0.849±0.018	0.706±0.036
Group 2	BLOSUM	0.810±0.034	0.796±0.031	0.802±0.024	0.803±0.025	0.608±0.048
	DPC-PSSM	**0.950±0.020**	0.929±0.019	**0.938±0.013**	**0.940±0.013**	**0.878±0.025**
	S-FPSSM	0.915±0.014	0.918±0.020	0.914±0.012	0.915±0.012	0.831±0.024
	Pse-PSSM	0.925±0.015	**0.944±0.019**	0.932±0.012	0.933±0.012	0.868±0.023
Group 3	CTDC	0.857±0.025	0.847±0.033	0.850±0.021	0.851±0.020	0.705±0.042
	CTDT	0.774±0.031	0.764±0.030	0.771±0.024	0.767±0.026	0.544±0.049

Note: The values were expressed as mean ± standard deviation. For each metric, the best performance value across different encoding methods is highlighted in bold for clarification.

Our supervised analysis also revealed differences with respect to the unsupervised analysis. In particular, we found that CTDC and QSO achieved an equivalent performance as the second-best feature encoding methods (with a performance that was slightly better than that of DPC). This suggests that the performance of individual encoding schemes may depend on the machine learning method being applied.

Generally, while there is a preference for high SN and SP values, a trade-off between SN and SP is necessary for a predictor to achieve a comprehensive and stable performance. Otherwise, it could generate predictions that are biased by a preference for a certain class of samples. In this work, the gaps between SN and SP were minor across all the encoding methods, which formed a solid basis for our model to achieve a stable performance over all metrics, including ACC, F-value, MCC and AUC.

***3.3.2* Performance evaluation using various sequence similarity rates**

Considering that the features used for training the models were derived from protein sequences, the training datasets curated with different sequence similarity cut-offs could result in different model performances. To examine the effect of the sequence similarity cut-off on the overall performance of the models, six sequence identity thresholds (i.e. 70, 75, 80, 85, 90 and 95%) were applied when

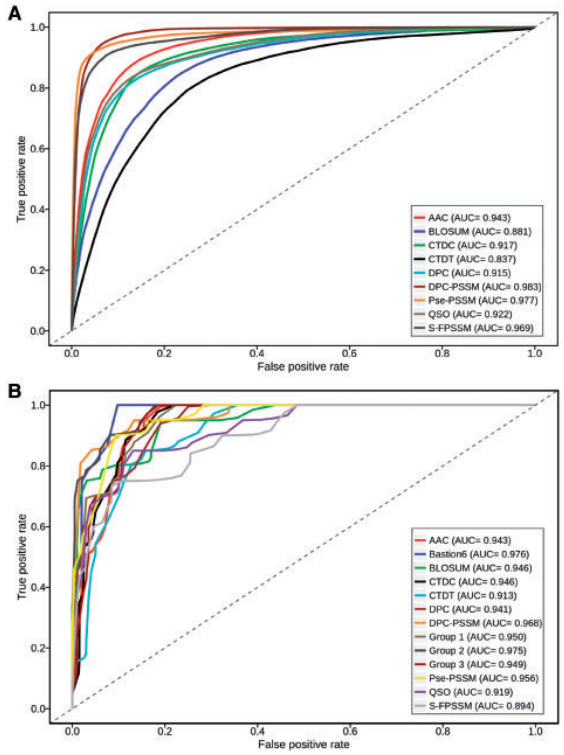


Fig. 3. (**A**) ROC curves of different feature encoding methods for T6SS effector prediction based on 5-fold cross-validation tests; (**B**) ROC curves of single feature-based models, one-layer models and the final model used by Bastion6 on the independent test. The results were distinguished by color curves. AUC values for each model are also presented

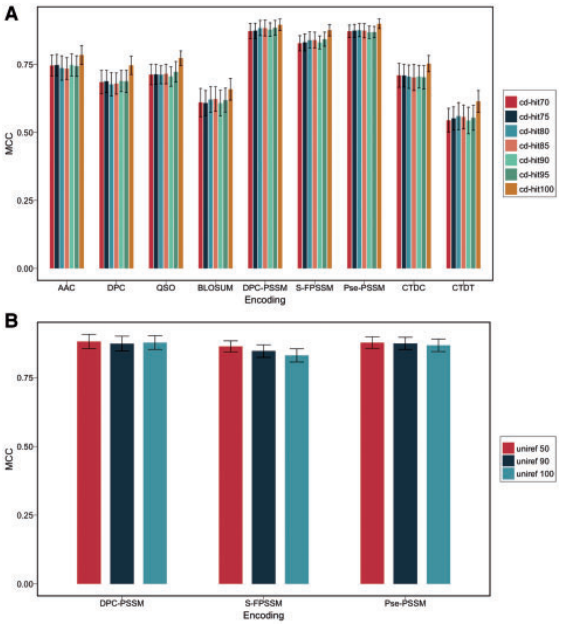


Fig. 4. (**A**) Performance of various feature encoding methods using different sequence similarity cut-offs based on 5-fold cross-validation tests; (**B**) Performance of various PSSM-based feature encoding methods against different uniref databases based on 5-fold cross-validation test

constructing training datasets. Using these generated datasets and the original dataset (without homologous sequence reduction), the performance of each model was evaluated using the same fivefold cross-validation. As can be seen from Figure 4A, in all cases, the models trained with the original dataset outperformed those trained with other datasets (i.e. after removal of homologous sequences) in terms of the MCC value. This suggests that high sequence homology in the original dataset can indeed lead to overestimated performances of the corresponding models, thereby highlighting the importance and necessity of performing sequence homology reduction prior to model training. However, models trained with datasets resulting from different sequence similarity cut-offs show a similar performance, indicating the robustness of the proposed models.

3.3.3 The effect of searched databases for PSSM-based features

To characterize the potential effect of the size of searched databases on the performance of PSSM-based models, we further generated PSSM profiles by searching against three uniref databases with different sizes (i.e. uniref50, uniref90 and uniref100) with parameters of $j = 3$ and $h = 0.001$. Based on these PSSM profiles, PSSM-based models were trained and performance evaluated using the same fivefold cross-validation procedure. The results indicate that there was no significant difference in the performance between these PSSM-based models (Fig. 4B), suggesting that the size of searched databases did not have a significant impact on the performance of the PSSM-based models on the curated T6SE dataset.

3.3.4 The effect of various selected features on the model performance

GainRatio (Frank *et al.*, 2004) was applied to conduct a set of feature selection experiments using the same fivefold cross-validation. We found that for different types of features, models trained using the entire features generally resulted in a better predictive performance compared to models trained using selected features (such as the top 50, 100, 150, 200, 250, 300 and 350 features) (Supplementary Fig. S3). The only exception was the BLOSUM-based model, which achieved a similar performance when compared to the corresponding model trained using selected features. A possible explanation is that the original size of each generated feature set was so small (i.e. less than 400 dimensions) that all features in the feature set without further selection could be interpreted well by machine learning methods, contributing to the models' overall performance.

3.3.5 Comparison with homology-based baseline predictor

To compare with the proposed models, we applied a homology-based approach to develop a baseline predictor. For each query sequence in the test set, the blastp program—implemented in the Blast+ software (Camacho *et al.*, 2009)—was used to search against the training dataset. Based on the blastp search results, the query sequence was assigned the same label as that of the top ranked protein sequence with the lowest E-value in the training dataset. We thus assessed the performance of this homology-based baseline predictor using the same fivefold cross-validation. The results showed that the baseline predictor achieved a lower performance with an F-value of 0.787, an ACC of 0.741 and an MCC of 0.517, than our proposed models. An explanation is that the homology-based baseline predictor could not recognize valuable patterns beyond the sequence identity, thus resulting in an unsatisfactory performance compared with our machine learning-based models.

3.3.6 **Performance validation on the independent test**

Using the independent test, the proposed two-layer ensemble model was further assessed, and benchmarked against the single feature-based, one-layer ensemble models. All experiments were conducted 10 times. Each time, a balanced independent dataset was formed by the positive samples and 20 randomly chosen negative samples. As shown in Figure 3B and Supplementary Table S3, most of the ensemble models display a better and more stable performance in terms of ACC, F-value, MCC and AUC, when compared to their single feature-based models, while Bastion6 achieved the best performance among them with respect to ACC (0.943), F-value (0.946), MCC (0.892) and AUC (0.976).

To measure the ability of positive sample identification, we further looked into the numbers of true positives predicted by various models in the independent test. Bastion6 outperformed the single feature-based models and one-layer ensemble models (Supplementary Table S4), without misclassifying any T6SE. In contrast, single feature-based models misclassified a larger number of T6SEs. As expected, ensemble models were able to correct the misclassifications of single feature-based models, and consequently achieved more stable performances.

Two previous motif search-based methods were assessed as a benchmark for the independent test, since motif strategies referred to as MIX and SAVC (Secretome analysis of *Vibrio cholera*) were recently used to discover T6SEs (Altindis *et al.*, 2015; Salomon *et al.*, 2014). Regarding the capability of recognizing T6SEs, Bastion6 successfully retrieved 20 positive samples, while MIX and SAVC retrieved 0 and 2 positive samples, respectively, from 20 T6SEs of the independent dataset (Supplementary Table S5). This result suggested that motif-based searching methods do not function well across bacterial species, and demonstrated the usefulness and necessity of our universal and highly accurate T6SE prediction method.

3.4 Case study

To examine the scalability and robustness of the proposed method, we carried out a case study using two very recent experimentally validated T6SEs: neither of these effectors was present in the training dataset, and both differ significantly from all other proteins in the training dataset (Supplementary Figs S4 and S5). Detailed prediction results are listed in Supplementary Table S6.

Our first case study protein was TseM (Si *et al.*, 2017), a T6SS-4–dependent Mn^{2+}-binding effector experimentally characterized from *Burkholderia thailandensis*. The proposed model correctly identified TseM as a T6SE, with a probability score of 0.544. As a comparison, models trained using sequence-based features generated lower probability scores (<0.5) due to the low sequence similarity between TseM and the protein sequences in the training dataset (Supplementary Figs S4 and S5). Models trained using PSSM (except S-FPSSM) and physicochemical properties could correctly recognize TseM as a T6SE with higher prediction scores. More specifically, the CTDT model correctly predicted this protein with the highest score of 0.763, despite its poorer performance in benchmarking experiments.

The second case study was the T6SE TseF recently identified in *Pseudomonas aeruginosa* (Lin *et al.*, 2017). TseF is secreted by the H3-T6SS, and then incorporated into outer membrane vesicles to facilitate the uptake of iron (Lin *et al.*, 2017). The proposed model successfully predicted TseF as a T6SE with a score of 0.681. Surprisingly, DPC-PSSM and Pse-PSSM models, which performed best in benchmarking experiments, failed to predict this T6SE. This highlights the necessity of exploiting the different but complementary feature encoding schemes that can capture useful 'signals' from different perspectives.

These results confirm the usefulness and reliability of our proposed method, and the value of integrating various models into ensemble learning models. By taking all these single models into account, the developed two-layer model achieved balanced predictive power, thus providing a reliable tool for identifying novel potential T6SEs.

3.5 Genome-scale prediction across various species

Currently, there are only a limited number of experimentally validated T6SEs. This has restricted our understanding of the functional roles in their interactions with their eukaryotic hosts or prokaryotic competitors. To facilitate the functional characterization, we performed a genome-wide prediction of T6SEs in 12 different bacterial species, including those that have been previously shown to possess T6SEs. As a result, a total of 94 putative T6SEs (with probability scores larger than 0.9) were extracted from 54 212 protein sequences. A statistical summary of the genome-wide prediction results is listed in Supplementary Table S7. A full list of the predicted T6SEs can be found at the Bastion6 server.

4 Discussion

Identification of T6SEs is a key to understanding the role of T6SS in bacteria's anti-bacterial competition, inter-bacterial interaction and virulence to their eukaryotic hosts (Ho *et al.*, 2014). Bacterial genome sequencing is advancing at an unprecedented pace and, consequently, rapid and accurate identification of T6SEs from genome sequence data is both achievable and highly desirable. Previous studies have reported motifs in N- or C-terminal sequences in some bacterial (Lien and Lai, 2017) suggested to define T6SEs. However, these motifs prove to be specific to a subset of T6SEs in only certain bacterial species. The latter was shown through sequence analysis and further validated in the benchmark tests in this work. To provide highly accurate prediction of T6SEs in and across diverse bacterial species, we extracted nine widely used features based on amino acid sequence information, evolutionary information and physicochemical properties. These features have been systematically and comprehensively assessed through unsupervised and supervised learning. The features demonstrated their effectiveness in different scenarios. PSSM-based features achieved the overall best performance in most cases. They could accurately predict novel T6SEs especially in cases where they significantly differ from known effectors. However, we also noticed that in some cases, PSSM-based features did not perform well while other features performed better on independent tests and case studies. There might be several reasons for this. First, compared to the vast number of uncharacterized effectors the dataset of known T6SEs was very limited when it comes to extracting sufficient knowledge and useful patterns. Accordingly, it was hard to quantitatively assess how a feature performs, relative to other features. Second, different features may be suitable for predicting different T6SEs. A feature-based model may be good at recognizing a subset of T6SEs while it fails to identify another subset of T6SEs. Therefore, taking advantage of all single feature-based models and integrating them into an ensemble model helps to improve the prediction of T6SEs.

The relatively small number of T6SE samples in the benchmark dataset will likely result in some bias in the prediction performance. However, the discovery of new T6SEs: bioinformatically, genetically and through other experimental approaches, will expand the

benchmark dataset and, accordingly, improve the model by lessening any potential bias. Additionally, other features that have proved useful in other bioinformatics studies (such as structure-based features and GO-based features) may help identify new patterns and improve the model once more T6SE data becomes available.

In this study, we have developed Bastion6, a two-layer ensemble machine learning method integrating a number of individual SVM-based models. Extensive benchmarking experiments validated the effectiveness and robustness of our proposed model. We further applied Bastion6 to perform genome-wide predictions and obtained a list of high-confidence, putative T6SEs in 54 212 proteins across 12 bacterial species. With these promising results, we believe our predicted T6SEs can serve as a preliminary screen for follow-up experiments. In addition, we implemented a publicly accessible web server, to meet users' specific demands. We believe that our proposed method can be a vastly useful tool for T6SE prediction, and will expedite the discovery of novel T6SEs.

Acknowledgements

We thank Dr. Jonathan Wilksch, Dr. Romé Voulhoux and Dr. Badreddine Douzi for critical comments on the manuscript.

Funding

This work was supported by grants from the National Health and Medical Research Council of Australia (NHMRC) (1092262), the Australian Research Council (ARC), the National Institute of Allergy and Infectious Diseases of the National Institutes of Health (R01 AI111965) and the Natural Science Foundation of Guangxi Under No. 2016GXNSFCA380005. AL and TML were supported by informatics startup packages through the UAB School of Medicine. T.L. is an ARC Australian Laureate Fellow (FL130100038).

Conflict of Interest: none declared.

References

Altindis,E. *et al.* (2015) Secretome analysis of *Vibrio cholerae* type VI secretion system reveals a new effector-immunity pair. *mBio*, **6**, e00075-15.

An,Y. *et al.* (2017) SecretEPDB: a comprehensive web-based resource for secreted effector proteins of the bacterial types III, IV and VI secretion systems. *Sci. Rep.*, **7**, 41031.

An,Y. *et al.* (2018) Comprehensive assessment and performance improvement of effector protein predictors for bacterial secretion systems III, IV and VI. *Brief. Bioinf.*, **19**, 148–161.

Anfinsen,C. (1972) The formation and stabilization of protein structure. *Biochem. J.*, **128**, 737.

Blondel,C.J. *et al.* (2009) Comparative genomic analysis uncovers 3 novel loci encoding type six secretion systems differentially distributed in Salmonella serotypes. *BMC Genomics*, **10**, 354.

Bondage,D.D. *et al.* (2016) VgrG C terminus confers the type VI effector transport specificity and is required for binding with PAAR and adaptor-effector complex. *Proc. Natl. Acad. Sci. USA*, **113**, E3931–E3940.

Boyer,F. *et al.* (2009) Dissecting the bacterial type VI secretion system by a genome wide in silico analysis: what can be learned from available microbial genomic resources? *BMC Genomics*, **10**, 104.

Brooks,T.M. *et al.* (2013) Lytic activity of the Vibrio cholerae type VI secretion toxin VgrG-3 is inhibited by the antitoxin TsaB. *J. Biol. Chem.*, **288**, 7618–7625.

Camacho,C. *et al.* (2009) BLAST+: architecture and applications. *BMC Bioinformatics*, **10**, 421.

Capra,J.A. and Singh,M. (2008) Characterization and prediction of residues determining protein functional specificity. *Bioinformatics*, **24**, 1473–1480.

Chen,X.W. and Jeong,J.C. (2009) Sequence-based prediction of protein interaction sites with an integrative method. *Bioinformatics*, **25**, 585–591.

Chou,K.C. (2000) Prediction of protein subcellular locations by incorporating quasi-sequence-order effect. *Biochem. Biophys. Res. Commun.*, **278**, 477–483.

Chou,K.C. and Shen,H.B. (2007) MemType-2L: a web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. *Biochem. Biophys. Res. Commun.*, **360**, 339–345.

Cianfanelli,F.R. *et al.* (2016) Aim, Load, Fire: the Type VI secretion system, a bacterial nanoweapon. *Trends Microbiol.*, **24**, 51–62.

Dong,T.G. *et al.* (2013) Identification of T6SS-dependent effector and immunity proteins by Tn-seq in *Vibrio cholerae*. *Proc. Natl. Acad. Sci. USA*, **110**, 2623–2628.

Dubchak,I. *et al.* (1995) Prediction of protein folding class using global description of amino acid sequence. *Proc. Natl. Acad. Sci.*, **92**, 8700–8704.

Flaugnatti,N. *et al.* (2016) A phospholipase A1 antibacterial Type VI secretion effector interacts directly with the C-terminal domain of the VgrG spike protein for delivery. *Mol. Microbiol.*, **99**, 1099–1118.

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

Grantham,R. (1974) Amino acid difference formula to help explain protein evolution. *Science*, **185**, 862–864.

Ho,B.T. *et al.* (2014) A view to a kill: the bacterial type VI secretion system. *Cell Host Microbe*, **15**, 9–21.

Hulsman,M. *et al.* (2014) Scale-space measures for graph topology link protein network architecture to function. *Bioinformatics*, **30**, i237–i245.

Jamet,A. and Nassif,X. (2015) New players in the toxin field: polymorphic toxin systems in bacteria. *mBio*, **6**, e00285-15–e00215.

Jones,D.T. (1999) GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. *J. Mol. Biol.*, **287**, 797–815.

Jones,D.T. and Cozzetto,D. (2015) DISOPRED3: precise disordered region predictions with annotated protein-binding activity. *Bioinformatics*, **31**, 857–863.

Koskiniemi,S. *et al.* (2013) Rhs proteins from diverse bacteria mediate intercellular competition. *Proc. Natl. Acad. Sci. USA*, **110**, 7032–7037.

Liang,X. *et al.* (2015) Identification of divergent type VI secretion effectors using a conserved chaperone domain. *Proc. Natl. Acad. Sci. USA*, **112**, 9106–9111.

Lien,Y.W. and Lai,E.M. (2017) Type VI Secretion Effectors: methodologies and Biology. *Front. Cell. Infect. Microbiol.*, **7**, 254.

Lin,J. *et al.* (2017) A Pseudomonas T6SS effector recruits PQS-containing outer membrane vesicles for iron acquisition. *Nat. Commun.*, **8**, 14888.

Liu,T. *et al.* (2010) Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile. *Biochimie*, **92**, 1330–1334.

Ma,J. *et al.* (2017a) The Hcp proteins fused with diverse extended-toxin domains represent a novel pattern of antibacterial effectors in type VI secretion systems. *Virulence*, 1–14.

Ma,J. *et al.* (2017b) PAAR-Rhs proteins harbor various C-terminal toxins to diversify the antibacterial pathways of type VI secretion systems. *Environ. Microbiol.*, **19**, 345–360.

Ma,L.S. *et al.* (2014) Agrobacterium tumefaciens deploys a superfamily of type VI secretion DNase effectors as weapons for interbacterial competition in planta. *Cell Host Microbe*, **16**, 94–104.

Mougous,J.D. (2006) A virulence locus of Pseudomonas aeruginosa encodes a protein secretion apparatus. *Science*, **312**, 1526–1530.

Murdoch,S.L. *et al.* (2011) The opportunistic pathogen Serratia marcescens utilizes type VI secretion to target bacterial competitors. *J. Bacteriol.*, **193**, 6057–6069.

Pukatzki,S. *et al.* (2007) Type VI secretion system translocates a phage tail spike-like protein into target cells where it cross-links actin. *Proc. Natl. Acad. Sci. USA*, **104**, 15508–15513.

Pukatzki,S. *et al.* (2009) The type VI secretion system: translocation of effectors and effector-domains. *Curr. Opin. Microbiol.*, **12**, 11–17.

Rigard,M. *et al.* (2016) *Francisella tularensis* IglG belongs to a novel family of PAAR-like T6SS proteins and harbors a unique N-terminal extension required for virulence. *PLoS Pathogens*, **12**, e1005821.

Salomon,D. (2016) MIX and match: mobile T6SS MIX-effectors enhance bacterial fitness. *Mobile Genet. Elements*, **6**, e1123796.

Salomon,D. *et al*. (2014) Marker for type VI secretion system effectors. *Proc. Natl. Acad. Sci. USA*, **111**, 9271–9276.

Salomon,D. *et al*. (2015) Type VI secretion system toxins horizontally shared between marine bacteria. *PLoS Pathogens*, **11**, e1005128.

Schneider,G. and Wrede,P. (1994) The rational design of amino acid sequences by artificial neural networks and simulated molecular evolution: de novo design of an idealized leader peptidase cleavage site. *Biophys. J.*, **66**, 335–344.

Shneider,M.M. *et al*. (2013) PAAR-repeat proteins sharpen and diversify the type VI secretion system spike. *Nature*, **500**, 350–353.

Si,M. *et al*. (2017) Manganese scavenging and oxidative stress response mediated by type VI secretion system in Burkholderia thailandensis. *Proc. Natl. Acad. Sci. USA*, **114**, E2233–E2242.

Vettiger,A. and Basler,M. (2016) Type VI secretion system substrates are transferred and reused among sister cells. *Cell*, **167**, 99–110 e112.

Wang,J. *et al*. (2017a) Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches. *Brief. Bioinf*, doi: 10.1093/bib/bbx164.

Wang,J. *et al*. (2017b) POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. *Bioinformatics*, **33**, 2756–2758.

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

Whitney,J.C. *et al*. (2014) Genetically distinct pathways guide effector export through the type VI secretion system. *Mol. Microbiol*., **92**, 529–542.

Xiao,N. *et al*. (2015) protr/ProtrWeb: r package and web server for generating various numerical representation schemes of protein sequences. *Bioinformatics*, **31**, 1857–1859.

Zahiri,J. *et al*. (2013) PPIevo: protein-protein interaction prediction from PSSM based evolutionary information. *Genomics*, **102**, 237–242.

Zou,L. *et al*. (2013) Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles. *Bioinformatics*, **29**, 3135–3142.

2.3

Bastion3: a two-layer ensemble predictor of type III secreted effectors

The supplementary information for this manuscript is listed in **Appendix 3**.

Bioinformatics, 2018, 1–12
doi: 10.1093/bioinformatics/bty914
Advance Access Publication Date: 2 November 2018
Original Paper

OXFORD

Sequence analysis

Bastion3: a two-layer ensemble predictor of type III secreted effectors

Jiawei Wang[1], Jiahui Li[1,2], Bingjiao Yang[3], Ruopeng Xie[3], Tatiana T. Marquez-Lago[4,5], André Leier[4,5], Morihiro Hayashida[6], Tatsuya Akutsu[7], Yanju Zhang[3], Kuo-Chen Chou[8,9,10], Joel Selkrig[11,*], Tieli Zhou[2,*], Jiangning Song[12,13,14,*] and Trevor Lithgow[1,*]

[1]Infection and Immunity Program, Biomedicine Discovery Institute and Department of Microbiology, Monash University, Melbourne, VIC 3800, Australia, [2]Department of Clinical Laboratory, The First Affiliated Hospital of Wenzhou Medical University, Wenzhou 325000, China, [3]School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China, [4]Department of Genetics and [5]Department of Cell, Developmental and Integrative Biology, School of Medicine, University of Alabama at Birmingham, AL, USA, [6]National Institute of Technology, Matsue College, Matsue, Shimane 690-8518, Japan, [7]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Kyoto 611-0011, Japan, [8]Gordon Life Science Institute, Boston, MA 02478, USA, [9]Center for Informational Biology, University of Electronic Science and Technology of China, Chengdu 610054, China, [10]Center of Excellence in Genomic Medicine Research (CEGMR), King Abdulaziz University, Jeddah 21589, Saudi Arabia, [11]European Molecular Biology Laboratory, Genome Biology Unit, Heidelberg, Germany, [12]Infection and Immunity Program, Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University, Melbourne, VIC 3800, Australia, [13]Monash Centre for Data Science, Monash University, Melbourne, VIC 3800, Australia and [14]ARC Centre of Excellence in Advanced Molecular Imaging, Monash University, Melbourne, VIC 3800, Australia

*To whom correspondence should be addressed.

Associate Editor: John Hancock

Received on August 31, 2018; revised on October 15, 2018; editorial decision on October 26, 2018; accepted on October 31, 2018

Abstract

Motivation: Type III secreted effectors (T3SEs) can be injected into host cell cytoplasm via type III secretion systems (T3SSs) to modulate interactions between Gram-negative bacterial pathogens and their hosts. Due to their relevance in pathogen–host interactions, significant computational efforts have been put toward identification of T3SEs and these in turn have stimulated new T3SE discoveries. However, as T3SEs with new characteristics are discovered, these existing computational tools reveal important limitations: (i) most of the trained machine learning models are based on the N-terminus (or incorporating also the C-terminus) instead of the proteins' complete sequences, and (ii) the underlying models (trained with classic algorithms) employed only few features, most of which were extracted based on sequence-information alone. To achieve better T3SE prediction, we must identify more powerful, informative features and investigate how to effectively integrate these into a comprehensive model.

Results: In this work, we present Bastion3, a two-layer ensemble predictor developed to accurately identify type III secreted effectors from protein sequence data. In contrast with existing methods that employ single models with few features, Bastion3 explores a wide range of features, from various types, trains single models based on these features and finally integrates these models through ensemble learning. We trained the models using a new gradient boosting machine, LightGBM and further boosted the models' performances through a novel genetic algorithm (GA)

based two-step parameter optimization strategy. Our benchmark test demonstrates that Bastion3 achieves a much better performance compared to commonly used methods, with an ACC value of 0.959, *F*-value of 0.958, MCC value of 0.917 and AUC value of 0.956, which comprehensively outperformed all other toolkits by more than 5.6% in ACC value, 5.7% in *F*-value, 12.4% in MCC value and 5.8% in AUC value. Based on our proposed two-layer ensemble model, we further developed a user-friendly online toolkit, maximizing convenience for experimental scientists toward T3SE prediction. With its design to ease future discoveries of novel T3SEs and improved performance, Bastion3 is poised to become a widely used, state-of-the-art toolkit for T3SE prediction.
Availability and implementation: http://bastion3.erc.monash.edu/
Contact: selkrig@embl.de or wyztli@163.com or jiangning.song@monash.edu or trevor.lithgow@monash.edu
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Type III secretion systems (T3SSs) are central to many host–pathogen interactions, as one of the major means for the secretion of effector proteins into host cells (Deng *et al.*, 2017; Galan and Waksman, 2018). These effectors are diverse proteins by size and sequence, and they function to mimic factors in diverse host cell functions in order to pervert host cell biology to the advantage of the bacterium (Deng *et al.*, 2017; Galan *et al.*, 2014; Jennings *et al.*, 2017; Raymond *et al.*, 2013). In the biomedical arena, identifying type III secretion system effectors (T3SEs), characterizing their specific activity and thereby understanding their functions on human cells, are all key steps towards providing 'immune boost' treatments for critical infections.

Previous studies across various bacterial species had suggested that the key signals characterizing T3SEs exist in the first 30/100 amino acids at the N-terminus consisting of intrinsically disordered sequence features (Buchko *et al.*, 2010), as well as a chaperone-binding domain where structural motifs come together as three-dimensional signals to provide for effector recognition by the T3SS (Birtalan *et al.*, 2002; Ernst *et al.*, 2018; Lilic *et al.*, 2006). Based on this assumption, past computational methods have been developed to identify T3SEs through extracting features as inputs of machine learning models based on N-terminal protein sequences (Arnold *et al.*, 2009; Karavolos *et al.*, 2005; Lloyd *et al.*, 2001, 2002; Samudrala *et al.*, 2009). These methods successfully stimulated the discovery of new T3SEs. However, recent works have revealed that signals for T3SE recognition and transport are not confined to the N-terminus, but can exist dispersed through the protein sequence and do not necessarily require the action of chaperones for their recognition (Goldberg *et al.*, 2016). In addition, the existing methods largely employ few and simple features, when a wide range of features extracted from different aspects can better characterize a protein type, and provide superior protein classification performance (Wang *et al.*, 2017a, 2018; Zou *et al.*, 2013). Specifically, there is potential in a case such as that of T3SE detection where a great breadth of species has been sampled by wet-lab studies, computational models trained with evolutionary information based features can contribute greatly to the final prediction and can act as an essential factor when constructing predictors.

Some current methods for TS3E detection combine various features to train a single model based on classic machine-learning algorithms, e.g. support vector machine (SVM) (Dong *et al.*, 2013, 2015; Goldberg *et al.*, 2016; Samudrala *et al.*, 2009; Wang *et al.*, 2011, 2013b; Yang *et al.*, 2010), Naive Bayes (NB) (Arnold *et al.*, 2009; Tay *et al.*, 2010), random forest (RF) (Yang *et al.*, 2013), artificial neural network (ANN) (Löwer and Schneider, 2009) and Markov Model (MM) (Wang *et al.*, 2013a). Among these, Löwer and Schneider (2009) investigated the performances of both ANN and SVM, while Dong *et al.* (2015) and Goldberg *et al.* (2016) proposed a hybrid model by combining a BLAST-based predictor and an SVM-based classifier. Nevertheless, novel and promising machine learning algorithms (Ke *et al.*, 2017; Meng *et al.*, 2016; Wen *et al.*, 2016) with better performance are emerging, which might lead to improved integrated models based on ensemble learning (Zhou, 2015).

To take up these challenges, and to address the aforementioned shortcomings of existing methods, we present Bastion3 (Bacterial secreted effector classifier for type III secretion system) a two-layer ensemble learning-based predictor for accurately identifying T3SEs from protein sequences. Bastion3 is designed based on four novelties: (i) To gain more informative patterns for more accurate T3SE recognition, it extracts features from full-length sequences instead of exclusively from the N-terminus or C-terminus, the effects of which are further demonstrated by our experiments; (ii) in contrast to existing methods that employ fewer and simplistic features (see a list in Supplementary Table S1), Bastion3 explores a comprehensive set of features by considering multiple aspects that collectively characterize T3SEs. These include multiple evolutionary information-based features that have demonstrated power in previous protein attribution prediction studies (Wang *et al.*, 2017a, 2018; Zou *et al.*, 2013) and which are introduced, analyzed and integrated into our predictor. As expected, these features significantly improve the T3SE prediction accuracy and contribute meaningfully to the final ensemble model of Bastion3; (iii) Bastion3 employs a recent and powerful gradient boosting decision machine (GBM), LightGBM (Ke *et al.*, 2017), with high accuracy, efficiency and scalability, to train the models. Furthermore, a novel genetic algorithm (GA)-based two-step parameter optimization strategy is employed to boost performances of LightGBM-based models, and, lastly, (iv) a two-layer ensemble model is constructed to make full use of different informative groups of features. Benefiting from all these aspects, Bastion3 outperformed existing state-of-the-art predictors for T3SE prediction by more than 5.6% in ACC value, 5.7% in *F*-value, 12.4% in MCC value and 5.8% in AUC value, respectively. The proposed computational framework of Bastion3 is readily applicable and extensible to other different types of protein attributes and function prediction problems. To maximize the convenience of interested users, we further developed a user-friendly, well-designed and easy-to-use online toolkit, which is publicly accessible at http://bastion3.erc.monash.edu/. It is anticipated that this state-of-the-art

toolkit will allow effective and accurate screening for putative T3SEs, thereby expediting the discovery of novel T3SEs and facilitating experimental validation in the future.

2 Materials and methods

The overall workflow of Bastion3 is summarized according to the 5-step rule (Chou, 2011; Song *et al.*, 2018a) in Figure 1A: (i) collection and curation of both training and independent test datasets; (ii) extraction of useful features that describe the key patterns and characteristics of biological sequences; (iii) feature analysis, model parameterization and model ensemble construction; (iv) performance assessment and (v) web server development and deployment.

2.1 Data collection and curation

We constructed the training dataset by mining currently known effectors from the literature, as well as cross-referencing to several existing T3SE datasets (An *et al.*, 2017; Arnold *et al.*, 2009; Dong *et al.*, 2013, 2015; Samudrala *et al.*, 2009; Tay *et al.*, 2010; Wang *et al.*, 2011, 2013a; Yang *et al.*, 2013), and non-effectors from previous works (Wang *et al.*, 2017a, 2018). After manually removing wrongly annotated effectors and homologous sequences at the threshold of 70%, clustered by the CD-HIT program (Huang *et al.*, 2010), the final dataset contained 379 effectors (Supplementary Fig. S1) and 1112 non-effectors. It is worth noting that lower thresholds of the sequence identity (i.e. 50% or below) might help reduce bias introduced by sequence homology, and in principle result in more reliable and powerful trained models. However, due to the limited size of the dataset in this study, using CD-HIT with a higher threshold was deemed necessary.

We subsequently generated an independent test dataset by manually extracting T3SEs from recently published literature and non-T3SEs from various bacterial species, in order to rigorously evaluate the predictive capability of our proposed method, and compare it against the existing state-of-the-art T3SE predictors. After removing highly homologous samples (with more than 70% similarity) from our training dataset, we finally constructed the independent test dataset containing 108 T3SEs (Supplementary Fig. S2 and Supplementary Table S2) and 108 non-T3SEs. We further performed a case study, using additional three very recently experimentally validated T3SEs (Supplementary Table S3) and examined in detail the predictive performance of different approaches.

2.2 Feature extraction

Considering that it is less likely to recognize a T3SE based on single clues (such as N-terminal signals), we also extracted patterns and characteristics of T3SEs from their whole protein sequences, physicochemical properties and evolutionary information, so as to comprehensively assess and model how T3SSs recognize their specific effectors for secretion. Due to the complexity of evolutionary information-based methods and their overwhelming dominance in

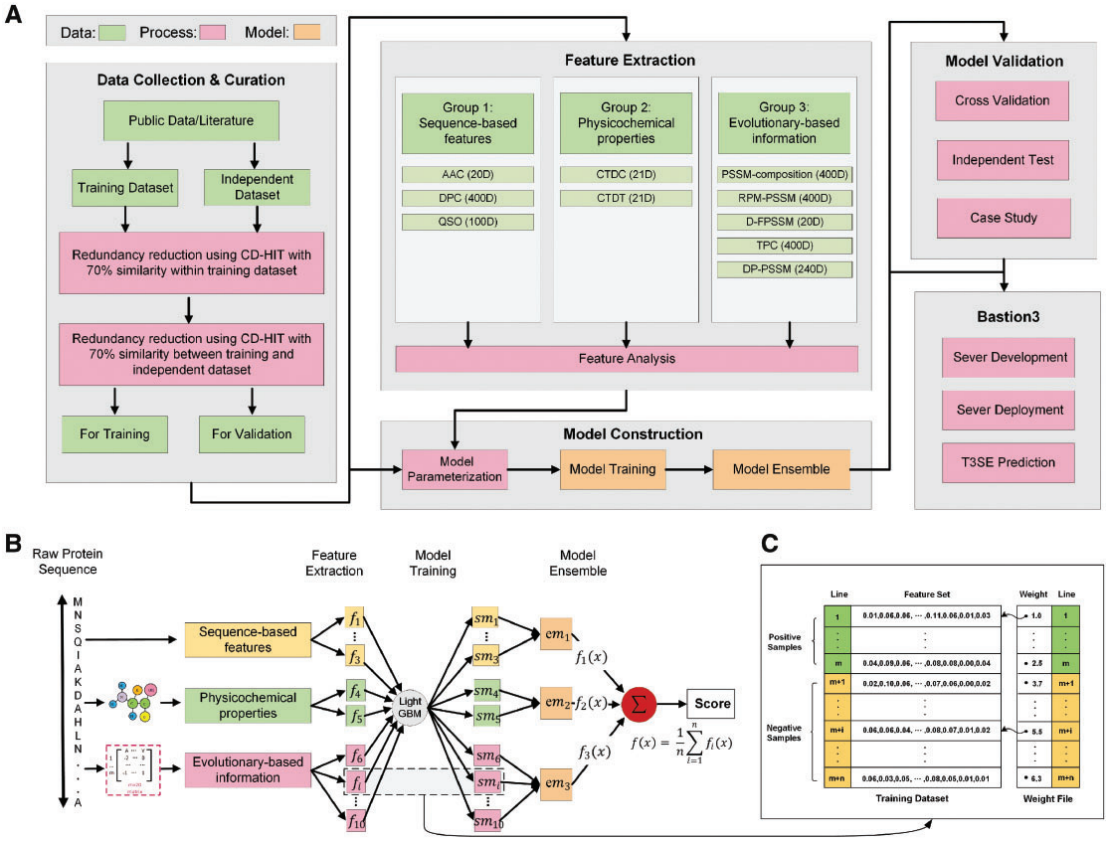


Fig. 1. Overall framework of Bastion3. (**A**) The flowchart of Bastion3 development; (**B**) Detailed procedures for constructing the prediction models within Bastion3's two-layer architecture and (**C**) Tackling the data imbalance problem by assigning a weight to each sample

T3SE prediction, we provide their detailed algorithm descriptions in the following sections.

2.2.1 Group 1: Sequence-based features

The difference between proteins can be directly reflected by their amino acid sequences. We thus extracted three types of sequence-based features: amino acid composition (AAC), dipeptide composition (DPC) and Quasi-Sequence-Order descriptors (QSO). AAC generates a 20-dimensional feature vector by characterizing the occurrence frequencies of 20 amino acids, DPC generates a 400-dimensional feature vector by characterizing the frequencies of dipeptides, while QSO (Chou, 2000) explores a protein's order effect to generate a 100-dimensional feature vector, by measuring the physicochemical distance between amino acids within the sequence. A detailed description of QSO with a set of equations is provided in (Chou, 2000; Wang *et al.*, 2018).

2.2.2 Group 2: Physicochemical properties

Physicochemical properties have been widely and successfully applied in a number of prediction tasks for protein (Wang *et al.*, 2018), DNA (Liu *et al.*, 2018) and RNA (Chen *et al.*, 2016) attributes. In this work, two types of physicochemical property-based features were included to describe the global composition of amino acid properties in protein sequence: CTDC and CTDT (Xiao *et al.*, 2015). Both feature encoding methods categorize the 20 primary amino acids into three main classes, according to seven specific types of physicochemical properties, leading to a 21-dimensional feature vector based on different formulas, respectively. A detailed description of CTDC and CTDT with a set of equations is given in our previous work (Wang *et al.*, 2018).

2.2.3 Group 3: Evolutionary information

Evolutionary information is useful for characterizing common features within specific types of effectors, and thus can be more informative than the sequence information alone when being applied to effector prediction (An *et al.*, 2018; Wang *et al.*, 2017a, 2018; Zou *et al.*, 2013). For a protein sequence with a length of L, a position-specific scoring matrix (PSSM) can be obtained in the form of a $L \times 20$ matrix, representing the sequence's evolutionary information. The (i, j)th element in this matrix represents the probability of amino acid j ($j = 1, 2, \ldots, 20$) to appear at the ith position of the protein sequence (Wang *et al.*, 2017a). Here, we generate the following PSSM-based features using the POSSUM standalone toolkit (Wang *et al.*, 2017b):

(1) The PSSM-composition encoding method (Liu *et al.*, 2010) converts the original PSSM profile into a 20×20 matrix by summing up all rows of the same amino acid residue:

$$R_i = \sum_{k=1}^{L} r_k \times \ \delta_k \tag{1}$$

subject to

$$\begin{cases} \delta_k = 1, \ p_k = a_i \\ \delta_k = 0, \ p_k \neq a_i \end{cases}, \ i = 1, \ldots, \ 20 \tag{2}$$

where R_i represents the ith row of the transformed matrix, r_k denotes the kth row of original PSSM, p_k denotes the kth amino acid in original sequence, and a_i denotes the ith of 20 basic amino acids. As a result, we obtained a 400-dimensional vector by transforming the 20×20 matrix into a straight line.

(2) Inspired by the probe concept used in microarray technologies, the RPM-PSSM encoding (Jeong *et al.*, 2011) applies the residue probing method to scan the original PSSM and filters all the negative elements as a preprocessing step to form a 'filtered' PSSM matrix, where all entities with values of less than 0 are set to 0. This 'filtered' PSSM matrix can be further converted into a 20×20 matrix using the same method as the PSSM-composition encoding method, and we subsequently transformed the 20×20 matrix into a 400-dimensional vector.

(3) Similar to RPM-PSSM, the D-FPSSM encoding (Zahiri *et al.*, 2013) preprocesses the original PSSM profile prior to matrix transformation. D-FPSSM first generates a 'filtered' matrix (termed as FPSSM) by setting all negative elements of the original PSSM profile to 0 and all positive elements greater than an expected value δ (with a default value of 7) to δ. Consequently, all elements in FPSSM vary within the range between 0 and δ, and in this way the negative elements' influence on the positive ones is eliminated when adding two elements during matrix transformation. Based on the FPSSM, we obtained a vector D of 20 dimensions, whose ith element d_i can be calculated as follows:

$$d_j = \sum_{i=1}^{L} fp_{i,j}, \ j = 1, \ldots, \ 20 \tag{3}$$

where $fp_{i,j}$ denotes the element at the ith row and jth column of FPSSM. To eliminate the potential influence of protein sequences with different lengths, d_i is further normalized as follows:

$$d_i = \frac{d_i - min}{max \times L} \tag{4}$$

where min and max denote the minimum and maximum values in the ith column of FPSSM, respectively.

(4) The TPC encoding method (Zhang *et al.*, 2012) calculates the correlation between two adjacent residues by importing the transition probability matrix into PSSM. After transition, we obtained a 400-dimensional vector, which can be defined as follows:

$$\mathrm{TPC} = (y_{1,1}, \ldots, \ y_{1,20}, \ldots, \ y_{i,1}, \ldots, y_{i,20}, \ldots, y_{20,1}, \ldots, y_{20,20})^{\mathrm{T}} \tag{5}$$

$$y_{i,j} = \frac{\sum_{k=1}^{L-1} P_{k,i} \times P_{k+1,j}}{\sum_{j=1}^{20} \sum_{k=1}^{L-1} P_{k+1,j} \times P_{k,i}}, \ 1 \leq i, \ j \leq 20 \tag{6}$$

where $P_{k,i}$ denotes the (k, i)th element in the original PSSM profile.

(5) DP-PSSM (Juan *et al.*, 2009) describes the relationship of an amino acid and the αth (with a default value of 5) succeeding amino acid, which is an extension of Chou's Pse-PSSM algorithm (Chou and Shen, 2007). DP-PSSM first normalizes the elements in the original PSSM profile to a matrix T according to the following three equations:

$$mean_i = \frac{\sum_{k=1}^{20} E_{i,k}}{20} \tag{7}$$

$$STD_i = \sqrt{\frac{\sum_{u=1}^{20} (E_{i,u} - mean_i)^2}{20}} \tag{8}$$

$$T_{i,j} = \frac{E_{i,j} - mean_i}{STD_i} \tag{9}$$

where $E_{i,\ k}$ represents the element at ith row and kth column of original PSSM profile.

Secondly, DP-PSSM calculates the average of squared differences between entries corresponding to amino acids at position i and $i+k$ in the jth column of matrix T:

$$\begin{cases} \overline{\Delta}_{k,j}^{P} = \dfrac{1}{NDP_j}\sum (T_{i,j} - T_{i+k,j})^2, \ \ if \ \ T_{i,j} - T_{i+k,j} \geq 0 \\ \overline{\Delta}_{k,j}^{N} = \dfrac{1}{NDN_j}\sum (T_{i,j} - T_{i+k,j})^2, \ \ if \ \ T_{i,j} - T_{i+k,j} < 0 \end{cases}, \quad 0 < k \leq \alpha, \tag{10}$$

where NDP_j and NDN_j denote the numbers of positive and negative values of $\{T_{i,j} - T_{i+k,j} \mid i = 1, \ 2, \ldots L\}$, respectively.

Thirdly, DP-PSSM combines these positive and negative terms into a feature row vector G_j:

$$G_j = [\overline{\Delta}_{1,j}^{P}, \overline{\Delta}_{1,j}^{N}, \overline{\Delta}_{2,j}^{P}, \overline{\Delta}_{2,j}^{N}, \ldots, \overline{\Delta}_{\alpha,j}^{P}, \overline{\Delta}_{\alpha,j}^{N}], \ j = 1, \ 2, \ldots, \ 20. \tag{11}$$

A feature vector G′ of $20 \times (\alpha \times 2)$ dimensions can be obtained by directly merging 20 vectors G_j:

$$G' = [G_1, G_2, \ldots, G_{20}] \tag{12}$$

Fourthly, DP-PSSM calculates the average of positive and negative terms in each column of the normalized matrix T:

$$\begin{cases} \overline{T}_j^{P} = \dfrac{1}{NP_j}\sum T_{i,j}, \ \ if \ \ T_{i,\ j} \geq 0 \\ \overline{T}_j^{N} = \dfrac{1}{NN_j}\sum T_{i,\ j}, \ \ if \ \ T_{i,j} < 0 \end{cases}, \quad j = 1, \ 2, \ldots, \ 20, \tag{13}$$

and combines these averaged values to a feature row vector T' with 40 dimensions:

$$T' = [\overline{T}_1^{P}, \overline{T}_1^{N}, \overline{T}_2^{P}, \overline{T}_2^{N}, \ldots, \overline{T}_{20}^{P}, \overline{T}_{20}^{N}] \tag{14}$$

Finally, a DP-PSSM feature vector $P_{DP-PSSM}^{\alpha}$ is obtained by combining the generated T′ and G′:

$$P_{DP-PSSM}^{\alpha} = [T', G'] = [p_1, p_2, \ldots, p_{40+40\times\alpha}] \tag{15}$$

In this study, α was set to 5 as the default value and we accordingly obtained a 240-dimensional vector.

2.3 Model training and optimization

2.3.1 LightGBM

Gradient Boosting Decision Tree (GBDT) (Friedman, 2001) is an iterative decision tree algorithm with a variety of successful applications in bioinformatics and computational biology (Chen *et al.*, 2017a; Liao *et al.*, 2016; Rawi *et al.*, 2018). With the explosive growth of feature dimensions and data size, the efficiency and scalability of a few implementations based on GBDT are unsatisfactory (Ke *et al.*, 2017). More recently, a new GBDT extension, LightGBM (Ke *et al.*, 2017) has been proposed, based on two novel techniques: Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) to solve the time-consuming problem of conventional GBDT while retaining high accuracy.

GOSS uses significant data instances, instead of all data instances, for calculations (Supplementary Algorithm S1). Conventional implementation of GBDT requires scanning all data instances for each feature to estimate the information gain for all possible split points. The gradient of the instances, which refer to their absolute values, is positively related to the information gain according to the definition of the latter (Ke *et al.*, 2017). In view of this, GOSS takes all instances with large gradients and randomly samples instances with small gradients, to estimate the information gain for reducing computational complexities. By doing so, it is equivalent to reducing the number of data instances at the time of calculation and further improving the operating efficiency.

In addition to the number of data instances, the number of features is also reduced in LightGBM. High-dimensional feature space is often sparse. The EFB algorithm uses a greedy idea to bundle many exclusive features into a single feature which rarely take non-zero values simultaneously without affecting the accuracy of the calculation (Supplementary Algorithm S2). Therefore, the speed of model training in LightGBM is significantly boosted over other GBDTs because the number of bundled features will be much smaller than those of the original features.

LightGBM improves the efficiency of model training by reducing both numbers of data instances and features. Multiple sets of experiments have shown that the training speed of LightGBM is 20 times higher than conventional GBDT on the premise of maintaining the same accuracy (Ke *et al.*, 2017). In this work, LightGBM was implemented using the *lightgbm* package in R language (https://github.com/Microsoft/LightGBM).

2.3.2 Parameter optimization

Compared with traditional machine learning algorithms, which only need to adjust two or fewer parameters, LightGBM requires users to tune a larger number of parameters (Supplementary Table S4) to ensure model's accuracy and robustness. To optimize the 12 required parameters, the simplest method is to use the grid-search parameter adjustment to exactly obtain optimal parameters. However, it is costly to use grid-searching to tune all the 12 parameters concurrently, especially when the search space is relatively large. To deal with this, we propose a novel GA-based two-step parameter tuning strategy (Algorithm 1) to approximate an optimal solution but in a largely reduced computational time.

Algorithm 1 A GA-based two-step parameter optimization

Step 1: One by one parameter tuning for 12 parameters
Input: *parameters*: 12 parameter intervals, *M*: max AUC
for *parameter* **in** *parameters* **do**
  for $i = 1$ **to** *len(parameter)* **do**
    AUC ← lightgbm.tune(*parameter[i]*)
    if $M \leq AUC$ **then**
      $M \leftarrow AUC$
      lightgbm.parameter ← *parameter[i]*
Output: 12 preliminarily adjusted parameters
Step 2: GA-based parameter tuning
Input: 12 preliminarily adjusted parameters
fitness: lightgbm.tune()
max ← α* preliminary adjusted parameters
min ← β * preliminary adjusted parameters
finally adjusted parameters ← GA(*fitness*, *max*, *min*,...)
Output: 12 finally adjusted, optimal parameters

First, tune parameters (one by one) for each and all 12 parameters. Considering that the amount of grid search calculation is too large to be handled by a normal computer, we adjusted the abovementioned 12 parameters one by one to maximize the value of AUC based on 10-fold cross validation. The tuned optimal parameters are then used as the input parameters for tuning the next parameter, until all 12 parameters have been adjusted. Second, parameter tuning based on GA (Hooker, 1995; Scrucca, 2013) is taken using the

output from step 1 as preliminary parameters. GA is a randomized search method inspired by the principles of evolution in living systems. After all 12 parameters are initially adjusted, one by one, GA is applied to further fine tune them and obtain all 12 final parameter values. In this process, the value of the AUC, based on 10-fold cross validation of the LightGBM model, is defined as the fitness value, and the upper or lower boundary values ($\alpha = 1.8$ and $\beta = 0.2$) of the 12 parameters are set to the by 80 percent increased or decreased values of those obtained in step1, respectively. The GA method was implemented using the GA package (Scrucca, 2013) in the R language.

2.3.3 Solving imbalanced problem

In the field of computational biology, there are often more negative samples than positive samples in collected datasets. The imbalance between the positive and negative samples can lead to overfitting of a model that favors the prediction of the sample class that has the larger proportion: namely, the prediction can be biased towards the category with more samples. This situation applies to our training dataset, where the number of T3SEs is much smaller than that of non-T3SEs. To solve this problem, LightGBM provides a means to assign a particular weight for each sample (Fig. 1C). Considering it is impossible to optimize a weight for every individual sample, we assign a weight for the samples with the label (positive or negative in our work) and then tune weights for both positive and negative samples. In this study, specifically, labels of the positive and negative samples were set to 1 and 0, respectively. The weight of each sample is defined as follows:

$$weight\ (i) = (label(i) \times w\ + 1) / \sum_{j=1}^{N} (label\ (j) \times w + 1) \qquad (16)$$

where *label (i)* denotes the *i*th instance label, *weight (i)* denotes the *i*th instance weight and N is the total number of instances (including positive and negative samples). Since the proportion of the negative and positive samples is about 3 (in the range of 1–10), we set $w \in \{1, 2, 3, \ldots, 9, 10\}$ to calculate ten different positive and negative sample weights. Accordingly, the *weight* parameter was optimized among the above ten different weights in this work.

2.4 Integrative model construction

Compared with the models trained simply with a combined set of features, the ensemble learning strategy can, in principle, significantly improve the model performance (Chen *et al.*, 2017b; Chen and Jeong, 2009; Wan *et al.*, 2017; Wang *et al.*, 2017a, 2018; Zhang *et al.*, 2018; Zou *et al.*, 2015). In this work, a LightGBM-based classifier is trained with each feature encoding method, and for each feature group (i.e. group 1, group 2 or group 3) the prediction scores of their classifiers were evenly averaged to obtain a one-layer ensemble model to represent each feature group's predictive contribution. The prediction scores of these one-layer ensemble models are then integrated as a final two-layer ensemble model (Fig. 1B) using different weights (group 1: group 2: group 3 = 1: 1: 2), considering that PSSM-based features (group 3) possess a dominant position in T3SE prediction (data shown in Section 3.1.2).

2.5 Performance evaluation

To analyze the contribution of different feature groups and to measure the performance of the ensemble models as compared with existing state-of-the-art methods, five metrics using Chou's intuitive representation (Lin *et al.*, 2014; Song *et al.*, 2018b) are applied based on cross-validation and independent test as defined by:

$$SN = 1 - \frac{N_-^+}{N^+} \quad 0 \le SN \le 1 \qquad (17)$$

$$SP = 1 - \frac{N_+^-}{N^-} \quad 0 \le SP \le 1 \qquad (18)$$

$$ACC = 1 - \frac{N_-^+ + N_+^-}{N^+ + N^-} \quad 0 \le ACC \le 1 \qquad (19)$$

$$F - value = 2 \times \frac{N^+ - N_-^+}{2N^+ - N_-^+ + N_+^-} \quad 0 \le F - value \le 1 \qquad (20)$$

$$MCC = \frac{1 - \left(\frac{N_-^+}{N^+} + \frac{N_+^-}{N^-}\right)}{\sqrt{\left(1 + \frac{N_+^- - N^+}{N^+}\right)\left(1 + \frac{N_-^+ - N_+^-}{N^-}\right)}} \quad -1 \le MCC \le 1 \qquad (21)$$

where N^+ and N^- represent the total numbers of positive and negative samples, respectively. N_-^+ denotes the number of positive samples incorrectly predicted to be negatives, while N_+^- denotes the number of the negative samples incorrectly predicted to be positives. Moreover, the Receiver Operating Characteristic (ROC) curves are used to visualize the performance results between different methods with a calculated area under ROC curve (AUC).

3 Experimental results

3.1 Performance evaluation based on 100-time 5-fold cross-validation test

All experiments in this section were conducted using the benchmark training dataset by performing 100-time 5-fold cross-validation test. The prediction models were trained and tuned using the two-step parameter optimization if not explicitly specified.

3.1.1 The effect of parameter optimization

To examine how our proposed two-step parameter optimization improved the trained LightGBM models, we compared the performance of the models tuned by this two-step parameter optimization with those tuned by the first-step-only parameter optimization and those trained with the initial parameter setting. As shown in

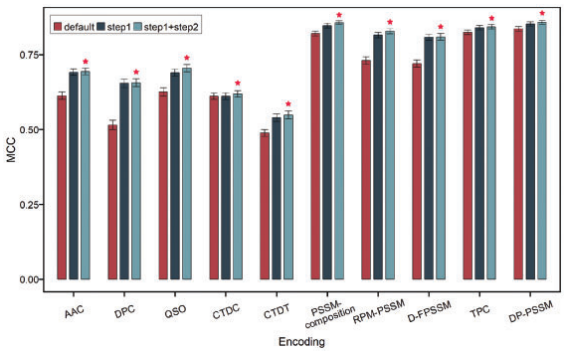


Fig. 2. The effect and performance comparison of two-step parameter optimization of different feature encoding methods, compared with one-step parameter optimization and initial parameter settings. The red star indicates the best performance amongst the three different parameter settings for each feature encoding method

Figure 2 and Supplementary Table S5, the first-step-only parameter optimization could significantly improve the model performance, compared with models trained using the initial parameters. This obvious performance improvement benefited from the efficiency of the step-wise parameter tuning, which makes it possible to preliminarily tune parameters in the prohibitively large search space. Based on the output of the first-step-only parameter optimization, the GA-based parameter tuning strategy could further improve the model performance by fine tuning within a relatively small parameter search range. Taken together, such new two-step parameter optimization strategy enables the LightGBM models trained with different feature encoding methods to achieve the best performance at a reasonable cost of the training time.

3.1.2 Performance evaluation between different feature encoding method

In this section, we further evaluated the performance of models trained using different feature encoding methods. We conducted a 100-time randomized 5-fold cross-validation test for each feature encoding method and compared corresponding predictive performances. More specifically, for each feature set, we first reduced their dimensions to two using the t-SNE algorithm (van der Maaten and Hinton, 2008), such that they could be projected and visualized in 2D. As shown in Figure 3A, the red and grey spots represent T3SE and non-T3SE samples, respectively, while spots with a black edge indicate samples that had been incorrectly predicted. The differences of T3SE and non-T3SE samples in the higher-dimensional space can be represented by their mutual distances in the 2D space. While there is an apparent inhomogeneity in the distribution of T3SE and non-T3SE samples in the 2D projections for all feature encodings, classifications associated with PSSM-based features appear more clustered than those for other features. Moreover, we observe that a decreased number of samples were marked with black edges when predicted using models trained on PSSM-based feature sets, which suggests that PSSM-based feature encoding methods could indeed extract more informative characteristics and patterns for T3SE classification.

We further examined the performance metrics on the cross-validation tests and, as shown in Figure 3B and Table 1, the PSSM-based feature encoding methods (e.g. PSSM-composition, RPM-PSSM, D-FPSSM, TPC and DP-PSSM) achieved the top-level performance with ACC of larger than 0.927, *F*-value of larger than 0.857 and MCC of larger than 0.809, respectively. All of these were respectively larger than those of sequence-based and physicochemical property-based feature encoding methods. Among these PSSM-based feature groups, the model trained using the DP-PSSM features achieved the overall best performance with an ACC of 0.945, *F*-value of 0.894 and MCC of 0.858, followed by the model trained using the PSSM-composition features, which achieved a slightly worse performance but had the highest SN value of 0.93 (Table 1). These results clearly demonstrate the dominating predictive power of PSSM-based features for predicting T3SEs, and are consistent with previous observations that such PSSM-based features are critical for predicting different types of effectors (An *et al.*, 2018; Wang *et al.*, 2017a; Wang and Li, 2013; Zou *et al.*, 2013). Nevertheless, the best-performing type of PSSM-based features might differ from each other depending on the specific type of effectors. Altogether, these results indicate that different types of effectors may share certain common characteristics and evolutionary features, but still have some subtle differences. In addition to the PSSM-based features, models trained using other sequence-based features also achieved a reasonable performance, among which the QSO model performed the best due to its excellent ability to extract amino acid order information from protein sequence (Table 1). In accordance with the results reported in a previous work (Wang *et al.*, 2018), physicochemical property-based features contributed to the T3SE prediction as another important and complementary feature type.

3.1.3 The effect of sequence segments based on N/C-terminus

To further examine whether the N-terminal or C-terminal sequence features can be used as effective features for T3SE prediction, we extracted three types of sequence-based features (i.e. AAC, DPC and QSO) with varying lengths of N-terminal, C-terminal and full protein sequences. Using these extracted features, LightGBM models were trained, tuned and validated based on 100-time 5-fold cross-validation.

As shown in Supplementary Figure S3, we did not observe any remarkable tendency showing that the model performance increased or decreased in association with the changes in N-terminal or C-terminal sequences. Generally, the models trained using N-terminal features achieved a relatively better performance than those trained with C-terminal features, which is within the expectation that the N-terminal sequences provided more useful information for T3SE classification than the C-terminal sequences. However, we found that the models trained using the commonly used 30 or 100 N-terminal residues failed to achieve a better performance compared to the models trained using full-length sequences, suggesting that the first 30 or 100 N-terminal residues alone could only provide partial information for the identification of T3SEs (Supplementary Fig. S3). Full-length sequence-based models consistently achieved the best performance compared to all other N/C-terminal sequence-based models, suggesting that some features for accurate effector prediction are contained at the full-length protein sequence level, instead of residing within the N-terminal or C-terminal regions. This observation will have important implications for future development of next-generation computational methods for identifying bacterial effectors.

3.1.4 Performance evaluation of PSSM-based features using different databases

Just as PSSM profiles differ when generated using different uniref databases, so might the different options of uniref databases have an influence on the performance of the corresponding models trained on such PSSM-based features. To examine this potential influence, we first generated the PSSM profiles by searching different uniref databases (i.e. uniref50, uniref90 and uniref100) with the same parameter settings, and then extracted PSSM-based features based on these PSSM profiles to train the models. By measuring the performance of these models using the same 100-time 5-fold cross-validation procedure, we characterized the potential effect of the generation of PSSM profiles by searching against different uniref databases on the model performance. As a result, the influence of different uniref databases on the performance of the PSSM-based models was marginal (Supplementary Fig. S4). This observation was consistent with a previous study on type VI secreted effector prediction (Wang *et al.*, 2018).

3.2 Performance validation using independent test

All single feature-based models assessed on the independent test were trained based on the benchmark training dataset and tuned by the proposed two-step parameter optimization. These models were then integrated into the one-layer group-based ensemble models and the final two-layer ensemble model (described in Section 2.4). The

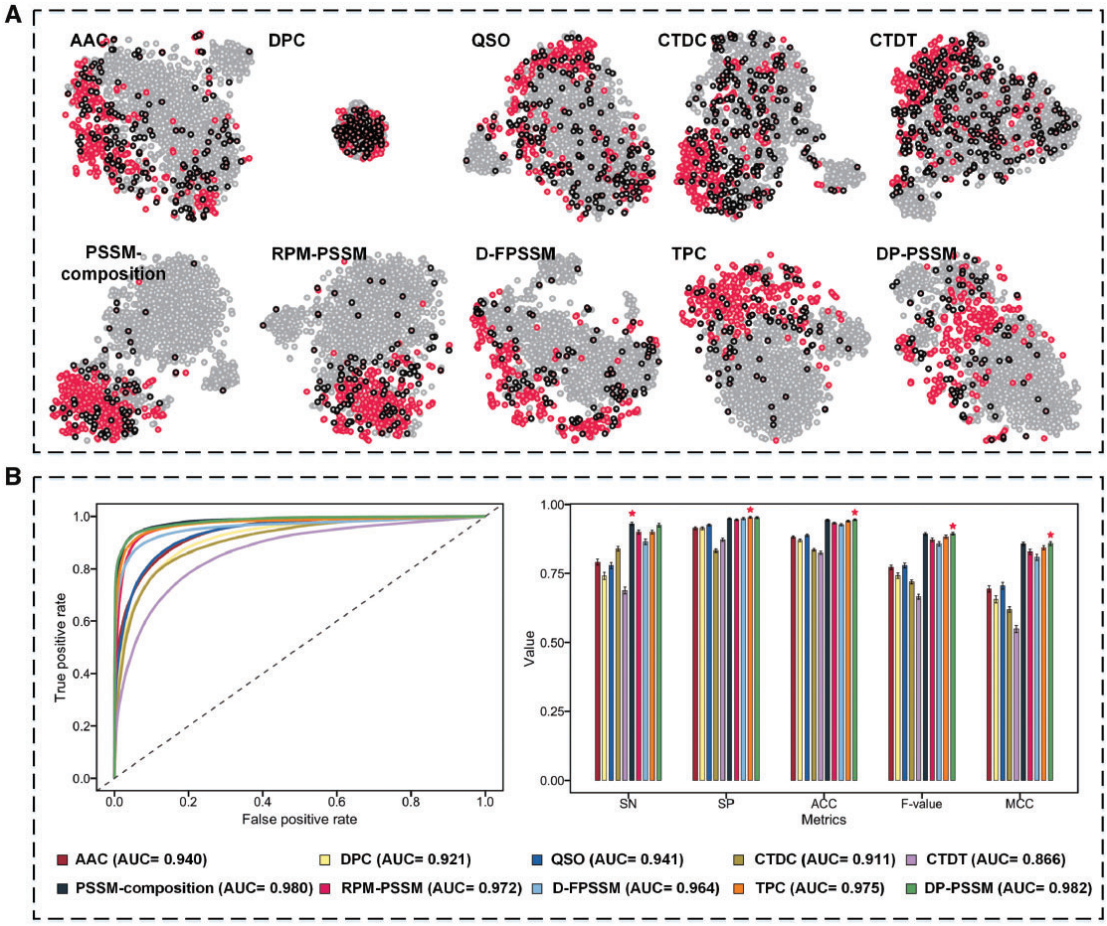


Fig. 3. Performance comparison of different types of feature encoding methods based on 100-time 5-fold cross-validation test. (**A**) Embedding of different types of features using t-SNE (van der Maaten and Hinton, 2008). The red and grey dots represent T3SEs and non-T3SEs, respectively. A black-edge dot indicates that this sample was incorrectly predicted during 100-time 5-fold cross-validation. (**B**) ROC curves and metrics for evaluating the performance of different types of feature encoding methods. The legends of the two panels were merged together with the same feature encoding method denoted by the same color in both panels. The red star on top of the bar chart marks the best performance across different feature encoding methods for each metric

Table 1. Performance comparison of different LightGBM classifiers on the 100-time 5-fold cross-validation test

	Encoding	SN	SP	ACC	*F*-value	MCC
Group 1	AAC	0.791 ± 0.011	0.914 ± 0.005	0.882 ± 0.004	0.772 ± 0.008	0.694 ± 0.011
	DPC	0.741 ± 0.013	0.914 ± 0.005	0.870 ± 0.005	0.742 ± 0.010	0.656 ± 0.013
	QSO	0.779 ± 0.011	0.926 ± 0.004	0.888 ± 0.004	0.779 ± 0.009	0.705 ± 0.012
Group 2	CTDC	0.839 ± 0.009	0.833 ± 0.006	0.835 ± 0.005	0.719 ± 0.007	0.619 ± 0.010
	CTDT	0.688 ± 0.012	0.872 ± 0.006	0.825 ± 0.005	0.666 ± 0.009	0.549 ± 0.013
Group 3	PSSM-composition	**0.930 ± 0.006**	0.949 ± 0.003	0.944 ± 0.003	0.893 ± 0.005	0.857 ± 0.006
	RPM-PSSM	0.900 ± 0.008	0.945 ± 0.003	0.933 ± 0.003	0.872 ± 0.007	0.828 ± 0.009
	D-FPSSM	0.865 ± 0.010	0.949 ± 0.004	0.927 ± 0.004	0.857 ± 0.008	0.809 ± 0.011
	TPC	0.900 ± 0.007	**0.953 ± 0.003**	0.940 ± 0.003	0.883 ± 0.006	0.843 ± 0.008
	DP-PSSM	0.925 ± 0.007	0.952 ± 0.003	**0.945 ± 0.003**	**0.894 ± 0.005**	**0.858 ± 0.007**

Note: Values were expressed as mean ± standard deviation. To facilitate understanding, the best performance value for each metric across different encoding methods is shown in bold font.

independent dataset was randomly and evenly divided into five subsets. Based on these subsets, five-time independent tests were performed and each time one subset was used to assess the predictive performance. The final performance results were obtained by averaging the outcomes of these five random independent tests.

3.2.1 The performance of group-based ensemble models

We first assessed and benchmarked the predictive performance of different single feature-based models on the independent dataset, and present the results in Supplementary Table S6 and Supplementary Figures S5 and S6.

Generally, there is a consistent observation (with results obtained on 5-fold cross-validation test) that PSSM-based, sequence-based and physicochemical property-based models achieved the best, second best and third best performance successively. Additionally, similar to the results on the 5-fold cross-validation test, the model trained with QSO features achieved the best performance among all the sequence-based models, followed by the model trained with AAC features, which highlights the benefit of exploiting the amino acid order information for enhancing the T3SE prediction. However, we noticed that the model trained with TPC features, who were ranked the third on the 5-fold cross-validation test, outperformed all single PSSM-based models with an ACC value of 0.939, *F*-value of 0.939 and MCC value of 0.880. A similar observation also applies to the model trained with D-FPSSM features, whose performance was the worst of all PSSM-based features in 5-fold cross-validation tests, but was ranked the third in the independent test, only slightly inferior to the model trained with PSSM-composition in terms of MCC value. This observation indicates that, while PSSM-based features contributed the most to the T3SE prediction which benefits from the informative evolutionary profile, their performance and relative ranking varied depending on the T3SE sets. This also reflects the necessity and importance to explore different types of PSSM-based encodings and further integrate them into a consolidated framework for improving the prediction of T3SEs.

Next, we examined the performance of one-layer group-based ensemble models and the final two-layer ensemble model, in comparison with single feature-based models. As shown in Supplementary Table S6 and Supplementary Figures S5 and S6, all one-layer group-based ensemble models apparently improved the performance when compared with their corresponding intra-group models as the baseline. After integrating these one-layer ensemble models, the final two-layer model further enhanced its performance to a new level with a remarkable ACC value of 0.959, *F*-value of 0.958, MCC value of 0.917 and AUC value of 0.978.

3.2.2 Comparison with other existing state-of-the-art methods

To further validate the performance of the proposed two-layer ensemble model (termed Bastion3), we compared its performance with that of several existing state-of-the-art methods, including BEAN 2.0 (Dong *et al.*, 2015), pEffect (Goldberg *et al.*, 2016), EffectiveT3 (Arnold *et al.*, 2009), T3_MM (Wang *et al.*, 2013a), BPBAac (Wang *et al.*, 2011) and SIEVE (Samudrala *et al.*, 2009) on the independent test dataset. The performance results of these methods are provided in Figure 4, Supplementary Table S7 and Supplementary Figure S7. Note that the majority of the existing toolkits predict samples with a true or false label without providing a detailed probability score, which may result in slight differences on the performance comparison. Therefore, to make a fair comparison, we first transformed the generated probability scores of Bastion3 and SIEVE into the predictive labels (true or false), and then uniformly generated the ROC curves for all the toolkits based on their predictive labels.

As can be seen, Bastion3 comprehensively outperformed all the compared methods by more than 5.6% in ACC value, 5.7% in *F*-value, 12.4% in MCC value and 5.8% in AUC value. While achieving a better SP value than Bastion3, SIEVE and BPBAac yielded a much lower SN value, which indicates a tendency to generate more false negatives. In addition, we noticed that BEAN 2.0 and pEffect, which integrated a BLAST search procedure (similar to the PSSM profile generation procedure used by Bastion3) in the prediction process, achieved a comparatively better and more stable performance than that of other existing methods. This indicates that BLASTing a query sequence against a specified T3SE dataset (in the case of BEAN 2.0 and pEffect) or the commonly used uniref database (in the case of Bastion3), could provide useful information, which can be further exploited for improved T3SE prediction.

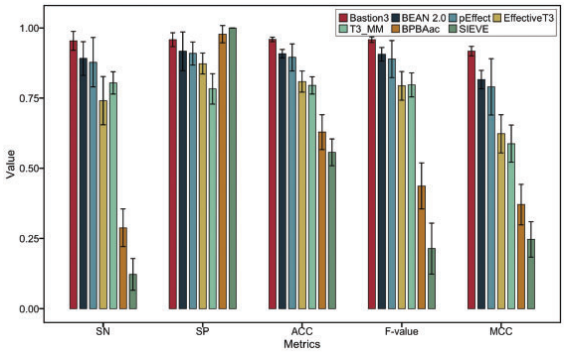


Fig. 4. Performance comparison between Bastion3 (using the final two-layer ensemble model) and six other existing methods for T3SE prediction on the independent test

3.3 Case study

In this section, we performed a case study based on three recent experimentally validated T3SEs to validate the predictive capability of our proposed Bastion3 model, as compared with other existing methods. Phylogenetic analysis was performed using Clustal Omega (Li *et al.*, 2015), with the results visualized using iTOL (Letunic and Bork, 2016) so as to show the relationships between the three study proteins and all T3SEs in the training dataset and independent dataset in terms of the sequence similarity (Supplementary Fig. S8). In addition, the pair-wise sequence alignment between each of the three proteins and its closest sequence homologue was generated using T-Coffee (Notredame *et al.*, 2000) and visualized using Jalview (Clamp *et al.*, 2004) (Supplementary Figs S9–S11). We found that sequences from this study differ significantly in terms of the similarity, compared with those that were included in the training and independent test datasets. Besides, the conserved sequence motifs in the case study proteins were visualized through searching the case study proteins against Pfam (Bateman *et al.*, 2002) by using the MOTIF Search service in GenomeNet (Kanehisa, 1997) (Supplementary Fig. S12). Detailed prediction results generated from different variant models of Bastion3 and other compared methods are provided in Supplementary Tables S8 and S9.

The first case study protein is XopAV (Teper *et al.*, 2016), an experimentally confirmed T3SE protein in *Xanthomonas campestris*. As originally discovered by Teper *et al.*, XopAV did not have sequence homology to any previously known T3SEs obtained in other bacterial species and its biological function remained to be characterized. Only Bastion3 and BEAN 2.0 correctly identified XopAV as a T3SE. Upon a closer look at the prediction results, we found that all sequence-based models (group 1), physicochemical property-based models (group 2) and their one-layer ensemble models failed to predict this protein. However, PSSM-based models (except D-FPSSM) and their ensemble models could precisely recognize XopAV as a T3SE. This again confirmed the effectiveness of incorporating the PSSM-based features to capture the evolutionary

relationship of T3SEs' and improve their prediction, even under the circumstances where there only exists a low sequence similarity.

Next, we proceeded to investigate two more, recently validated T3SE proteins: HaRxL23 from *Hyaloperonospora arabidopsidis* (Deb *et al.*, 2018) and YggG from *Salmonella Typhimurium* (Li *et al.*, 2018). Through convergent evolution, HaRxL23 is structurally and functionally similar to the bacterial effector protein AvrE, but at the sequence level they are unrelated (Deb *et al.*, 2018). Bastion3 readily predicted HaRxL23as a T3SE with a relatively higher score of 0.820, while BEAN 2.0, EffectiveT3 and T3_MM also successfully predicted this T3SE protein. As to YggG, although its biological function is not entirely clear, this effector and its protease activity were demonstrated essential for the virulence of *Salmonella* (Li *et al.*, 2018). Bastion3 successfully recognized YggG as a T3SE protein with a relatively high score of 0.815. As a comparison, amongst all other six existing tools, only EffectiveT3 correctly predicted YggG as a T3SE.

Taken together, Bastion3 outperformed all the currently available T3SE methods and achieved a more accurate and robust predictive performance. Our analyses indicate that different feature-based models contributed to the T3SE prediction, making it necessary to make full use of such heterogeneous features extracted from different perspectives. We also show that even in cases where single feature-based models or their one-layer group-based models fail to make the correct prediction, combining all these features in the final ensemble model could make an important difference and helped correct errors, thus improving the performance of T3SE prediction.

3.4 Web server construction and usage

To maximize user convenience without going through complicated algorithmic details, we have developed a user-friendly and easy-to-use web server as an implementation of the proposed two-layer ensemble model of Bastion3. The web server is deployed and hosted by an extensible and well-maintained cloud computing server machine at Monash University, publicly accessible at http://bastion3.erc.monash.edu/.

Using Bastion3's user submission interface (Supplementary Fig. S13A), users can directly fill the input form or upload a query sequence file in the raw or FASTA format, and submit their job tasks. Once submitted, unique URL links will be returned, enabling the users to check the processing status of their jobs. After jobs are accomplished, users will be notified if they choose to provide an email address along with their job submission. Users could check the status and output of their finished jobs via the prediction results page (Supplementary Fig. S13B), which provides the detailed prediction result for each query protein sequence (including prediction scores of single feature-based models and the final two-layer ensemble model). The result will be marked as 'predicted protein' for a computationally predicted protein, and as 'experimentally validated protein' if the query protein is a known, experimentally validated effector. In the latter case, a corresponding URL link of BastionHub (a public database in development that integrates all the annotations of currently known secreted effectors, coupled with the analysis and prediction functionality) is available to provide detailed information for this effector. Moreover, users can download all the prediction results in multiple formats for local analysis and research purposes.

4 Discussion

In this work, we have presented Bastion3, a two-layer ensemble predictor developed for accurate identification of T3SE proteins. First, we showed that full-length protein sequences contained more useful information and patterns for T3SE prediction than their N-terminus and/or C-terminus alone. Second, we exploited a wide range of complementary and heterogeneous features, and trained and assessed the model performance based on 100 randomization runs of 5-fold cross-validation tests. Specifically, the evolutionary information-based features, which have proved useful in a number of previous prediction studies of protein attributes, significantly improved the predictive performance and contributed the most to the final ensemble model of Bastion3. Third, we leveraged a recently proposed machine learning algorithm, LightGBM, to improve the models for each feature type coming from imbalanced datasets. Moreover, we also proposed a novel GA-based two-step parameter optimization strategy to boost the performance of LightGBM models, with considerably reduced computational time (compared to grid search parameter optimization) during the multiple parameter tuning process. Finally, we integrated single feature-based LightGBM models based on each main feature group into one-layer ensemble models, and further integrated such one-layer ensemble models to construct a final two-layer ensemble model. The extensive benchmarking test and case study validations demonstrated that Bastion3 represents a comprehensive, state-of-the-art predictor, which has clearly outperformed all other existing methods for T3SE prediction.

It is anticipated that our Bastion3 methodology and user-friendly online web server, will expedite the discovery of putative T3SEs and greatly facilitate the effort of a wider research community for functional characterization and understanding of the roles of bacterial effectors. In addition to effector prediction, we believe that the proposed computational framework, including feature analysis, model training and parameter optimization, and ensemble model construction strategy, can serve as useful guidance and inspire researchers to develop novel computational methods in a broader context in the field of bioinformatics and computational biology.

Funding

This work was financially supported by grants from the National Health and Medical Research Council of Australia (NHMRC) (1092262, 1127948 and 1144652), the Australian Research Council (ARC) (LP110200333 and DP120104460), the National Institute of Allergy and Infectious Diseases of the National Institutes of Health (R01 AI111965), the Natural Science Foundation of Guangxi (2016GXNSFCA380005) and the collaborative research program of the Institute for Chemical Research, Kyoto University (2018–28). TML and AL's work was supported in part by the Informatics Institute of the School of Medicine at UAB. TL is an ARC Australian Laureate Fellow (FL130100038).

Conflict of Interest: none declared.

References

An,Y. *et al.* (2018) Comprehensive assessment and performance improvement of effector protein predictors for bacterial secretion systems III, IV and VI. *Brief. Bioinf.*, **19**, 148–161.

An,Y. *et al.* (2017) SecretEPDB: a comprehensive web-based resource for secreted effector proteins of the bacterial types III, IV and VI secretion systems. *Sci. Rep.*, **7**, 41031.

Arnold,R. *et al.* (2009) Sequence-based prediction of type III secreted proteins. *PLoS Pathogens*, **5**, e1000376.

Bateman,A. *et al.* (2002) The Pfam protein families database. *Nucleic Acids Res.*, **30**, 276–280.

Birtalan,S.C. *et al*. (2002) Three-dimensional secretion signals in chaperone-effector complexes of bacterial pathogens. *Mol. Cell,* **9**, 971–980.

Buchko,G.W. *et al*. (2010) A multi-pronged search for a common structural motif in the secretion signal of Salmonella enterica serovar Typhimurium type III effector proteins. *Mol. BioSyst.,* **6**, 2448–2458.

Chen,W. *et al*. (2016) PAI: predicting adenosine to inosine editing sites by using pseudo nucleotide compositions. *Sci. Rep*., **6**, 35123.

Chen,F. *et al*. (2017a) Prediction of luciferase inhibitors by the high-performance MIEC-GBDT approach based on interaction energetic patterns. *Phys. Chem. Chem. Phys.,* **19**, 10163–10176.

Chen,W. *et al*. (2017b) Detecting N(6)-methyladenosine sites from RNA transcriptomes using ensemble support vector machines. *Sci. Rep*., **7**, 40242.

Chen,X.W. and Jeong,J.C. (2009) Sequence-based prediction of protein interaction sites with an integrative method. *Bioinformatics*, **25**, 585–591.

Chou,K.-C. (2011) Some remarks on protein attribute prediction and pseudo amino acid composition. *J. Theor. Biol.,* **273**, 236–247.

Chou,K.C. (2000) Prediction of protein subcellular locations by incorporating quasi-sequence-order effect. *Biochem. Biophys. Res. Commun.,* **278**, 477–483.

Chou,K.C. and Shen,H.B. (2007) MemType-2L: a web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. *Biochem. Biophys. Res. Commun.,* **360**, 339–345.

Clamp,M. *et al*. (2004) The Jalview Java alignment editor. *Bioinformatics,* **20**, 426–427.

Deb,D. *et al*. (2018) Application of alignment-free bioinformatics methods to identify an oomycete protein with structural and functional similarity to the bacterial AvrE effector protein. *PLoS One,* **13**, e0195559.

Deng,W. *et al*. (2017) Assembly, structure, function and regulation of type III secretion systems. *Nat. Rev. Microbiol.,* **15**, 323–337.

Dong,X. *et al*. (2015) BEAN 2.0: an integrated web resource for the identification and functional analysis of type III secreted effectors. *Database*, **2015**, bav064.

Dong,X. *et al*. (2013) Using weakly conserved motifs hidden in secretion signals to identify type-III effectors from bacterial pathogen genomes. *PLoS One,* **8**, e56632.

Ernst,N.H. *et al*. (2018) High-throughput screening of type III secretion determinants reveals a major chaperone-independent pathway. *mBio*, **9**, e01050–18.

Friedman,J.H. (2001) Greedy function approximation: a gradient boosting machine. *Ann. Stat.,* **29**, 1189–1232.

Galan,J.E. *et al*. (2014) Bacterial type III secretion systems: specialized nanomachines for protein delivery into target cells. *Annu. Rev. Microbiol*., **68**, 415–438.

Galan,J.E. and Waksman,G. (2018) Protein-injection machines in bacteria. *Cell,* **172**, 1306–1318.

Goldberg,T. *et al*. (2016) Computational prediction shines light on type III secretion origins. *Sci. Rep*., **6**, 34516.

Hooker,C.A. (1995) Adaptation in natural and artificial systems – Holland, Jh. *Philos. Psychol.,* **8**, 287–299.

Huang,Y. *et al*. (2010) CD-HIT Suite: a web server for clustering and comparing biological sequences. *Bioinformatics,* **26**, 680–682.

Jennings,E. *et al*. (2017) Salmonella SPI-2 type III secretion system effectors: molecular mechanisms and physiological consequences. *Cell Host Microbe,* **22**, 217–231.

Jeong,J.C. *et al*. (2011) On position-specific scoring matrix for protein function prediction. *IEEE/ACM Trans. Comput. Biol. Bioinform. IEEE ACM*, **8**, 308–315.

Juan,E.Y. *et al*. (2009) Predicting protein subcellular localizations for gram-negative bacteria using DP-PSSM and support vector machines. In *Complex, Intelligent and Software Intensive Systems*, 2009. CISIS'09. International Conference on IEEE. pp. 836–841.

Kanehisa,M. (1997) Linking databases and organisms: genomeNet resources in Japan. *Trends Biochem. Sci.,* **22**, 442–444.

Karavolos,M.H. *et al*. (2005) Type III secretion of the Salmonella effector protein SopE is mediated via an N-terminal amino acid signal and not an mRNA sequence. *J. Bacteriol.,* **187**, 1559–1567.

Ke,G. *et al*. (2017) LightGBM: a highly efficient gradient boosting decision tree. *Adv. Neural Inf. Process. Syst.*, 3149–3157.

Letunic,I. and Bork,P. (2016) Interactive tree of life (iTOL) v3: an online tool for the display and annotation of phylogenetic and other trees. *Nucleic Acids Res.,* **44**, W242–W245.

Li,M. *et al*. (2018) YggG is a novel SPI-1 effector essential for *Salmonella* virulence. *bioRxiv*, 300152.

Li,W. *et al*. (2015) The EMBL-EBI bioinformatics web and programmatic tools framework. *Nucleic Acids Res.,* **43**, W580–W584.

Liao,Z. *et al*. (2016) In silico prediction of gamma-aminobutyric acid type-A receptors using novel machine-learning-based SVM and GBDT approaches. *BioMed Res. Int.,* **2016**, 1.

Lilic,M. *et al*. (2006) A common structural motif in the binding of virulence factors to bacterial secretion chaperones. *Mol. Cell,* **21**, 653–664.

Lin,H. *et al*. (2014) iPro54-PseKNC: a sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition. *Nucleic Acids Res.,* **42**, 12961–12972.

Liu,G. *et al*. (2018) DNA physical properties outperform sequence compositional information in classifying nucleosome-enriched and -depleted regions. *Genomics*, doi: 10.1016/j.ygeno.2018.07.013.

Liu,T. *et al*. (2010) Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile. *Biochimie,* **92**, 1330–1334.

Lloyd,S.A. *et al*. (2001) Yersinia YopE is targeted for type III secretion by N-terminal, not mRNA, signals. *Mol. Microbiol.,* **39**, 520–531.

Lloyd,S.A. *et al*. (2002) Molecular characterization of type III secretion signals via analysis of synthetic N-terminal amino acid sequences. *Mol. Microbiol.,* **43**, 51–59.

Löwer,M. and Schneider,G. (2009) Prediction of type III secretion signals in genomes of gram-negative Bacteria. *PloS One,* **4**, e5917.

Meng,Q. *et al*. (2016) A communication-efficient parallel algorithm for decision tree. *Adv. Neural Inf. Process. Syst.*, 1279–1287.

Notredame,C. *et al*. (2000) Coffee: a novel method for fast and accurate multiple sequence alignment. *J. Mol. Biol.,* **302**, 205–217.

Rawi,R. *et al*. (2018) PaRSnIP: sequence-based protein solubility prediction using gradient boosting machine. *Bioinformatics,* **34**, 1092–1098.

Raymond,B. *et al*. (2013) Subversion of trafficking, apoptosis, and innate immunity by type III secretion system effectors. *Trends Microbiol.,* **21**, 430–441.

Samudrala,R. *et al*. (2009) Accurate prediction of secreted substrates and identification of a conserved putative secretion signal for type III secretion systems. *PLoS Pathogens,* **5**, e1000375.

Scrucca,L. (2013) GA: a Package for Genetic Algorithms in R. *J. Stat. Softw.*, **53**, 1–37.

Song,J. *et al*. (2018a) PREvaIL, an integrative approach for inferring catalytic residues using sequence, structural, and network features in a machine-learning framework. *J. Theor. Biol.,* **443**, 125–137.

Song,J. *et al*. (2018b) iProt-Sub: a comprehensive package for accurately mapping and predicting protease-specific substrates and cleavage sites. *Brief. Bioinf*, doi: 10.1093/bib/bby028.

Tay,D.M. *et al*. (2010) T3SEdb: data warehousing of virulence effectors secreted by the bacterial Type III Secretion System. *BMC Bioinformatics,* **11**, S4.

Teper,D. *et al*. (2016) Identification of novel *Xanthomonas euvesicatoria* type III effector proteins by a machine-learning approach. *Mol. Plant Pathol.,* **17**, 398–411.

van der Maaten,L. and Hinton,G. (2008) Visualizing data using t-SNE. *J. Mach. Learn. Res.*, **9**, 2579–2605.

Wan,S. *et al*. (2017) HPSLPred: an ensemble multi-label classifier for human protein subcellular location prediction with imbalanced source. *Proteomics*, **17**, 17–18.

Wang,J. *et al*. (2017a) Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches. *Brief. Bioinf*, doi: 10.1093/bib/bbx164.

Wang,J. *et al*. (2017b) POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. *Bioinformatics,* **33**, 2756–2758.

Wang,J. *et al*. (2018) Bastion6: a bioinformatics approach for accurate prediction of type VI secreted effectors. *Bioinformatics,* **34**, 2546–2555.

Wang,X. and Li,G.Z. (2013) Multilabel learning via random label selection for protein subcellular multilocations prediction. *IEEE/ACM Trans. Comput. Biol. Bioinf. IEEE ACM,* **10**, 436–446.

Wang,Y. *et al.* (2013a) T3_MM: a Markov model effectively classifies bacterial type III secretion signals. *PloS One,* **8**, e58173.

Wang,Y. *et al.* (2013b) Effective identification of bacterial type III secretion signals using joint element features. *PloS One,* **8**, e59754.

Wang,Y. *et al.* (2011) High-accuracy prediction of bacterial type III secreted effectors based on position-specific amino acid composition profiles. *Bioinformatics,* **27**, 777–784.

Wen,Z. *et al.* (2018) Efficient gradient boosted decision tree training on GPUs. In: *Parallel and Distributed Processing Symposium*, 2018. IPDPS'18. International Conference on IEEE, pp. 234–243.

Xiao,N. *et al.* (2015) protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences. *Bioinformatics,* **31**, 1857–1859.

Yang,X. *et al.* (2013) Effective identification of Gram-negative bacterial type III secreted effectors using position-specific residue conservation profiles. *PLoS One,* **8**, e84439.

Yang,Y. *et al.* (2010) Computational prediction of type III secreted proteins from gram-negative bacteria. *BMC Bioinformatics,* **11**, S47.

Zahiri,J. *et al.* (2013) PPIevo: protein-protein interaction prediction from PSSM based evolutionary information. *Genomics,* **102**, 237–242.

Zhang,S. *et al.* (2012) Using principal component analysis and support vector machine to predict protein structural class for low-similarity sequences via PSSM. *J. Biomol. Struct. Dyn.*, **29**, 634–642.

Zhang,Y. *et al.* (2018) Computational analysis and prediction of lysine malonylation sites by exploiting informative features in an integrative machine-learning framework. *Brief. Bioinf*, doi: 10.1093/bib/bby079.

Zhou,Z.H. (2015) Ensemble learning. *Encyclopedia Biometr.*, 411–416.

Zou,L. *et al.* (2013) Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles. *Bioinformatics,* **29**, 3135–3142.

Zou,Q. *et al.* (2015) Improving tRNAscan-SE annotation results via ensemble classifiers. *Mol. Inform.,* **34**, 761–770.

CHAPTER 3: Integrative system for annotation, analysis and prediction of various types of secreted substrates in Gram-negative bacteria

Due to the important role of the bacterial secreted substrates in pathogen–host interaction, bacterial survival, pathogenesis and competition, significant computational and experimental efforts have been made into the identification of various types of substrates. This significantly promotes the discovery of new substrates and substantially leads to a considerable accumulation of known substrates. The numbers of known substrates largely vary between different species or even among different strains within the same organism, and they display different structural and biochemical properties and functions. Here, it is aimed to comprehensively collect and annotate various types of secreted substrates in Gram-negative bacteria, to provide analytical and predictive functional modules, and to interconnect them as an integrative platform to provide a one-stop service for interested users.

This chapter presents the integrative platform for annotation, analysis and prediction of various types of secreted substrates in Gram-negative bacteria, which is developed in two steps. Section 3.1 describes the BastionX prediction system for systematic and accurate prediction of type I, II, III, IV and VI substrates in Gram-negative bacteria. As an integrative toolkit suite, BastionX consists of a user-friendly online server within the distributed framework and a command line based standalone toolkit to systematically predict various types of substrates from genome-scale data in Gram-negative bacteria in a high throughput manner. This guarantees that BastionX can be practically applied in genome-scale substrate annotation and thus builds up a landscape of substrate distributions within bacteria genomes. Section 3.2 describes the BastionHub, a universal platform for integrating and analyzing substrates secreted by Gram-negative bacteria. Through further integrating BastionX as a prediction option, BastionHub additionally annotates various types of substrates and provides a range of functional modules, including substrate analysis and their relationship analysis. By linking various functional modules together as an interactive system, BastionHub formulates and offers a series of pipelines to enable substrate investigation, prediction and relationship comparison or detection.

3.1

BastionX: Systematic and accurate prediction of secreted substrates in Gram-negative bacteria within a distributed framework

The supplementary information for this manuscript is listed in **Appendix 4**.

BastionX: Systematic and accurate prediction of secreted substrates in Gram-negative bacteria within a distributed framework

Jiawei Wang[1], Ruopeng Xie[2,3], Jiahui Li[1,3,4], Wei Dai[3], Tieli Zhou[4], Tatsuya Akutsu[5], Chaille Webb[1], Christopher Stubenrauch[1], Yanju Zhang[3], Jiangning Song[2,6,7,*] and Trevor Lithgow[1,*]

[1]Infection and Immunity Program, Biomedicine Discovery Institute and Department of Microbiology, Monash University, VIC 3800, Australia;
[2]Infection and Immunity Program, Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University, VIC 3800, Australia;
[3]School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, 541004, China;
[4]Department of Clinical Laboratory, The First Affiliated Hospital of Wenzhou Medical University, Wenzhou, Zhejiang Province, China;
[5]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Kyoto 611-0011, Japan;
[6]Monash Centre for Data Science, Faculty of Information Technology, Monash University, VIC 3800, Australia;
[7]ARC Centre of Excellence in Advanced Molecular Imaging, Monash University, VIC 3800, Australia.

* To whom correspondence should be addressed. Trevor Lithgow: Tel: +61-3-9902-9217; Fax: +61-3-9905-3726; Email: trevor.lithgow@monash.edu;
Correspondence may also be addressed to Jiangning Song: Tel: +61-3-9902-9304; Fax: +61-3-9902-9500; Email: jiangning.song@monash.edu.

ABSTRACT

Gram-negative bacteria have evolved an extraordinary array of secretion systems to export substrates into target cells or the surrounding environment. These substrates differ significantly in their structures and functions, and in the secretory pathways they use. Accordingly, it is particularly difficult to develop computational systems for accurate prediction of substrate type. Currently, several platforms are available to predict the secretory pathway of a given substrate, but they are severely limited in their practicality because they are restricted by their substrate range and/or are not amenable to large-scale sequence input. Considering these limitations, a universal platform has remained elusive until now. In this work, we present an integrative prediction system, BastionX, to comprehensively and accurately predict each type of secreted substrates in Gram-negative bacteria in high throughput. Remarkably, BastionX outperforms existing substrate predictors by three major upgrades: 1) BastionX incorporates the first predictor for type II secreted substrates, includes more accurate predictors for types I, III, IV, and VI, and achieves state-of-the-art performance for each single substrate predictor through an effective stacking strategy to intelligently combine multiple machine learning algorithms with a wide array of feature encoding methods; 2) In the output window, BastionX lists the most likely secretory pathway (if any) used by a given protein and includes additional prediction scores for each of the other pathways; 3) BastionX can be exploited in high throughput using an efficient and extensible distributed framework, thereby outperforming the existing single server-based predictors by up to 5.8 times. In combination with the provided additional standalone toolkit, BastionX can be conveniently executed locally to conduct sequence analysis and be readily integrated into user's own pipeline to facilitate the downstream follow-up analysis. Taken together, BastionX can simultaneously annotate thousands of protein sequences with their potential substrate types, and therefore map a global landscape of how secreted substrates are distributed in bacterial genomes. The distributed web server and standalone toolkit of BastionX are publicly accessible at http://bastionx.erc.monash.edu/.

1. INTRODUCTION

Gram-negative bacteria have evolved a remarkable variety of secretion systems, as their virulence "weapons", to export substrates into the surrounding environment or target cells (1). There are six distinct types of substrates (types I-VI) according to their secretion systems, which vary from type I to type VI secretion systems (abbreviated as T1SS to T6SS, respectively) (2,3). Compared to substrates secreted by T1SS (4), T2SS (5), T3SS (6), T4SS (7) and T6SS (8), substrates secreted by type V secretion systems (9,10) are instead surface-localized proteins that typically remain associated with the bacterial outer membrane. In this context, our study covers all five types of secreted substrates, and therefore excludes the type V secretion system.

Secreted substrates play a vital role in disease and survival; they are important for ensuring growth in particularly harsh environments (including host cells) and competitively killing other bacteria as both vie for the same nutrients. Type I and II substrates are usually hydrolytic enzymes that facilitate access to, and absorption of, nutrients from the environment. The majority of type III, IV and VI substrates however, directly imitate host-cell functions and are hence referred to as secreted "effectors". Recently, two experimentally validated type VI secreted substrates, Mn2+-binding protein TseM (11) and iron uptake assisted protein TseF (12), were discovered, highlighting a new role for type VI substrates in nutrient acquisition.

Using the type I, III, IV, or VI secretion system, substrates are translocated across the bacterial envelope by a one-step secretion mechanism (cytoplasm to extracellularly), while the secretion of type II substrates are conducted in a two-step process, first via the Sec or Tat translocons (cytoplasm to periplasm) then through the type II secretion system to be delivered extracellularly. Accordingly, experimental validation protocols of those substrates vary from one type to another, and even largely differ within the same secretion type, making experimental validation particularly difficult and time-consuming. Therefore, despite the difficulty, it is essential that there exists a computational toolkit that is capable of accurately and systematically predicting each type of secreted substrate in high throughput.

A considerable number of computational approaches have been developed to predict different types of substrates (13-15). These include machine learning-based methods developed to predict type I (16), III (17-30), IV (31-36) and VI substrates (37,38). Some of these toolkits have then been included into other toolkits or integrative toolkits (17,39,40), their capability and practicability could be further expanded by comprehensively predicting various types of substrates within a uniformed toolbox. Towards this target, (41) have developed a toolkit to identify substrates across various secretion systems, but mixed all substrates together as a predictive target. As a result, they

could identify a general substrate without an annotation of their exact type, which largely reduces its practical usability.

Technically, most methods make use of a single machine learning algorithm as a base to train predictive models, including support vector machine (SVM) (18,20-23,31,37,41,42), Naive Bayes (NB) (25), random forest (RF) (16,26), Markov Model (MM) (30), Convolutional Neural Network (CNN) (28,36), and Light Gradient Boosting Machine (LightGBM) (29). To further improve the accuracy of each predictor, some groups opted to use a combination of multiple machine learning algorithms. (27) adopted both ANN and SVM, while (17,19) combined a BLAST-based predictor and SVM-based classifier to predict type III substrates. (32,33,35) trained and integrated multiple machine learning algorithms together for more accurate identification of type IV substrates, while (38) employed multiple machine learning algorithms for predicting type VI substrates.

As the number of feature-encoding methods and machine learning algorithms increases, many combinatory strategies have been applied to obtain more powerful and stable models. For example, regarding to type IV substrate prediction, (31,33,35,36) integrated various types of features and machine learning algorithms as the final ensemble model via the majority vote strategy. Similarly, (38) constructed the final model based on the majority vote strategy to predict type VI secreted substrates. (37) trained a SVM-based model with each of nine features, and integrated them by averaging their output scores within a group-based two-layer framework in type VI substrate prediction. (29) further integrated a set of LightGBM models to predict type III substrates within the same framework but assigned unequal weights for different groups depending on their predictive contributions. (32) characterized type IV substrates by a single feature, based on which eight preliminary models were trained and then integrated using a stacking strategy. This method can be further expanded as a solution to combine the increasing number of feature encoding methods with machine learning algorithms.

In this work, we propose BastionX [Bacterial secreted substrate classifier for type X (X=I, II, III, IV and VI) secretion systems] as an integrative system to systematically predict each type of substrates secreted by Gram-negative bacteria. Aimed at bridging the gap between computational method and practical application, BastionX differentiates itself from previous toolkits by three noticeable advancements: Firstly, in addition to being the first type II substrate predictor, BastionX outperforms existing predictor methods for types I, III, IV, and VI substrate types by employing a stacking strategy to intelligently integrate a considerable number of features and machine learning algorithms. Secondly, BastionX seamlessly integrates each of these single substrate type predictors into a unified platform that lists the likelihood a given substrate is secreted through one of the five pathways and selects the most likely pathway for user convenience (**Fig. 1A**). Lastly, in order to

enable high throughput and genome-scale prediction, BastionX is designed within an extensible distributed framework. This enables BastionX to divide a prediction task into small slices of sub-tasks and then executes them in parallel using multiple computing nodes. Accordingly, BastionX improves its prediction throughput by up to 5.8 times as compared to the single-server-based predictors. In addition, a standalone toolkit is provided to maximize user convenience in executing local substrate prediction or further pipeline integration for downstream analysis. Both the distributed predictor and its standalone toolkit are publicly accessible via http://bastionx.erc.monash.edu/. In summary, BastionX offers systematic, accurate and high throughput prediction of various substrate types in Gram-negative bacteria. It can be used for genome-scale substrate annotation and will no doubt be an important tool for determining how secreted substrates are distributed amongst Gram-negative bacteria. Besides, the pilot distributed architecture outlined is extremely extensible in subsequent high-throughput program development, and is expected to inspire and motivate next-generation toolkit development in the era of big data.

2. MATERIAL AND METHODS

The general framework of BastionX consists of three parts (**Fig. 1**): (1) construction of a single powerful predictor for each substrate type using multiple techniques and strategies (**Fig. 1B**), (2) construction of a unified predictive system by integrating a series of single substrate type predictors to provide all-in-one service to identify various types of substrates, and (3) design and implementation of a practical architecture to achieve high-throughput prediction towards bacterial genome-scale annotation.

2.1 Single substrate predictor construction

2.1.1 Dataset collection and curation

For each type of secreted substrate, we conducted an exhaustive and thorough literature search to construct the benchmark dataset, followed by a redundancy reduction procedure using the CD-HIT program (43) at a sequence identity cut-off threshold of 0.7. Accordingly, we obtained 161, 79, 504, 414 and 148 protein sequences for type I, II, III, IV and VI substrates, respectively. The curated data for each type of substrate was randomly split into the training dataset (80%) and independent dataset (20%) as positive samples (**Table S1**). For each substrate type, 1112 non-substrates used in previous work (29,31,35,37) were included as negative samples in the training dataset. Non-substrates retrieved from the UniProt database were included as negative samples to construct the independent dataset with a ratio of 1:1 between positive and negative samples (**Table S1**).

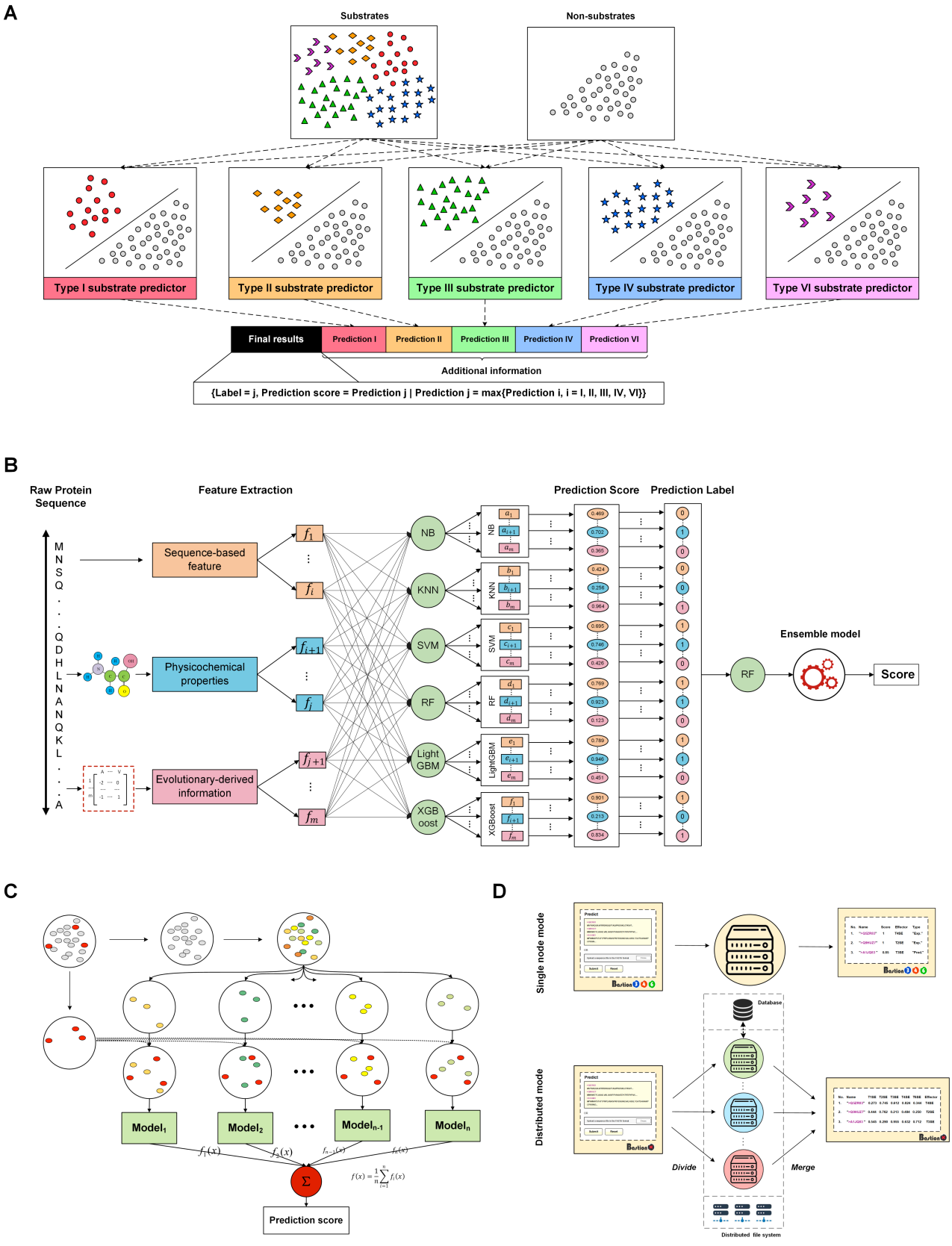


Fig. 1. Overall scheme of BastionX from integrative system design to ensemble model construction to toolkit implementation. (A) The integrative framework of BastionX; (B) Detailed procedures for constructing the single substrate type predictor within BastionX based on a multiple-

stage architecture; (C) Solving the data imbalance problem using multiple-time undersampling; (D) Illustration of the distributed framework of BastionX compared to traditional single-server-based modes.

We further selected five different substrates as a case study to illustrate the superior predictive performance of BastionX, as compared with existing single type predictors. These substrates are associated with different secretion systems, but each of them contains characteristics that can be mistakenly recognized by more than one single type predictor.

2.1.2 Feature representation

As a key step to constructing powerful machine learning-based predictors, feature encoding methods aim to extract distinguishable features from sequences in the form of informative vectors. Considering that a single feature often characterizes samples from one point of view, previous work suggests extracting multiple features to represent different aspects of a given protein could significantly enhance the predictive performance (35,44-46).

Accordingly, in this work, we consider extracting various patterns and features from known substrates by investigating a broad range of heterogeneous feature encoding methods. These features have been successfully applied to tackle different biomedical problems, and could be generally categorized into three major groups, including sequence-based features, physicochemical-based features and evolutionary-based features. Among these, sequence-based features, including AAC, DPC, DP and DDE, explore patterns directly reflected by the amino acids in the protein sequence. Physicochemical-based features, including QSOrder, CTDC and PDT, further characterize the protein sequence by additionally taking into account the physicochemical properties of each of its amino acids. Evolutionary-based features, including RPSSM, TPC-PSSM, RPM-PSSM, DP-PSSM, Pse-PSSM, AAC-PSSM, AB-PSSM, EEDP and PSSM-composition, describe characteristics of sequences based on their associated evolutionary profile, often in the form of position-specific scoring matrix (PSSM). The detailed information of these adopted features, including their brief descriptions, parameter settings and dimensions is listed in **Table S2**.

2.1.3 Machine learning algorithms

Machine learning algorithms mine and infer patterns from data based on statistical learning for generalized predictive power. For this purpose, numerous machine learning algorithms have been designed and implemented from different points of view. As there is no one-size-fits-all algorithm

that fits all datasets, it is necessary to take a trial-and-error approach with each machine learning algorithm for each of our datasets, which vary greatly in terms of the features collected.

To effectively learn patterns from various substrate datasets, we included six different machine learning algorithms: K-nearest neighbor (KNN) (47), Naïve Bayes (NB) (48), support vector machine (SVM) (49), random forest (RF) (50), eXtreme Gradient Boosting (XGBoost) (51), and Light Gradient Boosting Machine (LightGBM) (52). The detailed information of these machine learning algorithms, including brief descriptions, implementation, and parameter-tuning strategies, is listed in **Table S3**.

2.1.4 Solving the data imbalance problem

In our datasets, the number of known substrates is much smaller than that of non-substrates. This is a common situation in practice as experimentally-validated samples are often more difficult to obtain. This imbalance possibly leads to biased models that prefer to predict samples to the class in larger proportion. To solve this data imbalance problem, we applied a simple ensemble strategy in line with our previous work (37) (**Fig. 1C**). For each feature set, we randomly selected N (N=10 in this work) non-substrates using under-sampling, and combined each of them with known substrates, to obtain N balanced datasets. For each of these balanced datasets, we trained a model and then averaged their prediction outputs as an ensemble model (termed a single-method-based model).

2.1.5 Ensemble model construction

Using ensemble learning is a powerful technique for building more accurate and robust predictors (13,53-55). Previously, ensemble models for secreted substrate prediction have been constructed using the majority vote strategy (31,35), the weighted averaging strategy (29,37), or the stacking strategy (32), in order to better integrate multiple features (29,31,37), or machine learning algorithms (32,33), or both (35).

Considering that there are a number of ways to combine a list of ten features with six machine learning algorithms, it is ineffective to simply integrate these generated models together, or set different weights for different models based on an *a priori* assumption. Therefore, for each type of substrate, we constructed an integrative predictor using the stacking strategy, which could be divided into two stages **(Fig. 1B)**. At stage 1, for each feature, we trained 10 single-method-based models based on six individual machine learning algorithms. In total, 60 (10 × 6) single-method-based models were obtained. At stage 2, the numeric outputs of the single-method-based models were first transformed into binary labels (with the value of 0 or 1) and then fed into an RF-based model as the inputs to generate the final ensemble model. In this way, the integrative predictor

could make full use of multiple features and machine learning algorithms, and therefore boost its final prediction performance.

2.1.6 Performance assessment

To comprehensively and rigorously assess the performance of our proposed method, multiple validation methods were used in this study, including 5-fold cross validation, independent test, and case study. Accordingly, five metrics were applied to measure the prediction performance during different validation tests, which are defined as follows:

$$SN = \frac{TP}{TP + FN} \qquad 0 \le SN \le 1$$

$$SP = \frac{TN}{TN + FP} \qquad 0 \le SP \le 1$$

$$ACC = \frac{TP + TN}{TP + FP + TN + FN} \qquad 0 \le ACC \le 1$$

$$F\text{-}value = 2\times \frac{TP}{2TP + FP + FN} \qquad 0 \le F\text{-}value \le 1$$

$$MCC = \frac{(TP\times TN)\text{-}(FN\times FP)}{\sqrt{(TP + FN)\times(TN + FP)\times(TP + FP)\times(TN + FN)}} \qquad -1 \le MCC \le 1$$

where TP, TN, FP, and FN represent the numbers of true positives, true negatives, false positives, and false negatives, respectively. Moreover, the Receiver Operating Characteristic (ROC) curves were used to visualize the prediction performance of the assessed methods with the area under ROC curve (AUC) calculated.

2.2 Construction of the integrative prediction system

To construct the integrative prediction system that could systematically predict each type of substrates, we first trained independent predictors for each type of substrates (**Fig. 1A**). These single type predictors were established using the same method based on their own positive datasets using the same negative dataset. Using the same background data, it is logical that an inquiry protein will be scored highest by its most associated substrate predictor. This could potentially avoid confusion and incorrect predictions in the case where substrates in different types share certain common features and therefore will be positively predicted by two or more single type

substrate predictors (**Fig. S1**). For example, both type III and IV substrates contain featured signals in their sequences, which results in similar residue distribution or evolutionary information between some substrates (56). This situation would unfortunately increase the false positive rate when only single secretion type predictors are used (31). Therefore, to reduce the number of false positives, when given an inquiry protein sequence, BastionX will list the scores generated by each single type substrate predictor, and then specify the most likely substrate type with the highest prediction score.

2.3 Architecture design and implementation

To best meet users' demand, we designed and implemented a distributed web server and an additional standalone toolkit, in favor of users' potential applications at different levels.

2.3.1 The distributed web server

As web servers execute prediction tasks on their own central computing facilities, the prediction throughput is largely confined by their hardware computing capability, which is typically reflected by a limited number of sequences allowed for each individual submission. To improve the prediction throughput, we designed and implemented a distributed web server (**Fig. 1D**) towards bacterial genome-scale prediction. In contrast to previous servers that simply handle a request within the central computing node, BastionX parallelizes this procedure based on a distributed computing cluster, which can be formulated into the four following steps:

Step 1. Task splitting

Upon receiving a prediction request, BastionX splits the requested task into a number of subtasks.

Step 2. Sub-task dispatching

These subtasks are put into a task dispatching system for future task distribution.

Step 3. Subtask execution in parallel

Available threads in computing nodes within the distributing cluster fetch the subtasks and then execute separately on their corresponding nodes.

Step 4. Task merging

Once all subtasks are completed, their respective prediction results will be merged together and appear in the output page.

During the whole process, a database is responsible for sharing the status of each subtask, while a distributed file storage system is employed and configured to share intermediate and resultant files for all subtasks. In this way, the distributed architecture is highly uncoupled and extensible, and therefore readily to be expanded through adding new computing nodes for further increasing demands. It is also noticeable that the whole distributed procedure runs in background at the server

end, and consequently, its current execution and possible future expansion are both transparent to users.

2.3.2 Standalone toolkit

Another way to improve prediction throughput is by providing a standalone toolkit. This encourages users to execute prediction tasks on their local computing facilities, and therefore reduces the burden on the BastionX server. In addition, the standalone toolkit could extend the BastionX with the wider application scope, as it allows further integration into users' own pipelines for automatic and streamlined downstream sequence analysis.

To this end, we have developed a cross-platform standalone toolkit based on Python and R languages. After proper configuration in the local computer by following the user instructions (Python, R, and a number of their libraries are required and specified), users can use a simple command line to execute substrate prediction tasks across various operating systems, including Unix/Linux, Windows and Mac OS.

3. RESULTS

3.1 Performance evaluation based on cross-validation and independent tests

In this section, 5-fold cross-validation and independent tests were conducted to rigorously assess and validate the performance of each single-method-based models and their ensemble models.

3.1.1 Performance evaluation based on cross-validation and independent tests

For each type of secreted substrates, we evaluated the performance of 60 single-method-based models trained using different combinations of features and machine learning algorithms. As shown in **Fig. 2** and **Table S4-S13**, RF and SVM regularly achieved a better performance compared to other machine learning algorithms (i.e. KNN, NB, XGBoost and LightGBM) across the five types of secreted substrate data. This observation is consistent with general machine learning research (57) and our own previous computational biological research (35). On the other hand, two successful implementations of Gradient Boosting Decision Tree (GBDT) (58), XGBoost and LightGBM, achieved the third best performance on the type III substrate dataset, which has the largest number of known substrates, but both GBDT-derived models performed worse than KNN on the other four substrate types. This suggests that both XGBoost and LightGBM would perform better on large-scale datasets (29,59) as they were specifically designed for processing industrial-scale massive data (51,52). NB-based classifiers performed the worst across all types of substrate

datasets, suggesting that they should only act as the complementary elements to construct the final ensemble models.

The performance of models trained with evolutionary-based features was superior to that of the models trained with the two other groups of features, when evaluated on both cross-validation and independent tests across each type of secreted substrates. This demonstrates that the evolutionary-based features play a dominant role in determining which secretion system a substrate will be targeted to (29,35,37). In particular, on the cross-validation test, the SVM-based models trained with DP-PSSM features achieved the over-all best performance with MCC of 0.923, 0.851, 0.801, and 0.814 across type I, III, IV, and VI secreted substrate datasets, respectively, while the SVM-based model trained with Pse-PSSM features achieved the overall best performance with an AAC of 0.907, an F-value of 0.906 and an MCC of 0.818 for predicting type II secreted substrates. Single-method-based models trained with sequence-based features and physicochemical-based features achieved a relatively low performance in the cross-validation tests, but otherwise usually achieved equal or similar prediction performance as models trained with evolutionary-based features based on the independent test. This suggests that sequence-based features and physicochemical-based features could provide additional and complementary information for improving the performance of the final ensemble model (29,35,37,59,60).

3.1.2 Performance evaluation of ensemble models based on the stacking strategy

For each type of secreted substrates, we further compared the performance of 60 single method-based models with that of their ensemble model using the stacking strategy on both 5-fold cross-validation and independent tests.

As shown in **Fig. 2** and **Tables S4-S15**, it is obvious that in most cases the stacking-based ensemble models outperformed their respective single-method-based models. For example, for prediction of type IV secreted substrates, the ensemble model achieved a much better performance compared to its single-method-based models, with an ACC of 0.907, F-value of 0.903 and an MCC of 0.814, respectively on cross-validation test. Although the performance of a few single-method-based models was better than that of the ensemble model based on 10-time 5-fold cross-validation, the performance of these single-method-based models was much worse than that of their ensemble models on the independent test (e.g. the SVM-based model trained with the Pse-PSSM feature for the prediction of type II secreted substrates). These observations strongly agree with previous studies, which show that the performance of single-method-based models can be further improved by an ensemble model using the stacking strategy (61-63).

We also compared the stacking strategy with another two commonly used ensemble strategies, i.e. the averaging strategy (59) and the majority voting strategy (35). The averaging strategy averaged the prediction scores of the single method-based models, while the majority voting strategy conducted a majority vote based on the prediction label of the single method-based models. As a result, the performance of ensemble models using the stacking strategy outperformed their counterparts across all the five types of secreted substrates on 10-time 5-fold cross validation test. All three types of ensemble models achieved a similar performance on the independent test, possibly due to the limited number of independent test samples. Taken together, the ensemble models built using the stacking strategy were more accurate and robust than the ensemble model using either the simple average or majority voting strategy.

3.1.3 Effect of different single-method baseline models on the performance of the final ensemble model

We further explored the contributions of different single method-based models to the performance of the final ensemble model. As the final ensemble model was trained based on RF, the feature importance was measured and ranked according to the values of the Mean Decrease in Gini within the RF model. For each type of secreted substrate, ten RF models were trained during the stacking process based on ten subsets (Refer to the methods section), and the ten resultant groups of values of the Mean Decrease in Gini were then averaged as the final score used to rank the models' importance. For each type of secreted substrate, the importance of the top 30 models was listed and their proportions by each machine learning algorithms are shown in **Fig. 2**. As can be seen from **Tables S16-S20** and **Fig. 2**, the models trained with the evolutionary-based features were consistently first ranked regardless of which machine learning algorithm was used. Specifically, the SVM-based model trained with the DP-PSSM feature had the best performance with an importance value of 31.124, when constructing the final ensemble model for the prediction of type III secreted substrates. In contrast to the contribution of features toward accuracy, we noticed that the SVM- and RF-based machine learning models always accounted for around half of the proportions within the top 30 models across each type of secreted substrate. These results suggest that the performance of single-method-based models positively affects the contribution that they make in the final ensemble model.

3.2 Comparison with other existing state-of-the-art methods on the independent test

To further validate the performance of the BastionX (using the final ensemble model), we compared its performance with that of several existing state-of-the-art predictors across different types of

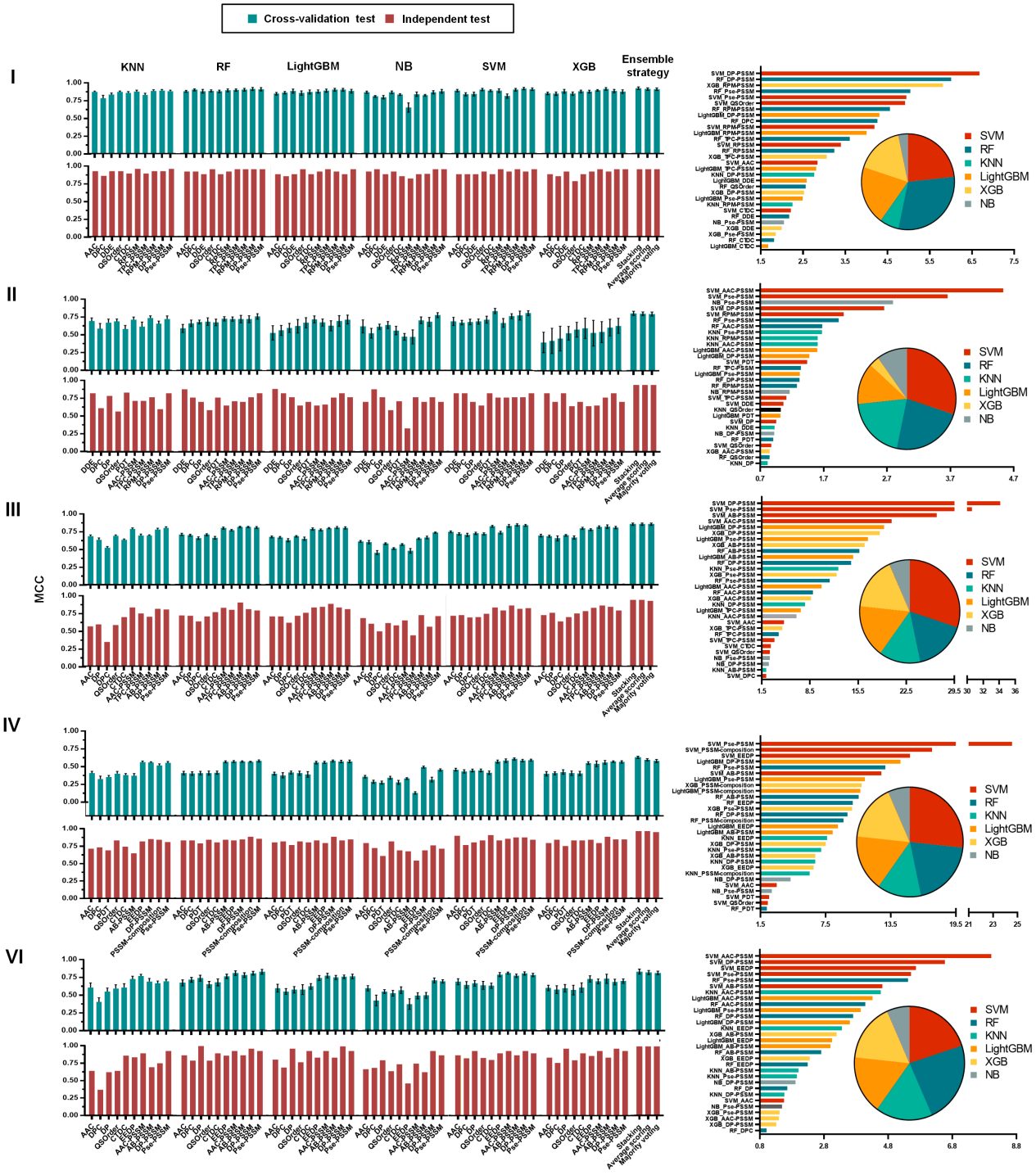
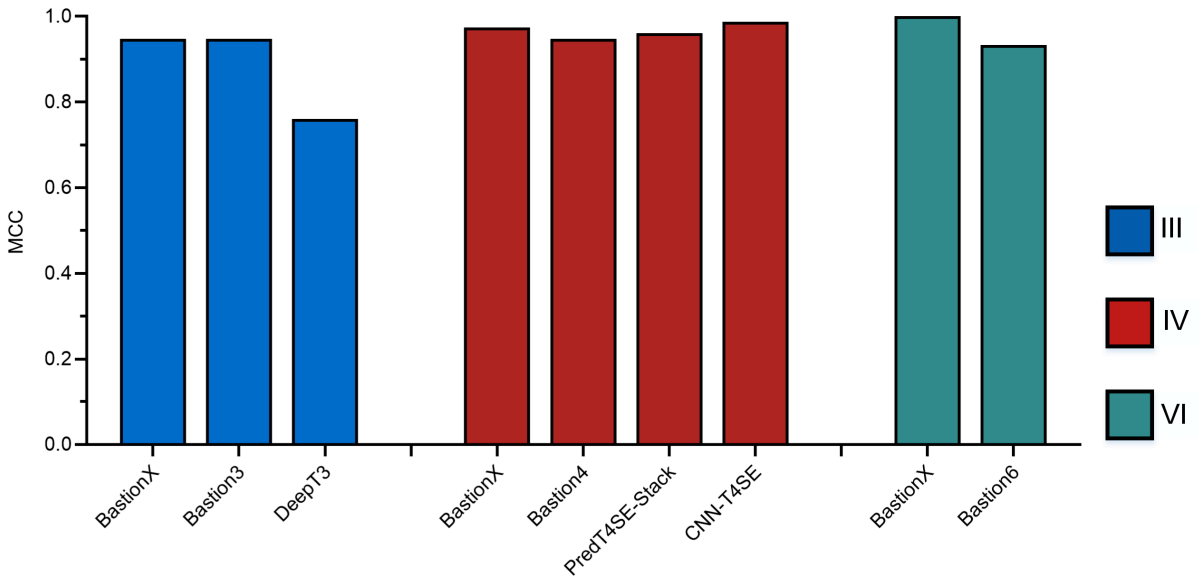


Fig. 2. Performance comparison and analysis of models trained using different methods. (A) The bar charts (left panel) compare the performance between different single method-based models and their ensemble modes in terms of the MCC value on both 10-time 5-fold cross-validation test and independent test. (B) The combination of the bar and pie charts (right panel) shows the contribution of different single method-based models to the performance of the final ensemble model based on the stacking strategy. Panels I, II, III, IV and VI represents the cases for type I, II, III, IV and VI secreted substrates, respectively.

A



B

ID	Type	Predicted results (Correct Incorrect Ambiguous)						
		Bastion X	Bastion 3	DeepT3	Bastion 4	PredT4SE-Stack	CNN-T4SE	Bastion 6
1	I	I or II	III	III	IV	non-IV	non-IV	VI
2	II	II	non-III	non-III	IV	non-IV	IV	VI
3	III	III	III	III	IV	IV	non-IV	VI
4	IV	IV	III	III	IV	IV	IV	VI
5	VI	VI	III	III	non-IV	non-IV	non-IV	VI

Fig. 3. Performance comparison between BastionX and the existing state-of-the-art toolkits. (A) Performance for predicting singe types of secreted substrates on the independent test. (B) Prediction results of the five different substrates in the case study.

secreted substrates by performing the independent test. The currently available toolkits include Bastion3 (29) and DeepT3 (28) for predicting type III secreted substrates, Bastion4 (35), PredT4SE-Stack (32) and CNN-T4SE (36) for predicting type IV secreted substrates, and Bastion6 (37) for predicting type VI secreted substrates. As can be seen from **Fig. 3** and **Table S21**, BastionX achieved an overall best performance than the existing state-of-the-art toolkits when predicting type

III and VI secreted substrates, respectively. It achieved the second-best performance in type IV substrate prediction, slightly inferior to the CNN-T4SE predictor. It is also noticeable that, for the prediction of type VI secreted substrates, BastionX achieved correct predictions on the independent tests. This situation may be due to the limited number of independent datasets, and the performance could be further benchmarked when more data become available in the future. Based on current data and experimental results, BastionX represents the most powerful predictor that provides the accurate and stable performance for predicting different types of secreted substrates.

To further illustrate the superior performance and effectiveness of BastionX as an integrative predictive system, five representative secreted substrates were selected as case studies (**Table. S22**). As a result, the single type predictors incorrectly predicted other types of secreted substrates to be positive as they may have similar characteristics to its own secreted substrates (**Fig. 3**). In contrast, the BastionX predictor accurately specified the secreted substrate pathway by selecting the highest predictive score (**Fig. 3**). More than one single type predictors within the BastionX predictor may predict a sample as their positive samples; however, the decision based on the highest predictive score enables the BastionX to make the final correct prediction (**Table S23**).

3.3 The usage and computational performance of the distributed web server

The BastionX distributed web server is currently equipped with 10 sub-nodes and deployed on Monash University cloud servers, which is freely accessible at http://bastionx.erc.monash.edu/. Users can submit their prediction tasks and have their jobs run in parallel among those sub-nodes. They are able to check their job status and also browse the prediction results via a unique URL link. If the submitted protein is already an experimentally validated secreted substrate, the results will be marked as "experimentally validated", otherwise annotated with a "predicted" label. In the former case, the detailed information about this known secreted substrate will be provided via corresponding URL link of BastionHub (http://bastionhub.erc.monash.edu/). Finally, BastionX provide prediction results downloadable in multiple formats to facilitate users to perform their follow-up analysis locally.

To demonstrate the computational efficiency of this distributed web server, we benchmarked it with a single-node-based server using a genome-scale sequence dataset of Escherichia coli IAI39. Benchmarking experiments were repeated using different subsets, each of which was randomly selected from the overall dataset, starting with 500 sequences, and in increments of 500. As can be seen from Fig. S2, the time taken by BastionX was approximately 1.8-fold to 5.8-fold shorter than that of the single node-based server across different experiments. The possible reasons for the variations of the performance gap are expected because: (1) the randomly selected subsets have a

variable sequence distribution that will affect the parallelization effect of the BastionX, and (2) even though the sub-nodes were equipped with the same hardware and software configuration, their computing abilities are influenced by other nodes within the same cluster environment. The performance gap became larger when the size of datasets increased, highlighting the strength of the distributed framework in the context of large genome-scale data.

4. DISCUSSION

In this work, we have developed BastionX, an integrative bioinformatics suite that is capable of accurately mapping the complete repertoire of substrates secreted by Gram-negative bacteria from sequences information. We employed a stacking-based ensemble strategy to intelligently combine a wide range of feature-encoding methods with multiple machine learning algorithms. Extensive benchmarking experiments demonstrate that this has achieved more robust and accurate prediction performance. Through a distributed architecture-based web server and the additional standalone toolkit, BastionX provides a publicly accessible high-throughput prediction service by up to a 5.8-fold improvement, and can also be further integrated into a user's own pipeline for downstream analysis and meet the user's specific need. With all of these characteristics, BastionX offers practicality with the ability to perform high-throughput screening of thousands of protein sequences and identify their possible substrate types, and will undoubtedly be used as an important first-step during genome-scale annotations.

In the future, newly experimentally-validated substrates with novel characteristics will require additional informative feature encoding methods to mine their intrinsic characteristics. These features accordingly call for interpretation and recognition by new and more attractive machine learning algorithms. The extensible ensemble framework of BastionX makes it particularly suitable to be extended by incorporating additional features together with machine learning algorithms. Specifically, if the substrate datasets expand rapidly to a large enough scale, deep learning techniques may be required to directly learn the underlying patterns and key characteristics from the sequence data without manual feature engineering. In this way, it is promising to take advantages of the strengths and merits of both classical machine learning and deep learning techniques, and achieve more accurate and robust substrate prediction.

AVAILABILITY

http://bastionx.erc.monash.edu/.

FUNDING

This work was financially supported by grants from the National Health and Medical Research Council of Australia (NHMRC) (1092262, 1144652 and 1127948), the Australian Research Council (ARC) (LP110200333 and DP120104460), the National Institute of Allergy and Infectious Diseases of the National Institutes of Health (R01 AI111965), and the Collaborative Research Program of Institute for Chemical Research, the National Natural Science Foundation of China (61862017), and the Natural Science Foundation of Guangxi (2018GXNSFAA138117, 2016GXNSFCA380005). T.L. is an ARC Australian Laureate Fellow (FL130100038).

CONFLICT OF INTEREST

None declared.

REFERENCES

1. Wandersman, C. (2013) Concluding remarks on the special issue dedicated to bacterial secretion systems: function and structural biology. *Research in microbiology*, **164**, 683-687.
2. Costa, T.R., Felisberto-Rodrigues, C., Meir, A., Prevost, M.S., Redzej, A., Trokter, M. and Waksman, G. (2015) Secretion systems in Gram-negative bacteria: structural and mechanistic insights. *Nature reviews. Microbiology*, **13**, 343-359.
3. Green, E.R. and Mecsas, J. (2016) Bacterial Secretion Systems: An Overview. **4**.
4. Welch, R.A., Dellinger, E.P., Minshew, B. and Falkow, S. (1981) Haemolysin contributes to virulence of extra-intestinal E. coli infections. *Nature*, **294**, 665-667.
5. d'Enfert, C., Ryter, A. and Pugsley, A.P. (1987) Cloning and expression in Escherichia coli of the Klebsiella pneumoniae genes for production, surface localization and secretion of the lipoprotein pullulanase. *EMBO J*, **6**, 3531-3538.
6. Galan, J.E. and Curtiss, R., 3rd. (1989) Cloning and molecular characterization of genes whose products allow Salmonella typhimurium to penetrate tissue culture cells. *Proceedings of the National Academy of Sciences of the United States of America*, **86**, 6383-6387.
7. Kuldau, G.A., De Vos, G., Owen, J., McCaffrey, G. and Zambryski, P. (1990) The virB operon of Agrobacterium tumefaciens pTiC58 encodes 11 open reading frames. *Mol Gen Genet*, **221**, 256-266.
8. Pukatzki, S., Ma, A.T., Sturtevant, D., Krastins, B., Sarracino, D., Nelson, W.C., Heidelberg, J.F. and Mekalanos, J.J. (2006) Identification of a conserved bacterial protein secretion system in Vibrio cholerae using the Dictyostelium host model system. *Proceedings of the National Academy of Sciences of the United States of America*, **103**, 1528-1533.
9. Fan, E., Chauhan, N., Udatha, D.B., Leo, J.C. and Linke, D. (2016) Type V Secretion Systems in Bacteria. *Microbiol Spectr*, **4**.
10. Heinz, E., Stubenrauch, C.J., Grinter, R., Croft, N.P., Purcell, A.W., Strugnell, R.A., Dougan, G. and Lithgow, T. (2016) Conserved Features in the Structure, Mechanism, and Biogenesis of the Inverse Autotransporter Protein Family. *Genome Biol Evol*, **8**, 1690-1705.
11. Si, M., Zhao, C., Burkinshaw, B., Zhang, B., Wei, D., Wang, Y., Dong, T.G. and Shen, X. (2017) Manganese scavenging and oxidative stress response mediated by type VI secretion system in Burkholderia thailandensis. *Proceedings of the National Academy of Sciences of the United States of America*, **114**, E2233-E2242.
12. Lin, J., Zhang, W., Cheng, J., Yang, X., Zhu, K., Wang, Y., Wei, G., Qian, P.Y., Luo, Z.Q. and Shen, X. (2017) A Pseudomonas T6SS effector recruits PQS-containing outer membrane vesicles for iron acquisition. *Nature communications*, **8**, 14888.
13. An, Y., Wang, J., Li, C., Leier, A., Marquez-Lago, T., Wilksch, J., Zhang, Y., Webb, G.I., Song, J. and Lithgow, T. (2018) Comprehensive assessment and performance improvement of effector protein predictors for bacterial secretion systems III, IV and VI. *Briefings in bioinformatics*, **19**, 148-161.
14. Zeng, C. and Zou, L. (2017) An account of in silico identification tools of secreted effector proteins in bacteria and future challenges. *Briefings in bioinformatics*.
15. McDermott, J.E., Corrigan, A., Peterson, E., Oehmen, C., Niemann, G., Cambronne, E.D., Sharp, D., Adkins, J.N., Samudrala, R. and Heffron, F. (2011) Computational prediction of type III and IV secreted effectors in gram-negative bacteria. *Infection and immunity*, **79**, 23-32.

16. Luo, J., Li, W., Liu, Z., Guo, Y., Pu, X. and Li, M. (2015) A sequence-based two-level method for the prediction of type I secreted RTX proteins. *Analyst*, **140**, 3048-3056.
17. Dong, X., Lu, X. and Zhang, Z. (2015) BEAN 2.0: an integrated web resource for the identification and functional analysis of type III secreted effectors. *Database : the journal of biological databases and curation*, **2015**, bav064.
18. Dong, X., Zhang, Y.J. and Zhang, Z. (2013) Using weakly conserved motifs hidden in secretion signals to identify type-III effectors from bacterial pathogen genomes. *PloS one*, **8**, e56632.
19. Goldberg, T., Rost, B. and Bromberg, Y. (2016) Computational prediction shines light on type III secretion origins. *Scientific reports*, **6**, 34516.
20. Samudrala, R., Heffron, F. and McDermott, J.E. (2009) Accurate prediction of secreted substrates and identification of a conserved putative secretion signal for type III secretion systems. *PLoS pathogens*, **5**, e1000375.
21. Wang, Y., Sun, M., Bao, H., Zhang, Q. and Guo, D. (2013) Effective identification of bacterial type III secretion signals using joint element features. *PloS one*, **8**, e59754.
22. Wang, Y., Zhang, Q., Sun, M.A. and Guo, D. (2011) High-accuracy prediction of bacterial type III secreted effectors based on position-specific amino acid composition profiles. *Bioinformatics*, **27**, 777-784.
23. Yang, Y., Zhao, J., Morgan, R.L., Ma, W. and Jiang, T. (2010) Computational prediction of type III secreted proteins from gram-negative bacteria. *BMC bioinformatics*, **11 Suppl 1**, S47.
24. Arnold, R., Brandmaier, S., Kleine, F., Tischler, P., Heinz, E., Behrens, S., Niinikoski, A., Mewes, H.W., Horn, M. and Rattei, T. (2009) Sequence-based prediction of type III secreted proteins. *PLoS pathogens*, **5**, e1000376.
25. Tay, D.M., Govindarajan, K.R., Khan, A.M., Ong, T.Y., Samad, H.M., Soh, W.W., Tong, M., Zhang, F. and Tan, T.W. (2010) T3SEdb: data warehousing of virulence effectors secreted by the bacterial Type III Secretion System. *BMC bioinformatics*, **11 Suppl 7**, S4.
26. Yang, X., Guo, Y., Luo, J., Pu, X. and Li, M. (2013) Effective identification of Gram-negative bacterial type III secreted effectors using position-specific residue conservation profiles. *PloS one*, **8**, e84439.
27. Löwer, M. and Schneider, G. (2009) Prediction of Type III Secretion Signals in Genomes of Gram-Negative Bacteria. *PloS one*, **4**, e5917.
28. Xue, L., Tang, B., Chen, W. and Luo, J. (2018) DeepT3: deep convolutional neural networks accurately identify Gram-Negative Bacterial Type III Secreted Effectors using the N-terminal sequence. *Bioinformatics*.
29. Wang, J., Li, J., Yang, B., Xie, R., Marquez-Lago, T.T., Leier, A., Hayashida, M., Akutsu, T., Zhang, Y., Chou, K.C. *et al.* (2019) Bastion3: a two-layer ensemble predictor of type III secreted effectors. *Bioinformatics*, **35**, 2017-2028.
30. Wang, Y., Sun, M., Bao, H. and White, A.P. (2013) T3_MM: a Markov model effectively classifies bacterial type III secretion signals. *PloS one*, **8**, e58173.
31. Zou, L., Nan, C. and Hu, F. (2013) Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles. *Bioinformatics*, **29**, 3135-3142.
32. Xiong, Y., Wang, Q., Yang, J., Zhu, X. and Wei, D.Q. (2018) PredT4SE-Stack: Prediction of Bacterial Type IV Secreted Effectors From Protein Sequences Using a Stacked Ensemble Method. *Front Microbiol*, **9**, 2571.

33. Burstein, D., Zusman, T., Degtyar, E., Viner, R., Segal, G. and Pupko, T. (2009) Genome-scale identification of Legionella pneumophila effectors using a machine learning approach. *PLoS pathogens*, **5**, 6974-6974.
34. Wang, Y., Wei, X., Bao, H. and Liu, S.L. (2014) Prediction of bacterial type IV secreted effectors by C-terminal features. *BMC Genomics*, **15**, 50.
35. Wang, J., Yang, B., An, Y., Marquez-Lago, T., Leier, A., Wilksch, J., Hong, Q., Zhang, Y., Hayashida, M., Akutsu, T. *et al.* (2019) Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches. *Briefings in bioinformatics*, **20**, 931-951.
36. Hong, J., Luo, Y., Mou, M., Fu, J., Zhang, Y., Xue, W., Xie, T., Tao, L., Lou, Y. and Zhu, F. (2019) Convolutional neural network-based annotation of bacterial type IV secretion system effectors with enhanced accuracy and reduced false discovery. *Briefings in bioinformatics*.
37. Wang, J., Yang, B., Leier, A., Marquez-Lago, T.T., Hayashida, M., Rocker, A., Zhang, Y., Akutsu, T., Chou, K.C., Strugnell, R.A. *et al.* (2018) Bastion6: a bioinformatics approach for accurate prediction of type VI secreted effectors. *Bioinformatics*, **34**, 2546-2555.
38. Sen, R., Nayak, L. and De, R.K. (2019) PyPredT6: A python-based prediction tool for identification of Type VI effector proteins. *Journal of bioinformatics and computational biology*, **17**, 1950019.
39. Jehl, M.A., Arnold, R. and Rattei, T. (2011) Effective--a database of predicted secreted bacterial proteins. *Nucleic acids research*, **39**, D591-595.
40. Eichinger, V., Nussbaumer, T., Platzer, A., Jehl, M.-A., Arnold, R. and Rattei, T. (2016) EffectiveDB—updates and novel features for a better annotation of bacterial secreted proteins and Type III, IV, VI secretion systems. *Nucleic acids research*, **44**, D669-D674.
41. Dhroso, A., Eidson, S. and Korkin, D. (2018) Genome-wide prediction of bacterial effector candidates across six secretion system types using a feature-based statistical framework. *Scientific reports*, **8**, 17209.
42. Sato, Y., Takaya, A. and Yamamoto, T. (2011) Meta-analytic approach to the accurate prediction of secreted virulence effectors in gram-negative bacteria. *BMC bioinformatics*, **12**, 1.
43. Huang, Y., Niu, B., Gao, Y., Fu, L. and Li, W. (2010) CD-HIT Suite: a web server for clustering and comparing biological sequences. *Bioinformatics*, **26**, 680-682.
44. Han, S., Liang, Y., Ma, Q., Xu, Y., Zhang, Y., Du, W., Wang, C. and Li, Y. (2018) LncFinder: an integrated platform for long non-coding RNA identification utilizing sequence intrinsic composition, structural information and physicochemical property. *Briefings in bioinformatics*.
45. Khatun, M.S., Hasan, M.M. and Kurata, H. (2019) PreAIP: Computational Prediction of Anti-inflammatory Peptides by Integrating Multiple Complementary Features. *Frontiers in genetics*, **10**, 129.
46. Qiang, X., Zhou, C., Ye, X., Du, P.F., Su, R. and Wei, L. (2018) CPPred-FL: a sequence-based predictor for large-scale identification of cell-penetrating peptides by feature representation learning. *Briefings in bioinformatics*.
47. Altman, N.S. (1992) An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression. *Am Stat*, **46**, 175-185.
48. Friedman, N., Dan, G. and Goldszmidt, M. (1997) Bayesian Network Classifiers. *Machine Learning*, **29**, 131-163.
49. Cortes, C. and Vapnik, V. (1995) Support-vector networks. *Machine learning*, **20**, 273-297.

50. Breiman, L. (2001) Random forests. *Machine learning*, **45**, 5-32.
51. Chen, T. and Guestrin, C. (2016), *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, pp. 785-794.
52. Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.Y. (2017) LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems*, 3149-3157.
53. Lihu, A. and Holban, S. (2016) A review of ensemble methods for de novo motif discovery in ChIP-Seq data. *Briefings in bioinformatics*, **17**, 731.
54. Laczny, C.C., Galata, V., Plum, A., Posch, A.E. and Keller, A. (2017) Assessing the heterogeneity of in silico plasmid predictions based on whole-genome-sequenced clinical isolates. *Briefings in bioinformatics*.
55. Nam, D., Yoon, S.H. and Kim, J.F. (2007) Ensemble learning of genetic networks from time-series expression data. *Bioinformatics*, **23**, 3225-3231.
56. Xu, S., Zhang, C., Miao, Y., Gao, J. and Xu, D. (2010) Effector prediction in host-pathogen interaction based on a Markov model of a ubiquitous EPIYA motif. *BMC Genomics*, **11 Suppl 3**, S1.
57. Fern, Ndez-Delgado, M., Cernadas, E., Barro, S. and Amorim, D. (2014) Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, **15**, 3133-3181.
58. Friedman, J.H. (2001) Greedy function approximation: A gradient boosting machine. *Ann Stat*, **29**, 1189-1232.
59. Zhang, Y., Xie, R., Wang, J., Leier, A., Marquez-Lago, T.T., Akutsu, T., Webb, G.I., Chou, K.C. and Song, J. (2018) Computational analysis and prediction of lysine malonylation sites by exploiting informative features in an integrative machine-learning framework. *Briefings in bioinformatics*.
60. Zhang, Y., Yu, S., Xie, R., Li, J., Leier, A., Marquez-Lago, T.T., Akutsu, T., Smith, A.I., Ge, Z., Wang, J. *et al.* (2019) PeNGaRoo, a combined gradient boosting and ensemble learning framework for predicting non-classical secreted proteins. *Bioinformatics*.
61. Zhang, L., Zhang, C., Gao, R. and Yang, R. (2015) An Ensemble Method to Distinguish Bacteriophage Virion from Non-Virion Proteins Based on Protein Sequence Characteristics. *International journal of molecular sciences*, **16**, 21734-21758.
62. Garg, A. and Gupta, D. (2008) VirulentPred: a SVM based prediction method for virulent proteins in bacterial pathogens. *BMC bioinformatics*, **9**, 62.
63. Wei, L., Zhou, C., Chen, H., Song, J. and Su, R. (2018) ACPred-FL: a sequence-based predictor based on effective feature representation to improve the prediction of anti-cancer peptides. *Bioinformatics*.

3.2

BastionHub: A universal platform for integrating and analyzing substrates secreted by Gram-negative bacteria

BastionHub: a universal platform for integrating and analyzing substrates secreted by Gram-negative bacteria

Jiawei Wang[1], Jiahui Li[1,2,3], Ruopeng Xie[3], Yi Hou[3], Tatiana T. Marquez-Lago[4,5], André Leier[4,5], Tieli Zhou[2], Von Torres[1], Iain Hay[1], Yanju Zhang[3], Jiangning Song[6,7,8,*] and Trevor Lithgow[1,*]

[1]Infection and Immunity Program, Biomedicine Discovery Institute and Department of Microbiology, Monash University, VIC 3800, Australia;
[2]Department of Clinical Laboratory, The First Affiliated Hospital of Wenzhou Medical University, Wenzhou, Zhejiang Province, China;
[3]School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, 541004, China;
[4]Department of Genetics, School of Medicine, University of Alabama at Birmingham, AL, USA;
[5]Department of Cell, Developmental and Integrative Biology, School of Medicine, University of Alabama at Birmingham, AL, USA;
[6]Monash Centre for Data Science, Faculty of Information Technology, Monash University, VIC 3800, Australia;
[7]Infection and Immunity Program, Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University, VIC 3800, Australia;
[8]ARC Centre of Excellence in Advanced Molecular Imaging, Monash University, VIC 3800, Australia.

* To whom correspondence should be addressed to: Jiangning.Song@monash.edu; Trevor.Lithgow@monash.edu.

Abstract

Gram-negative bacteria utilize secretion systems to export substrates, so-called effectors, into their environment and sometimes directly inject these substrates into neighboring target cells. These substrates play pivotal roles in interactions between pathogens and their hosts, competitors, and with the external environment. Resulting from a rapid development of computational and experimental techniques in the field, a growing number of substrates have been discovered. To date, several online knowledgebases have been developed to present substrates with a focus on experimentally validated attributes and functions, as to maximize users' convenience in exploring known substrates. However, these resources usually focus on one (or a few) types of substrates and present limited options for users to analyze known and potential substrates. In response, we developed a universal platform, BastionHub, which integrates various types of substrates with detailed annotations from five different secretion systems (types I-IV and VI). To the best of our knowledge, BastionHub is the most comprehensive knowledgebase available, and it is the first to cover type I and type II secreted substrates while simultaneously providing updated information and increased numbers of other types of substrates. Moreover, BastionHub provides and extends basic functions, and integrates new tools to facilitate sequence analysis of both known and potential substrates: (i) the prediction module, including machine learning based prediction and hidden Markov model (HMM) based prediction; (ii) a relationship analysis, including BLAST-based similarity analysis and multiple alignment based phylogenetic analysis. By systematically integrating these modules as both a universal and interconnected platform, BastionHub facilitates analysis of known substrates and allows prediction of potential substrates and identification of their relationships in terms of sequence and phylogenetic similarity. BastionHub is freely available at http://bastionhub.erc.monash.edu/.

1. Introduction

Secretion systems are one of the key virulence "weapons" of bacteria, used to release numerous substrates into eukaryotic host cells or into neighboring bacterial cells to disrupt their cell biology (1). Following experimental discovery (2), these substrates have been classified into six distinct types according to their secretion systems (from type I to type VI, abbreviated as T1SS to T6SS). Among them, substrates secreted by T1SS (3), T2SS (4), T3SS (5), T4SS (6) and T6SS (7) are often named as type I, II, III, IV and VI secreted effectors (abbreviated as T1SE, T2SE, T3SE, T4SE and T6SE), respectively. A further group, the substrates secreted by the Type V secretion system (T5SS) (8,9) are a highly diverse set of sub-systems (T5SSa-T5SSf) with diverse functions, many of which are not released from the bacterial cell surface (10).

Proteins secreted by T1SS-T6SS are referred to generally as "substrates". A sub-set of these substrates are referred to as "effectors" but, specifically, this term is used for those substrates where the function is known to be a direct imitator of a host-cell functionality; effects thereby manipulate host-cell biology by mimicking a host-cell function. Hereafter, we used the general term substrate in this work, but still kept the term effector in the description of previous work. The substrates secreted by the T1SS and T2SS are usually enzymes, often hydrolases, and usually facilitate access and uptake of nutrients from the environment. The majority of studied substrates secreted by the T3SS, T4SS and T6SS have proved to be effectors (T3SE, T4SE and T6SE). Considering that secreted substrates vary in sequence, structure, secretion mechanism and function, it has been challenging to establish a universal platform that integrates various types of effectors with detailed attribute information, and to further provide analytic functions to assist users in systematic analysis and comparison of effectors by taking their functions, secreted types and bacterial species into consideration.

Considerable computational efforts have been put into collecting the various types of secreted effectors, and providing necessary functions supporting known effector analysis (**Table 1**) (11). Among them, T3SEdb (12), T3DB (13) and BEAN2.0 (14) collect and annotate T3SE proteins, but differ in the numbers of T3SEs and functions provided by each. Examples of other web-based resources are SecReT4 (15) and SecReT6 (16), which present T4SEs and T6SEs, respectively. SecretEPDB (17) further integrates previous known datasets and manually collects additional effectors to build a more universal resource for three types of effectors (T3SEs, T4SEs and T6SEs). EffectiveDB (18,19) is a database that provides a very large number of predicted T3SEs, T4SEs and T6SEs across multiple bacterial species,

picking up experimentally validated effectors but without means for browsing them or investigating their detailed information. Beyond providing data annotation and basic functions to investigate known secreted substrates, these toolkits offer various advanced functions to facilitate prediction of potential effectors. Specifically, to provide T3SE prediction, T3SEdb employs a selectable NaiveBayes or BayesNet model, T3DB integrates BPBAac (20), T3SEpre and a Markov model, and BEAN 2.0 integrates an updated model based on BEAN (21). Lastly, EffectiveDB integrates EffectiveT3 (22) and T4SEpre (23), and includes the algorithms EffectiveCCBD and EffectiveELD for T3SE or T4SE prediction.

Here, we present BastionHub, a universal platform to integrate and analyze various types of substrates secreted by Gram-negative bacteria. By manually mining current literature and curated data, we collected sequence information for those proteins that have been experimentally validated as secreted by T1SS and T2SS from a range of bacterial species. To the best of our knowledge this is the first time T1SS and T2SS substrate proteins are incorporated in a database. Further, by integrating current known datasets followed by manual checks, to remove incorrectly classified substrates, we obtained preliminary datasets for type III, IV and VI substrates. The latter were further supplemented with previously ignored substrates and recent experimentally validated substrates, through exhaustive literature screens. In total, BastionHub integrates five types of secreted substrate proteins, providing detailed sequence information, functional and structural annotations. To facilitate users' exploration and analysis of known substrate proteins, we incorporated basic functions including various data searches, download, and multiple visualizations. We then updated, integrated our developed machine learning based predictor BastionX and further developed hidden Markov model (HMM) based predictors to comprehensively predict all five types of substrates as a preliminary screening in a high-throughput manner. Finally, we integrated BLAST-based sequence similarity analysis and multiple alignment based phylogenetic analysis, allowing users to easily locate a potential homologs and functional analogs from comparisons to known and other substrates. By comprehensively integrating various types of secreted proteins alongside different functions, and providing interactive services as a pipeline across different functional modules, BastionHub aims to provide all-in-one service for users to analyze known substrates, predict potential effectors, and easily recognize their relationships.

Table 1. Comparison between BastionHub and previous web resources of various secreted proteins in Gram-negative bacteria.

Resource	Dataset[a]					Function					URL	Reference
	T1SE	T2SE	T3SE	T4SE	T6SE	Browse	Basic Search	Statistics	Prediction	Other functions		
T3SEdb	-	-	504	-	-	√	√	√	√[d]	BLAST based search	http://effectors.bic.nus.edu.sg/	(12)
T3DB	-	-	325	-	-	√	√	×	√[d]	-	https://biocomputer.bio.cuhk.edu.hk/T3DB/	(13)
BEAN 2.0	-	-	1215	-	-	-[b]	-[b]	×	√[d]	Subcellular location prediction Pfam domain scan Protein disorder prediction	http://systbio.cau.edu.cn/bean/	(14)
SecReT4	-	-	-	239	-	√	√	×	×	T4SS location T4SS component/effector search	http://db-mml.sjtu.edu.cn/SecReT4/	(15)
SecReT6	-	-	-	-	92	√	√	×	×	T6SS gene cluster detection and comparison Effector/Immunity/Component protein searches Bacterial genome rapid annotation	http://db-mml.sjtu.edu.cn/SecReT6/	(16)
EffectiveDB	-	-	-[c]	-[c]	-[c]	-[c]	-[c]	-[c]	√[d]	Subcellular location prediction	http://effectors.org/	(18,19)
SecretEPDB	-	-	1230	731	259	√	√	√	×	-	http://secretepdb.erc.monash.edu/	(17)
BastionHub	196	83	1236	731	195	√	√	√	√[d]	BLAST based similarity analysis Phylogenetic analysis	http://bastionhub.erc.monash.edu/	this work

Abbreviations: T*x*SE, type *x* secreted protein (*x*=1,2,3,4,6); T*x*SS, type *x* secretion system (*x*=1,2,3,4,6);
Note: [a]Only experimentally validated substrates were counted for each web resource;
[b]These functions were inaccessible at the time of undertaking this project;
[c]EffectiveDB doesn't have separate modules for experimentally validated effectors;
[d]T3SEdb can predict T3SEs using a NaiveBayes or BayesNet model; T3DB integrates BPBAac (20), T3SEpre and a Markov model for T3SE prediction; BEAN 2.0 integrates an updated model of BEAN (21) for T3SE prediction; EffectiveDB integrates EffectiveT3 (22) and T4SEpre (23), and develops EffectiveCCBD and EffectiveELD for T3SE/T4SE prediction; BastionHub integrates BastionX and developed a set of HMM based models to predict various types of secreted proteins.

2. The BastionHub framework

We wish to illustrate BastionHub's functionality in the following three ways (**Figure 1**): (i) data representation, including substrate data collection, curation, and annotation; (ii) data analysis, including known substrate investigation, potential substrate prediction, and relationship analysis between known and potential substrates; (iii) data pipeline: interactions between functional modules showcase how BastionHub facilitates substrate analysis through automatic operations running in the background.

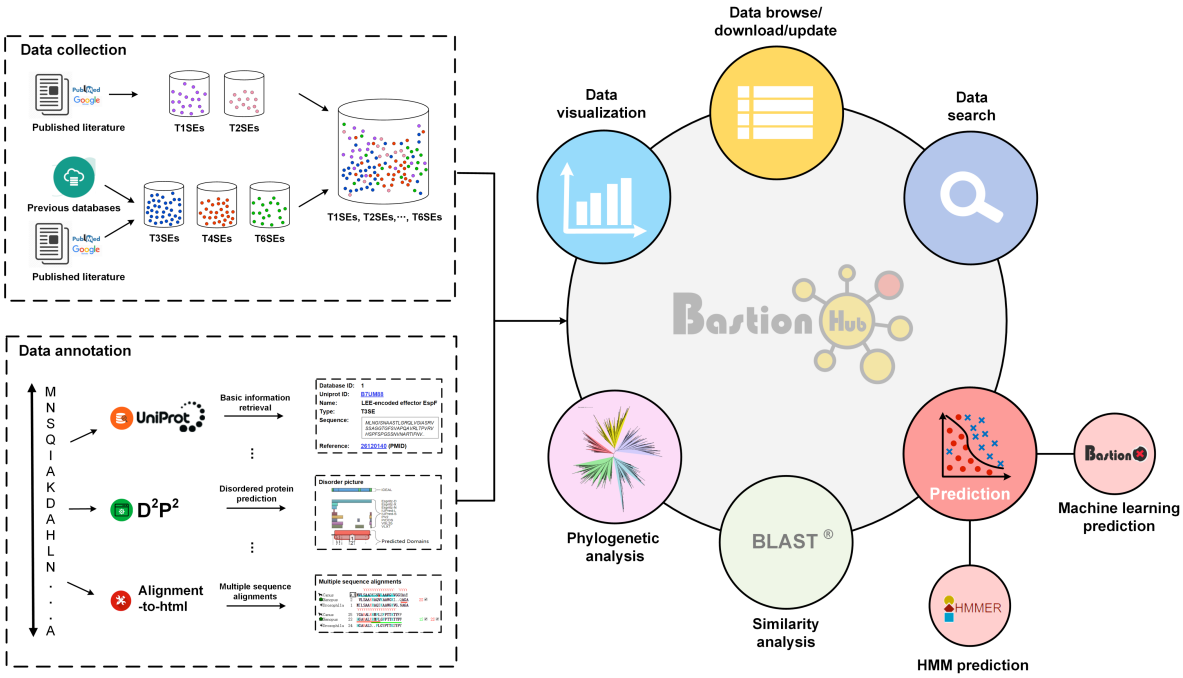


Figure 1. General framework showing how data construction, data annotations and various functions are provided by BastionHub.

2.1 Data representation

2.1.1 Data collection and curation

Development of universal web resources for type I and II secreted proteins had not been undertaken, largely due to the fact that there were no uniform names for these types of substrates, thus increasing difficulties in type I and II substrate retrieval. To fill this gap, we manually and thoroughly screened existing literature. These untargeted searches in a wide range resulted in more than 5000 non-repeating references retrieved. For each publication, we manually checked the abstract and main text, to obtain detailed information in aims to minimize potential mistakes of retrieving non-substrates, especially excluding any candidates without experimental validation. In total, 196 type I substrates across 63 species and 83 type II substrates across 13 species were obtained, after removing redundant entries.

Considering that there have been some web resources for type III, IV or VI substrates (**Table 1**), the retrieval of these three types of substrates could be divided into two steps. In step 1, for each type of substrates, we merged all publicly accessible datasets into a preliminary dataset. Through a manual check of these datasets, we carefully removed evident erroneous entries (e.g. membrane protein and secretion chaperone protein) based on BLAST alignment results against the UniProt database, and added the PubMed reference for each correct entry for future tracking and identification. In step 2, we conducted an exhaustive literature search, similar to that for T1SS and T2SS substrate retrieval, to obtain the previously missed substrates and the recent experimentally validated ones. In total, 1236 type III substrates across 65 species, 731 type IV substrates across 16 species and 195 type VI substrates across 69 species were obtained, after removing redundant entries.

Figure 2. Distribution (by secretion system types and bacterial species) of all 2441 substrate proteins covered in BastionHub. The doughnut chart illustrates the proportions of different types of substrates in BastionHub. Each subgraph shows the species distribution of one type of substances, among which the bar chart lists the numbers of secreted substrates per species and the pie chart present their percentages.

Altogether, we obtained 2441 substrates secreted by the five types of secretion systems across 168 species (**Fig. 2**). These were further annotated and organized in BastionHub, in a user-friendly manner, to facilitate users' understanding of diverse substrates in Gram-negative bacteria.

2.1.2 Data annotation

Beyond keeping detailed information that previous databases (i.e. SecretEPDB) provide, such as substrates' basic information, multiple alignment visualization or secondary structure and disordered region visualization, the following new features are present in BastionHub (**Fig. 3A**):

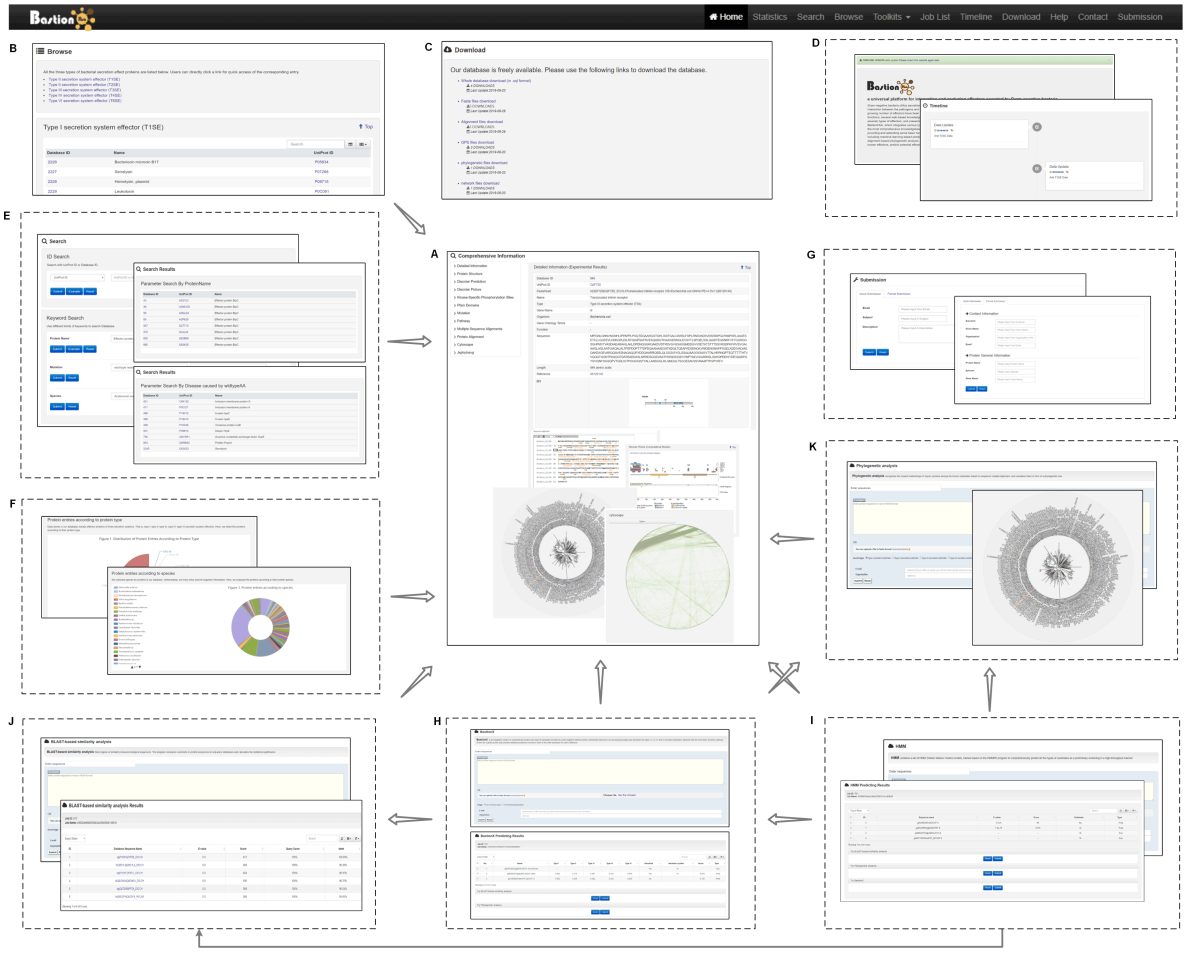


Figure 3. User interface of BastionHub. (A) The detailed information of substrates annotated in BastionHub. (B) The Browse web page. (C) The Download page. (D) The Timeline page and its update reminder popup at the main page. (E) The search page and its result page. (F) The statistics and visualization page. (G) The data submission page. (H) The input and output pages for the BastionX predictor. (I) The input and output pages for the HMM based predictor. (J) The input and output pages for the BLAST-based similarity analysis. (K) The input and output pages for the multiple alignment based phylogenetic analysis.

(1) For each substrate that was not retrieved in UniProt, BastionHub obtained an identical sequence in the same species by blasting it against UniProt, so that more information could be complemented for this secreted protein.

(2) For each substrate, the phylogenetic trees are provided using jsPhyloSVG (24) with this substrate highlighted, as to facilitate users locate it among its like and recognize its closest phylogenetic relationship.

(3) Similarly, each substrate's inter-species networks are provided using Cytoscape.js (25), with the substrate highlighted.

(4) For each substrate, the substrates with identical sequence content or high sequence similarity are provided, to inspire users to further investigate their attributes in a crosswise comparison.

2.2 Data analysis

2.2.1 Known substrate investigation

BastionHub provides a couple of functions to facilitate users' investigation of known substrates:

(1) ***Data browse***. Substrates are organized according to their associated types to provide a landscape of various substrates. A "search and filter" function is available in the BROWSE page, which enables users to narrow down the scopes of their interested substrates (**Fig. 3B**). By clicking the BastionHub ID, users can look into the detailed information of a substrate and refer to the original literature in PubMed to track this substrate's discovery and functional validation.

(2) ***Data download***. To facilitate users working with data in batch mode, whole datasets and related files can be downloaded, including the SQL format and FASTA format, multiple alignment files, phylogenetic tree structure files, and network structure files (**Fig. 3C**). All datasets are marked with download counts and last update time, and their update history is recorded in the TIMELINE module to help users identify their status (**Fig. 3D**).

(3) ***Data search***. Search conditions are refined to allow accurate searches by BastionHub ID and UniProt ID, and fuzzy searches by protein name, mutation and species (**Fig. 3E**).

(4) ***Data visualization***. We use ECharts (https://ecomfe.github.io/echarts-doc/public/en/index.html/) to visualize data statistics according to secreted types and species (**Fig. 3F**). Phylogenetic inter-species relationships are visualized by jsPhyloSVG, while network architectures within inter-species are visualized by Cytoscape.js.

(5) ***Data update***. Other than keeping track of new published literature, thoroughly checking and regularly updating the database, BastionHub allows users to contribute to the database update by means of crowdsourcing. Users may simply provide some clues that facilitate new substrate retrieval via the 'quick submission' module, or provide detailed information of a substrate, to speed up the substrate retrieval via the 'formal submission' module (**Fig. 3G**).

(6) ***Update reminder***. Beyond simply recording the information for database entry

alteration, as implemented in SecretEPDB, the TIMELINE module in BastionHub records the update history of both downloadable datasets and functional modules. In this way, users can easily and thoroughly comprehend latest changes to the database, from various angles. The latest development announcement will be presented in the homepage of BastionHub in the form of a pop-up box, providing a link to the TIMELINE module for detailed information (**Fig. 3D**).

2.2.2 Potential substrate prediction

Inference or prediction of new substrates, based on experimentally validated substrates, is an essential but challenging task. Both experimental and computational scientists have proposed a variety of solutions to recognize new substrates resulting from different types of secretion systems, such as sequence similarity based analysis (26,27), conserved domain based search (28,29), and machine learning based predictions (11,30). Aiming to provide integrative platform with all-in-one service for potential substrate prediction, BastionHub integrates two developed predictive modules as follows (**Fig. 4**):

(1) ***Machine learning based prediction***. BastionX is a newly developed machine learning based predictor within a distributed framework towards high-throughput prediction of various types of secreted substrates (**Fig. 3H**). Taking advantages of our previous singe type substrate predictors, *i.e.* Bastion3 (31), Bastion4 (32) and Bastion6 (33), BastionX further develops type I and II substrate predictors to comprehensively predict all types of substrates in Gram-negative bacteria. With additional options on its prediction outputs (**Fig. 3H**), BastionX can easily and seamlessly interact with other functional modules in BastionHub, and significantly reduce users' manual inputs and operations (illustrated in detail in Section 2.3).

(2) ***HMM based prediction.*** Although machine learning based predictors have demonstrated their prediction power of various types of substrates, they usually limit the number of input sequences per submission, due to underlying complex calculations. Accordingly, we constructed a set of HMM based models using HMMER (34) to predict potential type I, II, III, IV and VI substrates for preliminary control screening (**Fig. 3I**). These HMM based predictors allow users to submit genome-scale sequences and operate in a high-throughput manner. Generated results can be further processed by using machine learning based models, to increase accuracy of results (**Fig. 3I**). In this way, the tradeoff between prediction time and accuracy is balanced in BastionHub.

2.2.3 Relationship analysis between potential and known substrates

Considering that substrates with similar sequences may have similar structures, and usually similar functions as well, analyzing the relationship between potential substrates and known substrates may assist inference of possible structures and functions of potential substrates, based on that of known substrates. However, based on the patterns and characteristics learned from known substrate datasets, machine learning and HMM based models can only recognize

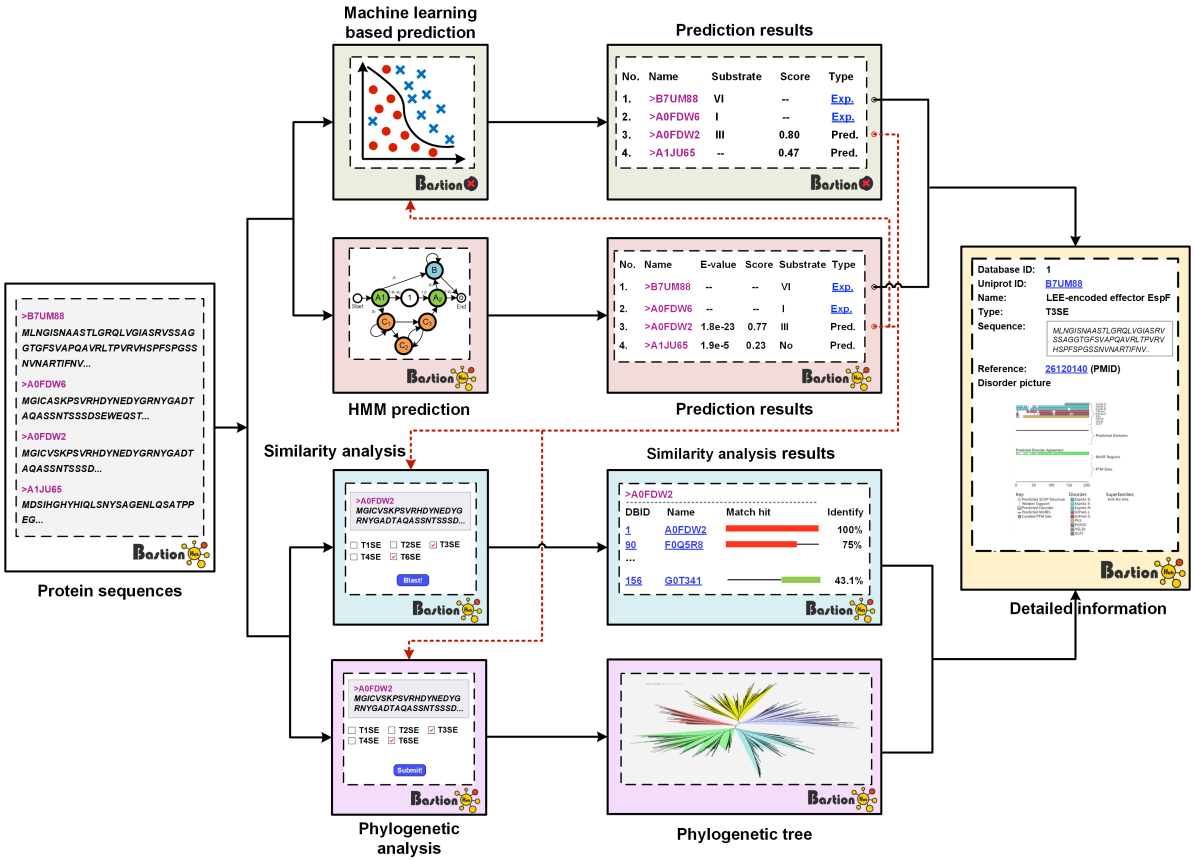


Figure 4. Illustration of BastionHub functional modules and their interactions. Solid lines indicate procedures in each functional model operating as an independent toolkit, while dotted lines highlight interactions between different functional modules.

potential substrates without specifying their closest relationship among known substrates. We therefore developed two relationship analysis modules (**Fig. 4**), to select a potential substrate's closest analogues out of known substrates. Those analogues and their detailed information, presented in BastionHub, are intended to inspire users to infer possible structural and functional attributes of a potential substrate, and guide design of experimental validation protocols.

(1) ***BLAST-based similarity analysis***. For a potential substrate (also referred to as an inquiry protein), BastionHub can search this protein sequence against a user-selected specific dataset (i.e. type I, II, III, IV or VI substrates or their combinations) using BLAST (35) (**Fig.**

3J). In this way, one can check if a potential substrate is homologous to any known substrates. All hits, namely resulting known substrates, will be listed (**Fig. 3J**). These will be sorted by similarity significance, and their BastionHub ID will also be provided, allowing users to further investigate detailed information such as structures and functions.

(2) ***Multiple alignment based phylogenetic analysis***. For a potential substrate, BastionHub first utilizes Maffet to search this protein against user-selected dataset (i.e. type I, II, III, IV or VI substrates or their combination). This allows generation of multiple alignment results, based on phylogenetic tree structure inferences by FastTree (36). Finally, the phylogenetic relationship between a potential substrate and the selected known substrates is obtained in form of a phylogenetic tree by using jsPhyloSVG (**Fig. 3K**). Within the presented phylogenetic tree, the potential substrate is highlighted, and links to the known substrates (identified by BastionHub ID) are provided to facilitate tracking their corresponding detailed information (**Fig. 3K**).

2.3 Data pipeline

All functional modules in BastionHub are designed and implemented as independent services; consequently, each can serve users as a standalone toolkit. With that said, there are natural interactions between different modules that are worth highlighting (**Fig. 4**): (i) potential substrates predicted by HMM based models may be fed into machine learning based models for more accurate outcomes; (ii) properties of predicted substrates from both HMM based models and machine learning based models may be fed into relationship analysis modules to observe their relationships with known substrates. To facilitate users' mining of potential and known substrates, BastionHub includes convenient interactive services between various modules.

(1) BastionHub provides functions allowing users to feed some or all of the identified potential substrates (predicted by HMM based models, in the prediction results page) as inputs to BastionX (**Figs. 3I and 4**). Without individually selecting these predicted potential substrates out of the prediction result generated by HMM based models, often followed by a manual copy-and-paste operation to fill them into BastionX, these potential substrates could be automatically gathered by BastionHub and then filled into the input filed of BastionX predictors for subsequent machine learning based prediction (**Fig. 3H**).

(2) When predicting potential substrates (using HMM-based models or the BastionX predictor), BastionHub first searches them against the known substrates. Whenever sequences are identified as known substrates, these are marked as "*Exp.*" in the prediction results, with

links to the corresponding detailed information for users' reference (**Fig. 3H-I**). For potential substrates (marked as "*Pred.*") predicted by either HMM based models or the BastionX predictor, BastionHub provides links on the prediction result page, allowing downstream relationship analysis between selected potential substrates and known substrates (**Figs. 3H-K and 4**). With all selected potential substrates automatically fed to the input field, users may select a known substrate dataset to launch the relationship analysis.

Altogether, BastionHub provides a user-friendly, intuitive, interconnected platform allowing analysis of known bacterial substrates (types I-IV, VI), prediction of potential substrates, and identification of relationships, to fit users' specific demands.

3. Conclusion

In this work, we present BastionHub, a universal platform developed with the intention to integrate and analyze various types of substrates secreted by Gram-negative bacteria. By comprehensively integrating various types of substrates, providing a range of functional modules, and interconnecting them, BastionHub provides users with an all-in-one service to facilitate known substrate investigation, potential substrate prediction, and relationship analyses between known substrates and potential substrates.

Acknowledgements

This work was financially supported by grants from the National Health and Medical Research Council of Australia (NHMRC) (1092262, 1144652 and 1127948), the Australian Research Council (ARC) (LP110200333 and DP120104460), the National Institute of Allergy and Infectious Diseases of the National Institutes of Health (R01 AI111965), and the Collaborative Research Program of Institute for Chemical Research, the National Natural Science Foundation of China (61862017), and the Natural Science Foundation of Guangxi (2018GXNSFAA138117, 2016GXNSFCA380005). TML and AL's work was supported in part by the Informatics Institute of the School of Medicine at UAB. T.L. is an ARC Australian Laureate Fellow (FL130100038).

Conflict of Interest: none declared.

References

1. Wandersman, C. (2013) Concluding remarks on the special issue dedicated to bacterial secretion systems: function and structural biology. *Research in microbiology*, **164**, 683-687.
2. Costa, T.R., Felisberto-Rodrigues, C., Meir, A., Prevost, M.S., Redzej, A., Trokter, M. and Waksman, G. (2015) Secretion systems in Gram-negative bacteria: structural and mechanistic insights. *Nature reviews. Microbiology*, **13**, 343-359.
3. Welch, R.A., Dellinger, E.P., Minshew, B. and Falkow, S. (1981) Haemolysin contributes to virulence of extra-intestinal E. coli infections. *Nature*, **294**, 665-667.
4. d'Enfert, C., Ryter, A. and Pugsley, A.P. (1987) Cloning and expression in Escherichia coli of the Klebsiella pneumoniae genes for production, surface localization and secretion of the lipoprotein pullulanase. *EMBO J*, **6**, 3531-3538.
5. Galan, J.E. and Curtiss, R., 3rd. (1989) Cloning and molecular characterization of genes whose products allow Salmonella typhimurium to penetrate tissue culture cells. *Proceedings of the National Academy of Sciences of the United States of America*, **86**, 6383-6387.
6. Kuldau, G.A., De Vos, G., Owen, J., McCaffrey, G. and Zambryski, P. (1990) The virB operon of Agrobacterium tumefaciens pTiC58 encodes 11 open reading frames. *Mol Gen Genet*, **221**, 256-266.
7. Pukatzki, S., Ma, A.T., Sturtevant, D., Krastins, B., Sarracino, D., Nelson, W.C., Heidelberg, J.F. and Mekalanos, J.J. (2006) Identification of a conserved bacterial protein secretion system in Vibrio cholerae using the Dictyostelium host model system. *Proceedings of the National Academy of Sciences of the United States of America*, **103**, 1528-1533.
8. Fan, E., Chauhan, N., Udatha, D.B., Leo, J.C. and Linke, D. (2016) Type V Secretion Systems in Bacteria. *Microbiol Spectr*, **4**.
9. Heinz, E., Stubenrauch, C.J., Grinter, R., Croft, N.P., Purcell, A.W., Strugnell, R.A., Dougan, G. and Lithgow, T. (2016) Conserved Features in the Structure, Mechanism, and Biogenesis of the Inverse Autotransporter Protein Family. *Genome Biol Evol*, **8**, 1690-1705.
10. Nicolay, T., Vanderleyden, J. and Spaepen, S. (2015) Autotransporter-based cell surface display in Gram-negative bacteria. *Crit Rev Microbiol*, **41**, 109-123.
11. Zeng, C. and Zou, L. (2017) An account of in silico identification tools of secreted effector proteins in bacteria and future challenges. *Briefings in bioinformatics*.
12. Tay, D.M., Govindarajan, K.R., Khan, A.M., Ong, T.Y., Samad, H.M., Soh, W.W., Tong, M., Zhang, F. and Tan, T.W. (2010) T3SEdb: data warehousing of virulence effectors secreted by the bacterial Type III Secretion System. *BMC bioinformatics*, **11 Suppl 7**, S4.
13. Wang, Y., Huang, H., Sun, M.a., Zhang, Q. and Guo, D. (2012) T3DB: an integrated database for bacterial type III secretion system. *BMC bioinformatics*, **13**, 66.
14. Dong, X., Lu, X. and Zhang, Z. (2015) BEAN 2.0: an integrated web resource for the identification and functional analysis of type III secreted effectors. *Database : the journal of biological databases and curation*, **2015**, bav064.
15. Bi, D., Liu, L., Tai, C., Deng, Z., Rajakumar, K. and Ou, H.Y. (2013) SecReT4: a web-based bacterial type IV secretion system resource. *Nucleic acids research*, **41**, D660-665.

16. Li, J., Yao, Y., Xu, H.H., Hao, L., Deng, Z., Rajakumar, K. and Ou, H.Y. (2015) SecReT6: a web-based resource for type VI secretion systems found in bacteria. *Environmental microbiology*, **17**, 2196-2202.
17. An, Y., Wang, J., Li, C., Revote, J., Zhang, Y., Naderer, T., Hayashida, M., Akutsu, T., Webb, G.I., Lithgow, T. *et al.* (2017) SecretEPDB: a comprehensive web-based resource for secreted effector proteins of the bacterial types III, IV and VI secretion systems. *Scientific reports*, **7**, 41031.
18. Jehl, M.A., Arnold, R. and Rattei, T. (2011) Effective--a database of predicted secreted bacterial proteins. *Nucleic acids research*, **39**, D591-595.
19. Eichinger, V., Nussbaumer, T., Platzer, A., Jehl, M.-A., Arnold, R. and Rattei, T. (2016) EffectiveDB—updates and novel features for a better annotation of bacterial secreted proteins and Type III, IV, VI secretion systems. *Nucleic acids research*, **44**, D669-D674.
20. Wang, Y., Zhang, Q., Sun, M.A. and Guo, D. (2011) High-accuracy prediction of bacterial type III secreted effectors based on position-specific amino acid composition profiles. *Bioinformatics*, **27**, 777-784.
21. Dong, X., Zhang, Y.J. and Zhang, Z. (2013) Using weakly conserved motifs hidden in secretion signals to identify type-III effectors from bacterial pathogen genomes. *PloS one*, **8**, e56632.
22. Arnold, R., Brandmaier, S., Kleine, F., Tischler, P., Heinz, E., Behrens, S., Niinikoski, A., Mewes, H.W., Horn, M. and Rattei, T. (2009) Sequence-based prediction of type III secreted proteins. *PLoS pathogens*, **5**, e1000376.
23. Wang, Y., Wei, X., Bao, H. and Liu, S.L. (2014) Prediction of bacterial type IV secreted effectors by C-terminal features. *BMC Genomics*, **15**, 50.
24. Smits, S.A. and Ouverney, C.C. (2010) jsPhyloSVG: a javascript library for visualizing interactive and vector-based phylogenetic trees on the web. *PloS one*, **5**, e12267.
25. Franz, M., Lopes, C.T., Huck, G., Dong, Y., Sumer, O. and Bader, G.D. (2016) Cytoscape.js: a graph theory library for visualisation and analysis. *Bioinformatics*, **32**, 309-311.
26. Ma, J., Sun, M., Dong, W., Pan, Z., Lu, C. and Yao, H. (2017) PAAR-Rhs proteins harbor various C-terminal toxins to diversify the antibacterial pathways of type VI secretion systems. *Environmental microbiology*, **19**, 345-360.
27. Bondage, D.D., Lin, J.S., Ma, L.S., Kuo, C.H. and Lai, E.M. (2016) VgrG C terminus confers the type VI effector transport specificity and is required for binding with PAAR and adaptor-effector complex. *Proceedings of the National Academy of Sciences of the United States of America*, **113**, E3931-3940.
28. Salomon, D., Kinch, L.N., Trudgian, D.C., Guo, X., Klimko, J.A., Grishin, N.V., Mirzaei, H. and Orth, K. (2014) Marker for type VI secretion system effectors. *Proceedings of the National Academy of Sciences of the United States of America*, **111**, 9271-9276.
29. Altindis, E., Dong, T., Catalano, C. and Mekalanos, J. (2015) Secretome analysis of Vibrio cholerae type VI secretion system reveals a new effector-immunity pair. *mBio*, **6**, e00075.
30. An, Y., Wang, J., Li, C., Leier, A., Marquez-Lago, T., Wilksch, J., Zhang, Y., Webb, G.I., Song, J. and Lithgow, T. (2018) Comprehensive assessment and performance improvement of effector protein predictors for bacterial secretion systems III, IV and VI. *Briefings in bioinformatics*, **19**, 148-161.

31. Wang, J., Li, J., Yang, B., Xie, R., Marquez-Lago, T.T., Leier, A., Hayashida, M., Akutsu, T., Zhang, Y., Chou, K.C. *et al.* (2019) Bastion3: a two-layer ensemble predictor of type III secreted effectors. *Bioinformatics*, **35**, 2017-2028.
32. Wang, J., Yang, B., An, Y., Marquez-Lago, T., Leier, A., Wilksch, J., Hong, Q., Zhang, Y., Hayashida, M., Akutsu, T. *et al.* (2019) Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches. *Briefings in bioinformatics*, **20**, 931-951.
33. Wang, J., Yang, B., Leier, A., Marquez-Lago, T.T., Hayashida, M., Rocker, A., Zhang, Y., Akutsu, T., Chou, K.C., Strugnell, R.A. *et al.* (2018) Bastion6: a bioinformatics approach for accurate prediction of type VI secreted effectors. *Bioinformatics*, **34**, 2546-2555.
34. Eddy, S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755-763.
35. Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K. and Madden, T.L. (2009) BLAST+: architecture and applications. *BMC bioinformatics*, **10**, 421.
36. Price, M.N., Dehal, P.S. and Arkin, A.P. (2010) FastTree 2--approximately maximum-likelihood trees for large alignments. *PloS one*, **5**, e9490.

CHAPTER 4: Computational toolkits to facilitate development of machine learning based predictors

As a fundamental step in the construction of high-quality machine learning-based predictors, feature extraction is key to ensure the effectiveness of their applications in bioinformatics and biomedicine. Towards streamlined automatic feature extraction, feature generating toolkits are desired to accelerate the development and elevate the predictive performance of machine learning models by sparing users from complex and arcane mathematical formula and expertised programming implementations.

To this end, this chapter presents two feature generating toolkits that have been developed to expedite machine learning based modelling and analysis. Section 4.1 describes the first specialized bioinformatics toolkit suite POSSUM for generating numerical features based on PSSM profiles from protein sequences. Both its online webserver and local standalone software enable users to generate more than 20 types of PSSM profile-based features, which substantially addresses a crucial need for bioinformaticians and computational biologists. Section 4.2 describes a universal feature generating toolkit, DIFFUSER, for generating a great variety of machine learning features based on protein, DNA and RNA sequences. Compared to POSSUM, DIFFUSER represents a remarkable enhancement and upgrade, as it both enables more some comprehensive feature generation, which it is a common requirement in practical applications, and allows high-throughput and genome-scale feature generation that is becoming ever critical in the era of big data. Although both toolkits were initially developed to facilitate the development of substrate predictors, they could be commonly applied in machine learning related studies and therefore contribute to more effective analysis and modeling in general bioinformatics and biomedicine research.

4.1

POSSUM: A bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles

The supplementary information for this manuscript is listed in **Appendix 5**.

Bioinformatics, 33(17), 2017, 2756–2758
doi: 10.1093/bioinformatics/btx302
Advance Access Publication Date: 10 May 2017
Applications Note

Sequence analysis

POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles

Jiawei Wang[1,†], Bingjiao Yang[2,†], Jerico Revote[1], André Leier[3], Tatiana T. Marquez-Lago[3], Geoffrey Webb[4], Jiangning Song[1,4,5,*], Kuo-Chen Chou[6,7,8] and Trevor Lithgow[1,*]

[1]Biomedicine Discovery Institute, Monash University, VIC 3800, Australia, [2]College of Mechanical Engineering, Yanshan University, Qinhuangdao 066004, China, [3]Informatics Institute and Department of Genetics, School of Medicine, University of Alabama at Birmingham, Birmingham, AL, USA, [4]Monash Centre for Data Science, Faculty of Information Technology, [5]ARC Centre of Excellence for Advanced Molecular Imaging, Monash University, VIC 3800, Australia, [6]Gordon Life Science Institute, Boston, MA 02478, USA, [7]Center for Informational Biology, University of Electronic Science and Technology of China, Chengdu 610054, China and [8]Center of Excellence in Genomic Medicine Research (CEGMR), King Abdulaziz University, Jeddah 21589, Saudi Arabia

*To whom correspondence should be addressed.
[†]The authors wish it to be known that these authors contributed equally.
Associate Editor: John Hancock

Received on March 12, 2017; revised on April 14, 2017; editorial decision on April 30, 2017; accepted on May 9, 2017

Abstract

Summary: Evolutionary information in the form of a Position-Specific Scoring Matrix (PSSM) is a widely used and highly informative representation of protein sequences. Accordingly, PSSM-based feature descriptors have been successfully applied to improve the performance of various predictors of protein attributes. Even though a number of algorithms have been proposed in previous studies, there is currently no universal web server or toolkit available for generating this wide variety of descriptors. Here, we present POSSUM (Position-Specific Scoring matrix-based feature generator for machine learning), a versatile toolkit with an online web server that can generate 21 types of PSSM-based feature descriptors, thereby addressing a crucial need for bioinformaticians and computational biologists. We envisage that this comprehensive toolkit will be widely used as a powerful tool to facilitate feature extraction, selection, and benchmarking of machine learning-based models, thereby contributing to a more effective analysis and modeling pipeline for bioinformatics research.
Availability and implementation: http://possum.erc.monash.edu/.
Contact: trevor.lithgow@monash.edu or jiangning.song@monash.edu
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Feature extraction or feature encoding is a fundamental step in the construction of high-quality machine learning-based models. Specifically, this step is key to determining the effectiveness of trained models in bioinformatics applications (Chou, 2011). In the last two decades, a variety of feature encoding schemes have been proposed in order to exploit useful patterns from protein sequences. Such schemes are often based on sequence information or representation of physicochemical properties. Although direct features derived from sequences themselves (such as amino acid compositions, dipeptide compositions and counting of *k*-mers) are regarded as essential for training models, an increasing number of studies

have shown that evolutionary information in the form of PSSM profiles is much more informative than sequence information alone (An *et al.*, 2016). Accordingly, PSSM-based feature descriptors have been commonly used as indispensable primary features to construct models, filling a major gap in the current bioinformatics research. For example, PSSM-based feature descriptors have successfully improved the prediction performance of structural and functional properties of proteins across a wide spectrum of bioinformatics applications (See Supplementary Table S1 in the Supplementary Material for a comprehensive lists of applications). These include for example protein fold recognition (Lobley *et al.*, 2009) and the prediction of protein structural classes (Liu *et al.*, 2010), protein-protein interactions (Zahiri *et al.*, 2013), protein subcellular localization (Xie *et al.*, 2005), RNA-binding sites (Cheng *et al.*, 2008) and protein functions (Radivojac *et al.*, 2013), to name a few.

A number of servers and standalone software packages have been developed to derive a variety of feature descriptors from protein, DNA and RNA sequences, including PROFEAT (Rao *et al.*, 2011), PseAAC (Shen and Chou, 2008), propy (Cao *et al.*, 2013), repDNA (Liu *et al.*, 2015), protr/ProtrWeb (Xiao *et al.*, 2015), Pse-in-One (Liu *et al.*, 2015; Liu *et al.*, 2017a), repRNA (Liu *et al.*, 2016) and Pse-Analysis (Liu *et al.*, 2017b). Despite their usefulness and popularity, these tools primarily focus on the generation of features related to sequence-based and/or physicochemical descriptors, instead of PSSM profile-based features. Indeed, there are over 20 different PSSM-based algorithms that calculate and model PSSM-based feature descriptors. However, to the best of our knowledge, there is currently no consolidated web server or toolkit available for generating these PSSM-based feature descriptors. Here, we present a bioinformatics toolkit, POSSUM, an effective tool that enables users to generate a broad spectrum of PSSM-based numerical representation schemes for protein sequences. It implements a wide range of algorithms available in the literature, provides an easy-to-use interface, and offers much needed functionality and flexibility for users to derive and customize these descriptors. We demonstrate the usage of POSSUM-calculated PSSM features for the prediction of bacterial secretion effector proteins (cf. Supplementary Material results).

2 Implementation

The POSSUM server consists of two major components: the client web interface and the server backend (See Supplementary Fig. S1). The former was implemented using jQuery, Bootstrap, Struts and Hibernate. Users can interact with the client web interface to input their protein sequences and choose the specific feature descriptors to be generated. Submitted jobs are then forwarded to the server backend. For the latter, a Perl CGI program lines up submitted jobs in a queue and invokes a Perl daemon thread for each job to execute the descriptor generation process. This architecture guarantees that multiple jobs can be executed simultaneously, within the maximum number of allowed threads predefined in the server, while any remaining jobs are queued until processing slots become available.

With the client web interface, users can upload a protein sequence file in the FASTA format, or directly input protein sequences (Supplementary Figs S2 and S3). Next, users need to customize parameters to generate PSSM profiles, which is followed by selection of the feature descriptors needed to be calculated. POSSUM generates PSSM profiles of the submitted sequences by running PSI-BLAST. Depending on the length of the input protein sequence, the PSSM profile generation process can be computationally time-consuming. To address this issue, we implemented a caching module in POSSUM, allowing re-use of generated PSSM profiles instead of computing them again. Based on the PSSM profiles, POSSUM can calculate the corresponding feature descriptors in the background inside the server backend. Users do not need to wait for job progress: they can track the progress of submitted jobs through a unique link, or be informed by email (if they opted for this in the client interface) once their jobs are finished. Both the raw PSSM files and resulting descriptors can then be downloaded from their unique link.

Table 1. A full list of PSSM-based feature descriptors that can be generated by POSSUM

Descriptors groups	Descriptor	Number	Original
Row transformations	AAC-PSSM	20	(Liu *et al.*, 2010)
	D-FPSSM	20	(Zahiri *et al.*, 2013)
	smoothed-PSSM	–[a]	(Cheng *et al.*, 2008)
	AB-PSSM	400	(Jeong *et al.*, 2011)
	PSSM-composition	400	(Zou *et al.*, 2013)
	RPM-PSSM	400	(Jeong *et al.*, 2011)
	S-FPSSM	400	(Zahiri *et al.*, 2013)
Column transformations	DPC-PSSM	400	(Liu *et al.*, 2010)
	k-separated-bigrams-PSSM	400	(Saini *et al.*, 2016)
	tri-gram-PSSM	8000	(Paliwal *et al.*, 2014)
	EEDP	400	(Zhang *et al.*, 2014)
	TPC	400	(Zhang *et al.*, 2012)
Mixture of row and column transformations	EDP	20	(Zhang *et al.*, 2014)
	RPSSM	110	(Ding *et al.*, 2014)
	Pse-PSSM	40	(Chou and Shen, 2007)
	DP-PSSM	–[a]	(Juan *et al.*, 2009)
	PSSM-AC	–[a]	(Dong *et al.*, 2009)
	PSSM-CC	–[a]	(Dong *et al.*, 2009)
Combination of above descriptors	AADP-PSSM	420	(Liu *et al.*, 2010)
	AATP	420	(Zhang *et al.*, 2012)
	MEDP	420	(Zhang *et al.*, 2014)

[a]The number of feature descriptor values depends on the choice of the parameter.

For users who prefer to apply their own parameter settings for specific research purposes and users who have the capacity to perform high throughput generation of PSSM files for a very large dataset using their local computers, an open source standalone software toolkit is also available. The standalone version of POSSUM (See Supplementary Fig. S4) was developed using Python and Perl, and can be executed on Unix/Linux, Windows and Mac OS. As an open source software, users can access, modify and redistribute the source codes, allowing users to tailor POSSUM according to their specific requirements.

PSSM-based algorithms are based on matrix transformations from original PSSM profiles, which can be categorized into three types: row transformations, column transformations, or a mixture of row and column transformations. For POSSUM, these descriptors are divided into four groups (Table 1). The first group consists of AAC-PSSM, D-FPSSM, smoothed-PSSM, AB-PSSM, PSSM-composition, RPM-PSSM and S-FPSSM, which are generated by row transformations of the original PSSM. The second group contains the descriptors generated by column transformations, including DPC-PSSM, k-separated-bigrams-PSSM, tri-gram-PSSM, EEDP and TPC. The third group includes EDP, RPSSM, Pse-PSSM, DP-PSSM, PSSM-AC and PSSM-CC, which are generated by a mixture of row and column transformations. The fourth group comprises of AADP-PSSM, AATP and MEDP, which simply combine descriptors in the former three groups.

3 Results

In this work, we present POSSUM, a comprehensive, flexible, user-friendly and publicly accessible toolkit (with both local standalone software and online webserver) that we developed to allow users to easily generate more than 20 types of PSSM profile-based feature descriptors. It greatly facilitates feature generation, analysis, training and benchmarking of machine-learning models and predictions. POSSUM has been extensively benchmarked to guarantee correctness of computations, and was deliberately designed to ensure workflow efficiency. To the best of our knowledge, this is the first toolkit for generating such a great variety of evolutionary feature descriptors. Future work will include parallelization of PSSM profile generation to improve the throughput of POSSUM server. POSSUM is freely accessible at http://possum.erc.monash.edu/.

Acknowledgements

This work was supported by grants from the National Health and Medical Research Council of Australia (NHMRC) (1092262) and the Australian Research Council (ARC). G.I.W. is a recipient of Discovery Outstanding Research Award (DORA) of the ARC. T.L. is an ARC Australian Laureate Fellow.

Conflict of Interest: none declared.

References

An,Y. *et al*. (2016) Comprehensive assessment and performance improvement of effector protein predictors for bacterial secretion systems III, IV and VI. *Brief Bioinform*., bbw100.

Cao,D.S. *et al*. (2013) Propy: a tool to generate various modes of Chou's PseAAC. *Bioinformatics*, **29**, 960–962.

Cheng,C.W. *et al*. (2008) Predicting RNA-binding sites of proteins using support vector machines and evolutionary information. *BMC Bioinform.*, **9**, S6.

Chou,K.-C. (2011) Some remarks on protein attribute prediction and pseudo amino acid composition. *J. Theor. Biol*., **273**, 236–247.

Chou,K.C. and Shen,H.B. (2007) MemType-2L: a web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. *Biochem. Biophys. Res. Commun*., **360**, 339–345.

Ding,S. *et al*. (2014) A protein structural classes prediction method based on predicted secondary structure and PSI-BLAST profile. *Biochimie*, **97**, 60–65.

Dong,Q. *et al*. (2009) A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. *Bioinformatics*, **25**, 2655–2662.

Jeong,J.C. *et al*. (2011) On position-specific scoring matrix for protein function prediction. *IEEE/ACM Trans. Comput. Biol. Bioinform./IEEE, ACM*, **8**, 308–315.

Juan,E.Y. *et al*. (2009) Predicting protein subcellular localizations for gram-negative bacteria using dp-pssm and support vector machines. In: *International Conference on Complex, Intelligent and Software Intensive Systems, 2009. CISIS'09*, pp. 836–841. IEEE Press, Fukuoka, Japan.

Liu,B. *et al*. (2015) repDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects. *Bioinformatics*, **31**, 1307–1309.

Liu,B. *et al*. (2016) repRNA: a web server for generating various feature vectors of RNA sequences. *Mol. Genet. Genom. MGG*, **291**, 473–481.

Liu,B. *et al*. (2015) Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nucleic Acids Res*., **43**, W65–W71.

Liu,B. *et al*. (2017a) Pse-in-One 2.0: an improved package of web servers for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Natural Science*, **9**, 67.

Liu,B. *et al*. (2017b) Pse-Analysis: a python package for DNA/RNA and protein/peptide sequence analysis based on pseudo components and kernel methods. *Oncotarget*, **8**, 13338–13343.

Liu,T. *et al*. (2010) Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile. *Biochimie*, **92**, 1330–1334.

Lobley,A. *et al*. (2009) pGenTHREADER and pDomTHREADER: new methods for improved protein fold recognition and superfamily discrimination. *Bioinformatics*, **25**, 1761–1767.

Paliwal,K.K. *et al*. (2014) A tri-gram based feature extraction technique using linear probabilities of position specific scoring matrix for protein fold recognition. *IEEE Trans. Nanobiosci*., **13**, 44–50.

Radivojac,P. *et al*. (2013) A large-scale evaluation of computational protein function prediction. *Nat. Methods*, **10**, 221–227.

Rao,H.B. *et al*. (2011) Update of PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic Acids Res*., **39**, W385–W390.

Saini,H. *et al*. (2016) Protein fold recognition using genetic algorithm optimized voting scheme and profile bigram. *J Softw.*, **11**, 756–767.

Shen,H.B. and Chou,K.C. (2008) PseAAC: a flexible web server for generating various kinds of protein pseudo amino acid composition. *Anal. Biochem*., **373**, 386–388.

Xiao,N. *et al*. (2015) protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences. *Bioinformatics*, **31**, 3555–3557.

Xie,D. *et al*. (2005) LOCSVMPSI: a web server for subcellular localization of eukaryotic proteins using SVM and profile of PSI-BLAST. *Nucleic Acids Res*., **33**, W105–W110.

Zahiri,J. *et al*. (2013) PPIevo: protein-protein interaction prediction from PSSM based evolutionary information. *Genomics*, **102**, 237–242.

Zhang,L. *et al*. (2014) Predict protein structural class for low-similarity sequences by evolutionary difference information into the general form of Chou's pseudo amino acid composition. *J. Theor. Biol*., **355**, 105–110.

Zhang,S. *et al*. (2012) Using principal component analysis and support vector machine to predict protein structural class for low-similarity sequences via PSSM. *J. Biomol. Struct. Dyn*., **29**, 634–642.

Zou,L. *et al*. (2013) Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles. *Bioinformatics*, **29**, 3135–3142.

4.2

DIFFUSER: A distributed framework to generate machine learning features based on protein, DNA and RNA sequences

The supplementary information for this manuscript is listed in **Appendix 6**.

DIFFUSER: A distributed framework for high-throughput generation of machine-learning features from DNA, RNA and protein sequences

Jiawei Wang[1], Ruopeng Xie[2,3], Jiahui Li[1,3], André Leier[4,5], Bingjiao Yang[3], Jerico Revote[2], Tatsuya Akutsu[6], Geoffrey I. Webb[7], A. Ian Smith[2,8], Tatiana T. Marquez-Lago[4,5], Yanju Zhang[3], Trevor Lithgow[1,*] and Jiangning Song[2,7,8,*6]

[1]Biomedicine Discovery Institute and Department of Microbiology, Monash University, Melbourne, VIC 3800, Australia

[2]Biomedicine Discovery Institute and Department of Biochemistry and Molecular Biology, Monash University, Melbourne, VIC 3800, Australia

[3]School of Computer Science and Information Security, Guilin University of Electronic Technology, Guangxi, China

[4]Department of Genetics, School of Medicine, University of Alabama at Birmingham, USA

[5]Department of Cell, Developmental and Integrative Biology, School of Medicine, University of Alabama at Birmingham, AL, USA

[6]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Kyoto 611-0011, Japan

[7]Monash Centre for Data Science, Monash University, Melbourne, VIC 3800, Australia

[8]ARC Centre of Excellence in Advanced Molecular Imaging, Monash University, Melbourne, VIC 3800, Australia

* To whom correspondence should be addressed. Jiangning Song: Tel: +61-3-9902-9304; Fax: +61-3-9902-9500; Email: jiangning.song@monash.edu;

Correspondence may also be addressed to Trevor Lithgow: Tel: +61-3-9902-9217; Fax: +61-3-9905-3726; Email: trevor.lithgow@monash.edu.

ABSTRACT

High-throughput sequencing technologies have generated unprecedented amounts of biological sequence data in the post-genomic era. Accordingly, numerous toolkits have been developed based on machine learning techniques for classifying and predicting properties of biological molecules from such sequence data. However, a significant challenge to overcome is the development of appropriate strategies for transforming the raw sequence information into meaningful features that allow for pattern-recognition, prior to feeding these extracted features into machine learning models. To address this challenge, we developed DIFFUSER, a distributed computational framework that enables cost-effective and high-throughput generation of a broad spectrum of heterogeneous features from biological sequences: whether they be DNA, RNA or protein sequences. DIFFUSER distinguishes itself from other existing feature generation toolkits by three key improvements: 1) a novel distributed architecture to improve the online feature generation process using decentralized/parallel computing and distributed storage; 2) a comprehensive feature generation package that can extract a wide range of features, and 3) implementation and availability of both user-friendly web-based server and a unified, cross-platform standalone toolkit with feature customization to cater for different user needs. Here, we outline the architecture of the distributed framework and standalone toolkit of DIFFUSER, demonstrate its improved performance by comparing it to the single server mode, and illustrate how DIFFUSER can benefit machine learning-based analysis and modelling of biological sequences by accelerating the core feature extraction pipeline. The distributed web server and standalone toolkit of DIFFUSER are publicly accessible at http://diffuser.erc.monash.edu/.

INTRODUCTION

Over the last two decades, an unprecedented amount of biological sequence data has been generated due to the wide application of high-throughput sequencing technologies. Ongoing and increasing efforts to sequence whole genomes of organisms will result in continued accumulation of even more biological sequence data in the foreseeable future. For example, the 100,000 Genomes Project by the National Health Service (NHS) of the UK aims at sequencing 100,000 genomes from around 85,000 people. The participants include patients with rare diseases, unusual family lineages, and patients suffering from different cancer types. In accordance with the flourishing growth of these genome-scale datasets, machine learning techniques have been increasingly developed and used to bridge the genotype-phenotype gap and gain insights into the features of biological systems and complex diseases, and in many recent studies, have been considered as a necessary and integral component in cutting-edge systems-level biomedical research (1-3). For various such applications, a common but crucial step is to transform raw sequence data into representative features that encode their underlying relationships in order to train machine learning models. However, feature extraction can pose a challenge for designing successful machine learning models in biology and biomedicine, partly due to the difficulty in formulating the biological sequences as machine learning-compatible vectors and matrices.

To bridge this gap, there have been tremendous efforts made that aim at transforming biological sequences to discrete or numerical vectors that can better capture and encode intrinsic patterns and characteristics of sequences (4-21). Such approaches differ from each other in several key aspects, including the types of sequence extraction algorithms, their implementation, and availability and functionality of web-based and/or standalone toolkits. While the currently available toolkits have individual advantages, they also have certain limitations. These typically include: 1) limited computing power and capacity of generating features, reflected by the limited number of query sequences allowed for each job submission; 2) lack of web server extensibility to facilitate users to customize the feature generation process, i.e. many web servers do not allow for flexible adjustment and do not provide options for numerical parameters, and 3) lack of versatile toolkits that allow the integration of heterogeneous feature types.

To overcome these shortcomings, we designed and developed a distributed framework termed DIFFUSER, for effective generation of a broad spectrum of heterogeneous features from different types of biological sequences. The contribution of DIFFUSER can be

summarized as follows: 1) First, DIFFUSER employs a novel distributed architecture to capitalize on a cluster of computing nodes, instead of relying on a single server, to substantially improve the throughput of feature generation. Compared to a single server, DIFFUSER is able to achieve up to 9-fold acceleration of the feature-generating procedure. Benefitting from the extensibility of the distributed framework, the computing power of DIFFUSER can be readily and flexibly extended in a linear scale, by simply adding more computing nodes with a simplified configuration; 2) Second, DIFFUSER represents the most comprehensive feature generator available to date, covering the largest number of biological sequence features from a broad spectrum, and 3) Third, DIFFUSER provides a user-friendly web server and a standalone toolkit, both of which possess the same functions to generate all types of features in full support of feature customization to meet the needs of different users.

In this paper, we outline the distributed framework of DIFFUSER with a detailed description of its distributed computing and storage strategy. Next, we conduct performance benchmarking experiments to investigate the impact of different numbers of computing nodes on the efficiency of DIFFUSER, and further evaluate its performance against currently available single server-based feature generation toolkits. Finally, we illustrate how users can gain significant benefits from the DIFFUSER web server and standalone toolkit to expedite their research involving machine learning-based sequence analysis and modelling.

MATERIAL AND METHODS

Overview of the DIFFUSER toolkit

The main purpose of developing the DIFFUSER toolkit is to provide a service for generating customizable and heterogeneous features from biological sequences, including DNA, RNA and protein sequences. To this end, we integrated and implemented 116 types of features in five major groups, i.e. sequence-based features, physicochemical property-based features, PSSM-based features, predicted structural features and other profile-based features (we refer to **Table 1** for a statistical summary of the features generated by 18 different toolkits, and **Tables S1-S3** for the descriptive summary of features generated by DIFFUSER). As shown in **Table 1**, DIFFUSER represents the most comprehensive and versatile tool that covers a wider range of heterogeneous features than any other existing feature generation toolkit in terms of the number, type and diversity of features.

To meet the needs of users with different requirements for processing the sequence data and generating features, we developed both a web-based server and a standalone toolkit.

Table 1. Comparison of various types of features generated by DIFFUSER and other currently available toolkits.

Toolkit (Ref.)	Protein					DNA		RNA			Total
	Group1[b]	Group2[b]	Group3[b]	Group4	Group5	Group1	Group2	Group1	Group2	Group4	
PROFEAT[a] (4-6)	1	8	-	-	-	-	-	-	-	-	9
PseAAC (7)	1	2	-	-	-	-	-	-	-	-	3
PseAAC-Builder (8)	1	2	-	-	-	-	-	-	-	-	3
propy (9)	1	10	-	-	-	-	-	-	-	-	11
PseKNC (10)	-	-	-	-	-	-	6	-	2	-	8
PseAAC-General (11)	1	9	1	-	2	-	-	-	-	-	13
repDNA (12)	-	-	-	-	-	3	12	-	-	-	15
PseKNC-General (13)	-	-	-	-	-	1	8	1	5	-	15
Pse-in-One (14)	1	5	-	-	-	2	12	1	5	-	26
protr (15)	1	15	3	-	-	-	-	-	-	-	19
repRNA (16)	-	-	-	-	-	-	-	1	2	**3**	6
PseKRAAC (17)	-	16	-	-	-	-	-	-	-	-	16
Pse-Analysis (18)	-	1	-	-	-	-	1	-	1	-	3
POSSUM (19)	-	-	**21**	-	-	-	-	-	-	-	21
BioSeq-Analysis (20)	3	6	7	2	1	**5**	**15**	3	**8**	**3**	53
iFeature (21)	10	33	1	**7**	-	-	-	-	-	-	51
iLearn (22)	10	33	1	**7**		12	14	**11**	7		95
DIFFUSER	**13**	**38**	**21**	**7**	**3**	**5**	**15**	3	**8**	**3**	**116**

Note: Group1 represents sequence-based features; Group2 represents physicochemical property-based features; Group3 represents PSSM-based features; Group4 represents predicted structural features; Group5 represents other profile-based features. For a feature group, the corresponding toolkit(s) that can generate the maximal number of features is highlighted as bold for comparison purpose.

[a]Only sequence-based features were included for the PROFEAT server;

[b]Feature groups with similar categorization or groupings, such as *k*-mer and its derivatives (e.g. AAC, DPC and TPC) were only counted once.

Unlike some existing toolkits that only provide a standalone version to generate features with no flexible options available for users to choose thereby limiting the practicality of such tools, DIFFUSER provides exactly the same functionality as part of its web server and the standalone toolkit. In addition, it provides support for users to customize the feature generation process to meet their specific needs.

The workflow of the DIFFUSER web server is illustrated in **Figure S1**. Depending on the input sequence type (DNA, RNA or protein), DIFFUSER provides three panels for each sequence type that users can manipulate parameters. Once a panel is selected, users can copy and paste or upload their input sequences in either FASTA format or as raw sequences (**Figure S1A**). Next, users can select some or all optional features, and customize the generation of these features by tuning the corresponding parameters or, alternatively, uploading their own defined files. Once submitted, a unique URL will pop up, allowing users to check the processing status of the submitted job and retrieve the results when the job is accomplished (**Figure S1B**). In addition, users are also permitted to provide a valid email address as an optional input; in such case, once their jobs are completed they will receive an email containing a URL link to the result page. At the result page (**Figure S1C**), users can check the summary information of the job, and download the features in the CSV (i.e. comma-separated values) format together with the intermediate files generated by the server. Moreover, for each type of the generated features, the corresponding command line with all selected parameters is also provided, which can be directly used to execute the standalone toolkit to generate the same feature result file in a local computer (**Figure S1D**). This user-friendly function conveniently provides users with the option to either quick start from the graphical user interface (GUI) or to use the command line for specialized operations.

Distributed framework of the DIFFUSER server

Apart from the advantage of providing easiness for researchers with little programming background, web servers have some intrinsic issues to address in regards to their computing capability, due to the fact that they process all the requests based on their own central computing power. This situation becomes difficult in cases where the users' submitted jobs (such as the PSSM-based feature generation, which is computationally expensive) are time consuming, or in other cases where many job requests are submitted within a short period of time. To deal with this dilemma, most existing feature generation servers set a maximal number of sequences allowed for each job submission (usually less than 500), which eases the servers' computational burden to some extent, but at the same time compromises users' experience and limits their high-throughput application potential.

Considering the increasing requirements for processing large datasets in machine learning applications, especially in the context of big data analytics, it is vital and necessary to develop and

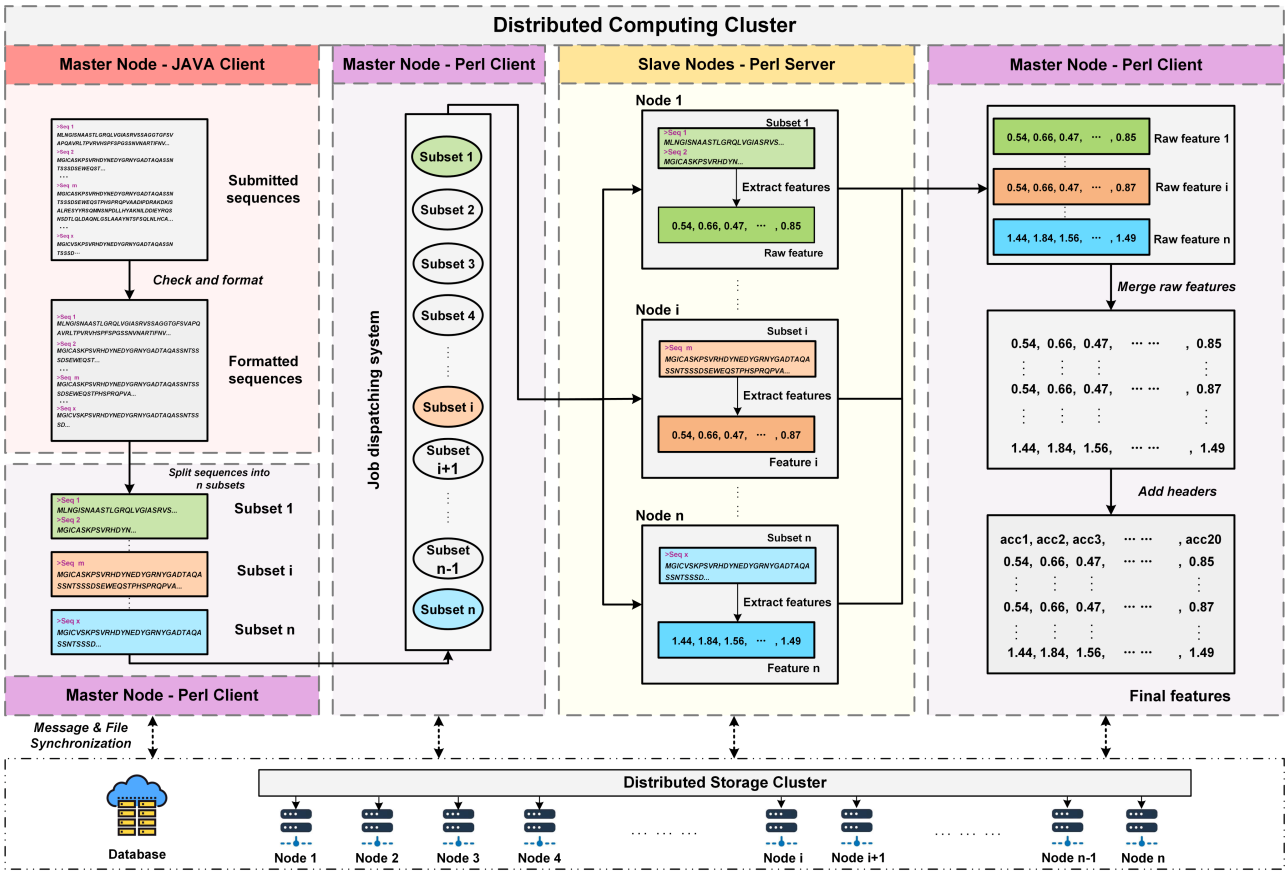


Figure 1. The overall framework of the DIFFUSER web server based on a distributed architecture. Users' requests are handled by DIFFUSER through a distributed computing cluster, consisting of two kinds of nodes: master node and slave node. The master node receives the users' submitted sequences followed by a data formatting procedure, splits them into multiple subsets, and then dispatches these subsets to multiple slave nodes for subsequent processing. The slave node accepts the subset(s) from the task dispatching system in the master node, and executes the feature generation procedure for these subsets and stores the generated features (for each subset) into the distributed file system. After all slave nodes complete the generation of all sub-feature sets for those subsets, the master node will merge all the divided feature segments and generate the final features. During the entire process, a MySQL database is used to synchronize messages, while a distributed file system, FastFile, is used to share files among the master node and slave nodes.

and implement next-generation web servers equipped with extensible computing power for processing large-scale data. To this end, we designed and implemented a distributed framework of DIFFUSER based on distributed/parallel computing and distributed file storage (**Figure 1**). This enables the DIFFUSER web server to process large amounts of biological sequence data at the whole genome scale.

The distributed computing cluster is responsible for parallel executions of the feature generation task, by dividing the task into a number of sub-tasks. Each sub-task will then be executed by a single computer node, and the results of these sub-tasks are merged into a final feature file. Specifically, the distributed computing cluster consists of two types of nodes: a master node and multiple slave nodes. Deployed in the master node, the web server provides users with a graphical and user-friendly interface to submit their query sequences after selecting a few or all of the preferred features and specifying the corresponding parameters. Once the sequences are submitted, they will be passed to the backend of the web server (developed by the JAVA web developing suite) for further check and formatting. The submitted sequences will be then forwarded in a uniform format to a job dispatching system, which is developed based on Gearman (http://gearman.org/) using Perl programming language and deployed across the master node and the slave nodes. The submitted sequences will be split into subsets and then put into the job queue by the master node. The slave nodes with idle threads will proactively fetch jobs from the job queue, and accordingly execute feature generation jobs separately for each individual subset. Once a slave node finishes the feature generation procedure for a subset, it will notify the client of the job dispatching system in the master node. Once all the subtasks are finished, the client of the job dispatching system in the master node is responsible for merging all of these sub-feature sets together into a final feature file.

The distributed file storage is responsible for sharing temporary and final files within the distributed computing cluster. Developed using FastDFS (https://github.com/happyfish100/fastdfs), it has been deployed in all the distributed computing cluster nodes. Each node can directly drop its files during the feature generation process, and those files will be automatically duplicated and shared by other nodes. Additionally, a database, which is developed based on MySQL, is used for recording and sharing the status of sub-jobs. In this way, the nodes within the distributed computing cluster can operate in parallel but manage to work well with each other as an overall distributed system.

Architecture of the DIFFUSER standalone toolkit

In addition to the web server, the standalone toolkit of DIFFUSER is also implemented to enable the users to customize large-scale features on their own by using local computers. This is particularly required if the feature extraction procedure has to be automatically executed or included in a sequence analysis pipeline. Existing toolkits have been developed for this purpose based on different programming languages, including Pse-in-One (14), iFeature (21) and iLearn (22) in Python; protr (15) in R, POSSUM (19) in Perl and Python; and PseAAC-General (11) in C/C++,

which sets an obstacle for users to deal with if they need to generate a group of informative features across a range of candidate features. This situation becomes even more difficult, in cases where users are required to write programming language-specific scripts, for example Python is essential for use of propy (9) and R for protr. To avoid this complexity, the standalone toolkit of DIFFUSER is designed and implemented to cover as many types of features as possible and execute the feature generation in accordance with its web server, in order to provide full support of users' configurations. As a Python-based toolkit, it is straightforward to configure (only a few Python-based libraries are required and can be easily installed by executing several consecutive commands) and user-friendly to operate on different operating systems, such as Unix/Linux, Windows and Mac OS. To generate a type of feature on a given biological sequence dataset, only a line of Shell command is needed, which can also be directly obtained from the DIFFUSER web server (at the result page).

RESULTS

Performance evaluation of DIFFUSER

The effect of the number of nodes used by the distributing computing cluster. Currently, the computing cluster used by DIFFUSER has 10 nodes, each of which is equipped with 16 cores and 64 GB memories. Among those computing nodes, all of them play the role as a slave node, while one of them is also nominated as a master node. To investigate the effect of the number of nodes used by the distributing computing cluster on the computational performance, we compared the computing performance of DIFFUSER in line with the different number of computing nodes. We used the DIFFUSER web server to generate a total of 18 types of representative time-consuming PSSM-based features (with default parameters) with a varying number of computing nodes. For this example, a genome-scale dataset from *Klebsiella pneumoniae* MGH78578 was used as input data. The results indicate that DIFFUSER's computing performance grew almost linearly in accordance with the increase of the number of its computing nodes (**Figure 2A**). On one hand, this observation clearly demonstrates the superiority of the distributed framework over its single node mode. On the other hand, the linear scalability of the DIFFUSER means that its computing performance can be augmented simply by adding more computing nodes to meet further demands in the big data era. It is also notable that, compared to the single node mode, the distributed framework with 10 nodes achieved a 5.4-time (less than 9-time) improvement. Possible reasons include: (1) the varying processing time for different sequences may affect the parallelizing effect; (2) The computing abilities are influenced by other nodes within the same cluster environment, even that they have

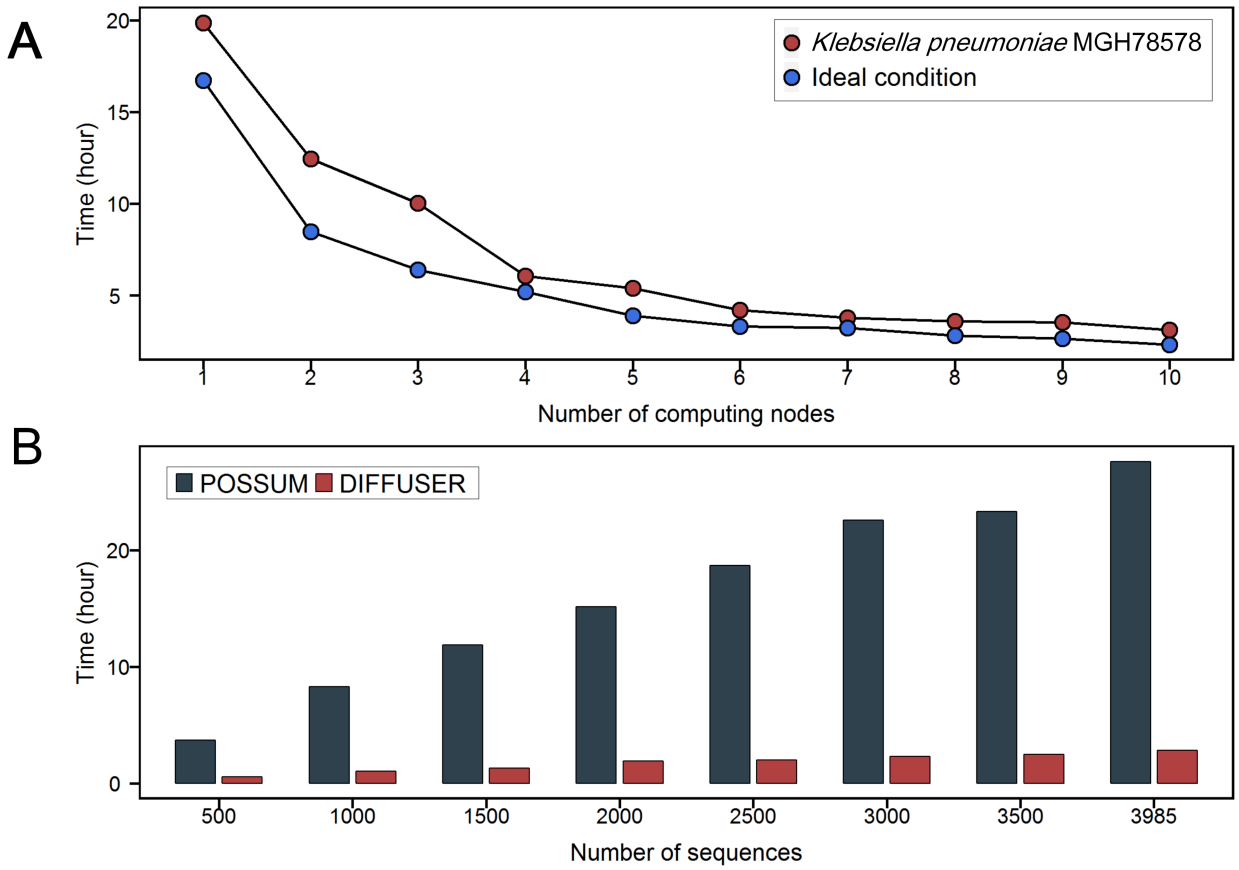


Figure 2. Computational performance comparison. (A) The effect of cluster nodes used by DIFFUSER on its computing performance, (B) Performance comparison between DIFFUSER and POSSUM in terms of the computing time. Both groups of experiments were conducted based on a genome-scale dataset from *Klebsiella pneumoniae* MGH78578, which was pre-processed to remove unexpectedly time-consuming sequences (**Figure S2**). Experiments in (A) were additionally conducted based on a simulated dataset that contains 3985 identical sequences as contrast under ideal condition. Sub-nodes and the single node involved in all experiments run 16 threads to concurrently process the sub-tasks.

been equally configured at the level of both hardware and software. The former normally happens in practical scenarios, which can be further validated by a 6-time improvement in additional experiments under a relatively ideal condition where all processed sequences are identical and thus their feature generation process consume equal time. The latter was further validated in next section by multiple times experiments to reduce the influence brought by stochastic outcomes.

Performance comparison between DIFFUSER and existing feature generation toolkits. In order to benchmark the performance between the DIFFUSER web server and existing toolkits, we selected POSSUM (19) as the baseline toolkit based on two considerations: (1) As a specialized toolkit, POSSUM is able to generate representative time-consuming PSSM-based features, and (2) the host server of POSSUM is relatively powerful and has the same configuration as a single node

of DIFFUSER's computing cluster. These features make it a reasonable comparison between DIFFUSER and POSSUM servers.

Based on the above dataset, the same numbers of features were generated using DIFFUSER and POSSUM web servers with default parameters. We repeated this operation using different subsets, each of which was randomly selected from the original dataset, starting with 500 sequences, and in increments of 500. The performance comparison results between DIFFUSER and POSSUM are summarized in **Figure 2B**. Since the time consumed on feature generation for a sequence is roughly the same, the total time consumed by both DIFFUSER and POSSUM grows linearly in accordance with the increase of the size of the submitted datasets (i.e. number of submitted sequences). However, for each submission, the consumed time by DIFFUSER ranged from 5.7 to 8.8-fold shorter than that by POSSUM. This observation shows more stable improved performance when increasing the scale of experiments. It is also notable that the DIFFUSER outperformed the POSSUM with more than 8-times improvement based on the subsets with more than 2000 sequences. This indicates that additional time consuming might be incurred by the processing procedures (e.g. file splitting & merging) on the master node and the communications and synchronizations among nodes, but as the number of sequences grows, the feature generation time on each node will grow much faster than the extra overhead. In this case, the relatively shorter time consumed on additional operations within the distributed architecture, will exert less influence on the overall performance.

Applications of DIFFUSER to two practical scenarios

As a key and indispensable step involved in machine learning-based analysis and modelling, feature extraction is time-consuming and complicated, and is often inundated with mathematical formulations (14). By automatically generating various types of features, existing toolkits can greatly facilitate the efforts and speed up machine learning-based studies in biology. Complementing currently available toolkits, DIFFUSER facilitates feature generation especially from large-scale sequence datasets, by providing a service to generate a great variety of heterogeneous features based on a distributed framework. In this section, we apply DIFFUSER to two real-world machine learning based sequence analysis scenarios and illustrate how it can better serve user demands in these studies.

Bacterial secreted effector protein prediction. As secreted effectors play an important role in bacterium–host interactions or inter-bacteria competitions, the in-silico identification of such proteins is fundamental to an understanding of their functions and roles in the pathogenic process

(23,24). A consensus of previous studies is that features extracted from various aspects could have an important influence on the predictor's performance and it remains a challenge to develop more accurate predictors for the prediction of different types of bacterial secreted effector proteins (25-28). Recent research (26) that focused on type VI effector prediction extracted nine features across three major groups (i.e. sequence-based features, evolutionary information-based features and physicochemical features) which collectively contributed to the predictor's performance. In order to generate such features, two toolkits, namely POSSUM (19) and protr (15), were used in this study. Furthermore, to construct the predictor, users need to install and configure the standalone toolkits of POSSUM and protr, configure Perl, Python and R runtime environments, and require proper skills of R to execute some necessary scripts using protr (developed using R). In addition, to perform genome-scale prediction of potential bacterial secreted effectors for a bacterial species (~3,000-5,000 sequences), the features should be first generated. For example, in the case of evolutionary information-based features, users need to submit 10 times to the POSSUM web server, each time with a maximum number of 500 sequences, and will wait for several days before they can get all results. With the availability of the standalone DIFFUSER toolkit, users can now run and complete these mentioned tasks more easily using local computers as only Python is required (Python is pre-installed for Linux or Mac OS systems). In addition, users can alternatively use the DIFFUSER web server (with 10 computing nodes) by only submitting one task (5,000 sequences permitted per submission) for a bacterial genome, and obtain the prediction results after several hours. In this way, DIFFUSER is able to substantially expedite the feature generation process, and enhance user's experience by significantly reducing the waiting time.

Prediction of post-translational modification (PTM). PTMs play important roles in the regulation of diverse cell functions and are often associated with various diseases (29,30). As an alternative approach for identifying PTM sites, computational methods can complement with experimental studies by narrowing down the PTM substrates for experimental validation. Among different types of PTMs, lysine malonylation is a recently discovered PTM and has been shown to be closely associated with the regulation of metabolic pathways, particularly the pathways of glucose and fatty acid metabolism. In a recent study on the prediction of lysine malonylation (31), a total of 11 features were extracted and investigated. At least three toolkits are required to generate these features. These include the following types of features: 1) 1-gram and 2-gram (instantiations of *k*-mer when *k* is set to 1 and 2, respectively) that can be generated by most feature generation toolkits; 2) QSO and AAINDEX features, which can be generated by protr or iFeature (21); 3) BINA and KNN features, which can be only generated by iFeature; 4) PSSM and S_FPSSM, which were

exclusively generated by POSSUM. The remaining features, including NUM, LOGO and EBGW, will need to be manually generated by writing computer scripts due to the unavailability of the corresponding toolkits. In real applications, it would be a long way to go for users to obtain all required features before training the machine learning-based models for lysine malonylation prediction. Nevertheless, with the availability of DIFFUER that covers all the major feature groups, all features can be generated using its online web server by unlocking the corresponding toggles, or the local standalone command line tool.

DISCUSSION

In this work we developed DIFFUSER, a bioinformatics toolkit for generating a broad and comprehensive range of heterogeneous features from biological sequences. DIFFUSER has been implemented as both an online web server and a local stand-alone software program. Unlike most other existing toolkits that only consider one or some groups of features, DIFFUSER has covered the largest number of feature categories, to facilitate seamless operations on extracting and generating features for machine learning applications.

To enable high-throughput feature generation, we designed and implemented a distributed framework of the DIFFUSER online server to enhance its processing capability based on distributed/parallel computing and distributed storage. This distributed framework connects and coordinates single computing nodes to enable them to operate as an integrated system. As a result, this has significantly improved the computing performance, compared to the single server based toolkits. Moreover, the computing power of this distributed framework can be readily extended by adding more numbers of computing nodes with a simple configuration.

Both web server and standalone toolkit have the same functionality to generate all types of features in full support for feature customization (including parameter adjustment and self-defined file definition). Specially, features generated by the online web server will be provided together with their corresponding commands, which can be also directly executed locally to generate the same features. In this way, DIFFUSER offers an easy and convenient way for users to switch from using the online web server to the standalone toolkit.

In summary, we present a bioinformatics toolkit based on a distributed computing framework that can be effectively used to generate a great variety of heterogeneous features from biological sequence data in a high-throughput manner. It also allows automatic feature extraction and provides opportunities to be further integrated into downstream computational pipelines in the future. We have also applied DIFFUSER to two real-world application scenarios and demonstrated its

practicality and value to greatly facilitate feature generation efforts. It is expected that DIFFUSER will be a useful tool for accelerating machine learning-based research in biology and medicine.

AVAILABILITY

http://diffuser.erc.monash.edu/.

ACKNOWLEDGEMENT

The authors would like to thank A/Prof. David Powell, A/Prof. Traude Beilharz and A/Prof. Vijay Dhanasekaran for their critical comments and fruitful discussions.

FUNDING

This work was financially supported by grants from the National Health and Medical Research Council of Australia (NHMRC) (1092262, 1144652 and 1127948), the Australian Research Council (ARC) (LP110200333 and DP120104460), the National Institute of Allergy and Infectious Diseases of the National Institutes of Health (R01 AI111965), and the Collaborative Research Program of Institute for Chemical Research, the National Natural Science Foundation of China (61862017), and the Natural Science Foundation of Guangxi (2018GXNSFAA138117, 2016GXNSFCA380005). TML and AL's work was supported in part by the Informatics Institute of the School of Medicine at UAB. T.L. is an ARC Australian Laureate Fellow (FL130100038).

CONFLICT OF INTEREST

None declared.

References

1. Camacho, D.M., Collins, K.M., Powers, R.K., Costello, J.C. and Collins, J.J. (2018) Next-Generation Machine Learning for Biological Networks. *Cell*, **173**, 1581-1592.
2. Larranaga, P., Calvo, B., Santana, R., Bielza, C., Galdiano, J., Inza, I., Lozano, J.A., Armananzas, R., Santafe, G., Perez, A. *et al.* (2006) Machine learning in bioinformatics. *Briefings in bioinformatics*, **7**, 86-112.
3. Libbrecht, M.W. and Noble, W.S. (2015) Machine learning applications in genetics and genomics. *Nature reviews. Genetics*, **16**, 321-332.
4. Li, Z.R., Lin, H.H., Han, L.Y., Jiang, L., Chen, X. and Chen, Y.Z. (2006) PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic acids research*, **34**, W32-37.
5. Rao, H.B., Zhu, F., Yang, G.B., Li, Z.R. and Chen, Y.Z. (2011) Update of PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucleic acids research*, **39**, W385-390.
6. Zhang, P., Tao, L., Zeng, X., Qin, C., Chen, S.Y., Zhu, F., Yang, S.Y., Li, Z.R., Chen, W.P. and Chen, Y.Z. (2017) PROFEAT Update: A Protein Features Web Server with Added Facility to Compute Network Descriptors for Studying Omics-Derived Networks. *Journal of molecular biology*, **429**, 416-425.
7. Shen, H.B. and Chou, K.C. (2008) PseAAC: a flexible web server for generating various kinds of protein pseudo amino acid composition. *Analytical biochemistry*, **373**, 386-388.
8. Du, P., Wang, X., Xu, C. and Gao, Y. (2012) PseAAC-Builder: a cross-platform stand-alone program for generating various special Chou's pseudo-amino acid compositions. *Analytical biochemistry*, **425**, 117-119.
9. Cao, D.S., Xu, Q.S. and Liang, Y.Z. (2013) propy: a tool to generate various modes of Chou's PseAAC. *Bioinformatics*, **29**, 960-962.
10. Chen, W., Lei, T.Y., Jin, D.C., Lin, H. and Chou, K.C. (2014) PseKNC: a flexible web server for generating pseudo K-tuple nucleotide composition. *Analytical biochemistry*, **456**, 53-60.
11. Du, P., Gu, S. and Jiao, Y. (2014) PseAAC-General: fast building various modes of general form of Chou's pseudo-amino acid composition for large-scale protein datasets. *International journal of molecular sciences*, **15**, 3495-3506.
12. Liu, B., Liu, F., Fang, L., Wang, X. and Chou, K.C. (2015) repDNA: a Python package to generate various modes of feature vectors for DNA sequences by incorporating user-defined physicochemical properties and sequence-order effects. *Bioinformatics*, **31**, 1307-1309.
13. Chen, W., Zhang, X., Brooker, J., Lin, H., Zhang, L. and Chou, K.C. (2015) PseKNC-General: a cross-platform package for generating various modes of pseudo nucleotide compositions. *Bioinformatics*, **31**, 119-120.
14. Liu, B., Liu, F., Wang, X., Chen, J., Fang, L. and Chou, K.C. (2015) Pse-in-One: a web server for generating various modes of pseudo components of DNA, RNA, and protein sequences. *Nucleic acids research*, **43**, W65-71.
15. Xiao, N., Cao, D.S., Zhu, M.F. and Xu, Q.S. (2015) protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences. *Bioinformatics*, **31**, 1857-1859.
16. Liu, B., Liu, F., Fang, L., Wang, X. and Chou, K.C. (2016) repRNA: a web server for generating various feature vectors of RNA sequences. *Molecular genetics and genomics : MGG*, **291**, 473-481.

17. Zuo, Y., Li, Y., Chen, Y., Li, G., Yan, Z. and Yang, L. (2017) PseKRAAC: a flexible web server for generating pseudo K-tuple reduced amino acids composition. *Bioinformatics*, **33**, 122-124.
18. Liu, B., Wu, H., Zhang, D., Wang, X. and Chou, K.C. (2017) Pse-Analysis: a python package for DNA/RNA and protein/ peptide sequence analysis based on pseudo components and kernel methods. *Oncotarget*, **8**, 13338-13343.
19. Wang, J., Yang, B., Revote, J., Leier, A., Marquez-Lago, T.T., Webb, G., Song, J., Chou, K.C. and Lithgow, T. (2017) POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. *Bioinformatics*, **33**, 2756-2758.
20. Liu, B. (2017) BioSeq-Analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches. *Briefings in bioinformatics*.
21. Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T.T., Wang, Y., Webb, G.I., Smith, A.I., Daly, R.J., Chou, K.C. *et al.* (2018) iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics*, **34**, 2499-2502.
22. Chen, Z., Zhao, P., Li, F., Marquez-Lago, T.T., Leier, A., Revote, J., Zhu, Y., Powell, D.R., Akutsu, T., Webb, G.I. *et al.* (2019) iLearn: an integrated platform and meta-learner for feature engineering, machine-learning analysis and modeling of DNA, RNA and protein sequence data. *Briefings in bioinformatics*.
23. An, Y., Wang, J., Li, C., Leier, A., Marquez-Lago, T., Wilksch, J., Zhang, Y., Webb, G.I., Song, J. and Lithgow, T. (2018) Comprehensive assessment and performance improvement of effector protein predictors for bacterial secretion systems III, IV and VI. *Briefings in bioinformatics*, **19**, 148-161.
24. Zeng, C. and Zou, L. (2017) An account of in silico identification tools of secreted effector proteins in bacteria and future challenges. *Briefings in bioinformatics*.
25. Wang, J., Yang, B., An, Y., Marquez-Lago, T., Leier, A., Wilksch, J., Hong, Q., Zhang, Y., Hayashida, M., Akutsu, T. *et al.* (2019) Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches. *Briefings in bioinformatics*, **20**, 931-951.
26. Wang, J., Yang, B., Leier, A., Marquez-Lago, T.T., Hayashida, M., Rocker, A., Zhang, Y., Akutsu, T., Chou, K.C., Strugnell, R.A. *et al.* (2018) Bastion6: a bioinformatics approach for accurate prediction of type VI secreted effectors. *Bioinformatics*, **34**, 2546-2555.
27. Wang, J., Li, J., Yang, B., Xie, R., Marquez-Lago, T.T., Leier, A., Hayashida, M., Akutsu, T., Zhang, Y., Chou, K.C. *et al.* (2019) Bastion3: a two-layer ensemble predictor of type III secreted effectors. *Bioinformatics*, **35**, 2017-2028.
28. Arnold, R., Brandmaier, S., Kleine, F., Tischler, P., Heinz, E., Behrens, S., Niinikoski, A., Mewes, H.W., Horn, M. and Rattei, T. (2009) Sequence-based prediction of type III secreted proteins. *PLoS pathogens*, **5**, e1000376.
29. Gallego, M. and Virshup, D.M. (2007) Post-translational modifications regulate the ticking of the circadian clock. *Nature reviews. Molecular cell biology*, **8**, 139-148.
30. Westermann, S. and Weber, K. (2003) Post-translational modifications regulate microtubule function. *Nature reviews. Molecular cell biology*, **4**, 938-947.
31. Zhang, Y., Xie, R., Wang, J., Leier, A., Marquez-Lago, T.T., Akutsu, T., Webb, G.I., Chou, K.C. and Song, J. (2018) Computational analysis and prediction of lysine malonylation sites by exploiting informative features in an integrative machine-learning framework. *Briefings in bioinformatics*.

CHAPTER 5: Conclusion

Responsible for many infectious diseases and deaths all over the world, Gram-negative bacteria have evolved a wide range of highly diverse secretion systems as their weapons to export substrates into either environments or host cell cytoplasm to modulate the interactions within their environments, target hosts or competitors. It is a common consensus that these secreted substrates play vital roles in competitive survival, host cell subversion and pathogenesis, and bacterial competitor elimination. Experimental validation of secreted substrates is a fundamental step that is required to uncover their structural and biochemical properties, as well as their function roles, which will further promote our understanding of the survival strategies and pathogenesis mechanisms of the bacteria. Despite the importance of the bacterial secreted substrates, how these specialized proteins are targeted for secretion by their corresponding secretion systems is still not fully understood.

Contrasting the time-consuming and labour-intensive experimental strategies with the rapid advance of high-throughput sequencing techniques, the gap between the number of the known bacterial secreted substrates and sequence-known proteins is growing ever wider. This imbalance could be significantly reduced through the introduction of high-throughput and accurate computational analysis and prediction methodologies. For this purpose, this thesis has made three contributions.

Three ensemble substrate predictors with demonstrated improved performance. Through systematically analyzing the known substrates, new methodologies have been developed to predict three well studied types of substrates. The methods I employed explore a wide spectrum of heterogeneous features that are extracted from different aspects, train models to mine patterns from them using different machine learning algorithms, and integrate these models as the final ensemble models based on multiple ensemble learning strategies. The proposed predictors have been demonstrated by extensive benchmarking tests to outperform existing state-of-the-art predictive toolkits, and are now providing public service to facilitate discoveries of new secreted substrates.

An integrative platform for annotation, analysis and prediction of secreted substrates. To comprehensively and systematically annotate, analyze and predict the repertoire of substrates secreted by Gram-negative bacteria, a universal platform has been developed to provide an all-in-one service to ensure there is a seamless transition between the prediction and analysis procedure. To achieve this, an integrative prediction system has been developed

to comprehensively and accurately predict each type of secreted substrates from genome-scale sequences within a distributed framework. It is further expanded and integrated into a new developed platform, which integrates various types of experimentally validated and annotated substrates and provides a range of analytic modules. Together these form the final one-stop platform to enable users to investigate known substrates, predict potential substrates, and analyze relationships between the two.

Two versatile toolkits for generating machine learning features. To speed up machine learning based modelling and analysis, and improve the performance of the constructed predictors, two versatile toolkits have been developed. This allows users to easily generate a great variety of heterogeneous features from their biological sequence. The distributed web architecture guarantees the practical application to genome-scale data, while their standalone toolkits offer opportunities to execute analysis locally whilst still being possible to be integrated into downstream computational pipelines. This allows automatic feature extraction or machine learning based sequence analysis and modelling thereafter. My work in secreted substrate prediction is a direct real-world application scenario that demonstrates how both feature generating toolkits could benefit machine learning-based analysis and modelling of biological sequences through accelerating the core feature extraction procedure.

The proposed predictors together with the universal platform is anticipated to expedite the overall understanding of known substrates, the discovery of putative substrates, and the computational analysis and comparison between known and potential substrates. Apart from providing specialized bacterial substrate predictors, the proposed computational frameworks, including feature extraction, analysis and visualization, model training and parameter optimization, ensemble model construction strategies, and the distributed architecture design may inspire researchers to design and develop novel computational methods in a broader context in the field of biomedicine, bioinformatics and computational biology.

In the future, as the number of experimentally validated substrates expands, our proposed models could be further updated: new substrates with novel characteristics need to be characterized by new and additional informative features, which are required be learned and interpreted by new or more suitable machine learning algorithms. If the datasets grow large enough, deep learning techniques could be introduced to directly learn the patterns and characteristics without manual feature engineering (Angermueller et al., 2016; Ekins, 2016; Esteva et al., 2019; Gawehn et al., 2016; Min et al., 2017; Park & Kellis, 2015; Wainberg et al., 2018; Zou et al., 2019). It would be of particular interest to compare effects of both deep

learning and classical machine learning algorithms on the substrate prediction and further investigate a proper way to integrate them for more accurate and robust prediction by taking advantages of their strengths and merits.

In addition to substrate analysis and prediction, more unified platform is promising to be designed and implemented in a scalable software architecture to allow to install or uninstall plugins for new functional extensions. In this way, it can be easily upgraded and continuously updated with increasing functionalities, such as virulence factor detection, subcellular localization prediction, bacterial function annotation, antibiotic resistance prediction and drug resistance prediction. Finally, it is expected to evolve as an enhanced, universal and automatic pipeline which could annotate bacteria genomes with various inferred functions and attributes, pinpoint their potential pathogenic or antibiotic resistant genes, or recommend possible drug therapy strategies. Hopefully, one day we can “diagnose” the capabilities of bacteria in the same way as genome technologies are being applied in human diagnosis and treatment in this era of medical big data and precision medicine.

References

Angermueller, C., Parnamaa, T., Parts, L., & Stegle, O. (2016). Deep learning for computational biology. *Mol Syst Biol, 12*(7), 878. doi:10.15252/msb.20156651

Ekins, S. (2016). The Next Era: Deep Learning in Pharmaceutical Research. *Pharm Res, 33*(11), 2594-2603. doi:10.1007/s11095-016-2029-7

Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., & Dean, J. (2019). A guide to deep learning in healthcare. *Nat Med, 25*(1), 24-29. doi:10.1038/s41591-018-0316-z

Gawehn, E., Hiss, J. A., & Schneider, G. (2016). Deep Learning in Drug Discovery. *Mol Inform, 35*(1), 3-14. doi:10.1002/minf.201501008

Min, S., Lee, B., & Yoon, S. (2017). Deep learning in bioinformatics. *Brief Bioinform, 18*(5), 851-869. doi:10.1093/bib/bbw068

Park, Y., & Kellis, M. (2015). Deep learning for regulatory genomics. *Nat Biotechnol, 33*(8), 825-826. doi:10.1038/nbt.3313

Wainberg, M., Merico, D., Delong, A., & Frey, B. J. (2018). Deep learning in biomedicine. *Nat Biotechnol, 36*(9), 829-838. doi:10.1038/nbt.4233

Zou, J., Huss, M., Abid, A., Mohammadi, P., Torkamani, A., & Telenti, A. (2019). A primer on deep learning in genomics. *Nat Genet, 51*(1), 12-18. doi:10.1038/s41588-018-0295-5

Appendix

Appendix 1 - Supplementary information for Chapter 2.1

Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches

Supplementary information

Table S1. Performance of various classifiers based on 5-fold cross-validation tests. Values are presented as mean±standard error. Except for AAC (20D) and PPT (72D), all feature vectors are 200-dimensional and were selected using GainRatio. In the Table, Group 1 denotes the amino acid composition group (AAC, DPC, CKSAAP and PPT); Group 2 denotes the PSSM group (PSSM, PSSM_AC and PSSM_SMTH); Group 3 denotes the structure group (SA, SS and DISO), while All features include all the 10 feature types and are used as a whole group. For each group, corresponding features were combined into a vector to train the model.

Feature	Method	PRE	SN	SP	F-score	ACC	MCC
AAC	RF	0.836±0.009	0.825±0.005	0.839±0.012	0.829±0.006	0.831±0.007	0.663±0.014
	SVM	0.856±0.007	**0.845±0.011**	0.859±0.009	**0.849±0.007**	**0.851±0.007**	**0.703±0.014**
	LR	0.816±0.006	0.834±0.005	0.813±0.007	0.824±0.003	0.823±0.004	0.647±0.009
	NB	0.792±0.005	0.837±0.004	0.782±0.005	0.813±0.004	0.809±0.003	0.619±0.007
	KNN	0.827±0.005	0.838±0.009	0.826±0.006	0.831±0.005	0.831±0.003	0.664±0.008
	MLP	**0.864±0.010**	0.727±0.008	**0.886±0.011**	0.788±0.007	0.805±0.007	0.620±0.013
PPT	RF	0.816±0.006	0.816±0.014	0.817±0.005	0.815±0.010	0.816±0.008	0.633±0.017
	SVM	0.818±0.009	**0.828±0.007**	0.817±0.011	**0.822±0.005**	**0.822±0.005**	**0.645±0.010**
	LR	0.803±0.007	0.788±0.003	0.808±0.008	0.794±0.004	0.797±0.004	0.596±0.008
	NB	0.715±0.006	0.348±0.003	0.860±0.004	0.464±0.004	0.603±0.002	0.243±0.007
	KNN	0.808±0.008	0.745±0.008	0.824±0.008	0.773±0.010	0.783±0.009	0.570±0.016
	MLP	**0.843±0.016**	0.689±0.035	**0.872±0.019**	0.755±0.020	0.779±0.014	0.571±0.027
DPC	RF	0.811±0.015	0.810±0.006	0.812±0.017	0.809±0.010	0.810±0.011	0.621±0.023
	SVM	**0.837±0.007**	0.805±0.010	**0.844±0.010**	0.819±0.005	**0.823±0.004**	**0.648±0.007**
	LR	0.812±0.003	0.839±0.005	0.806±0.002	**0.824±0.003**	0.822±0.002	0.645±0.005
	NB	0.793±0.002	**0.840±0.003**	0.782±0.004	0.815±0.002	0.811±0.003	0.623±0.006
	KNN	0.797±0.004	0.820±0.006	0.793±0.003	0.807±0.005	0.806±0.003	0.612±0.006
	MLP	0.813±0.015	0.681±0.012	0.843±0.014	0.739±0.013	0.761±0.012	0.531±0.023

CKSAAP	RF	0.840±0.004	0.813±0.009	0.846±0.005	**0.825±0.003**	**0.829±0.002**	**0.659±0.006**
	SVM	**0.877±0.005**	0.726±0.009	**0.900±0.006**	0.793±0.004	0.812±0.002	0.635±0.005
	LR	0.737±0.009	0.742±0.012	0.736±0.012	0.738±0.009	0.738±0.009	0.477±0.018
	NB	0.819±0.003	0.831±0.004	0.817±0.004	0.824±0.003	0.823±0.003	0.648±0.006
	KNN	0.763±0.008	**0.860±0.007**	0.732±0.010	0.808±0.006	0.796±0.006	0.598±0.011
	MLP	0.831±0.008	0.733±0.006	0.852±0.007	0.779±0.005	0.792±0.005	0.589±0.010
PSSM	RF	0.909±0.004	0.900±0.005	0.911±0.003	0.904±0.004	**0.905±0.003**	**0.811±0.007**
	SVM	**0.933±0.001**	0.861±0.008	**0.939±0.003**	0.895±0.004	0.900±0.003	0.803±0.006
	LR	0.808±0.007	0.851±0.016	0.797±0.011	0.828±0.008	0.824±0.006	0.649±0.012
	NB	0.888±0.004	0.887±0.003	0.889±0.003	0.887±0.004	0.888±0.003	0.776±0.006
	KNN	0.899±0.003	**0.911±0.003**	0.898±0.003	**0.904±0.003**	0.904±0.003	0.809±0.005
	MLP	0.935±0.013	0.859±0.010	0.943±0.010	0.895±0.009	0.902±0.008	0.806±0.016
PSSM_AC	RF	**0.906±0.006**	0.771±0.009	**0.921±0.005**	**0.832±0.007**	**0.846±0.006**	**0.699±0.012**
	SVM	0.897±0.012	0.765±0.022	0.914±0.012	0.825±0.015	0.839±0.012	0.686±0.022
	LR	0.720±0.011	0.757±0.012	0.705±0.015	0.736±0.008	0.730±0.008	0.463±0.017
	NB	0.610±0.001	**0.867±0.003**	0.447±0.003	0.715±0.002	0.656±0.002	0.346±0.006
	KNN	0.833±0.004	0.816±0.004	0.836±0.004	0.823±0.002	0.825±0.002	0.652±0.006
	MLP	0.896±0.021	0.690±0.009	0.921±0.018	0.777±0.007	0.805±0.007	0.628±0.018
PSSM_SMTH	RF	0.859±0.006	0.825±0.007	0.865±0.006	**0.840±0.005**	**0.844±0.005**	**0.691±0.011**
	SVM	0.873±0.007	0.790±0.014	0.886±0.004	0.828±0.010	0.837±0.008	0.679±0.017
	LR	0.733±0.017	0.734±0.014	0.730±0.026	0.732±0.008	0.732±0.011	0.466±0.020
	NB	0.658±0.003	**0.870±0.002**	0.548±0.006	0.748±0.001	0.708±0.002	0.441±0.006
	KNN	0.804±0.004	0.784±0.005	0.809±0.007	0.793±0.003	0.796±0.005	0.594±0.010
	MLP	**0.886±0.016**	0.756±0.022	**0.909±0.013**	0.815±0.018	0.835±0.016	0.675±0.030
DISO	RF	0.714±0.011	0.733±0.015	0.708±0.011	0.722±0.012	0.719±0.011	0.441±0.022
	SVM	**0.736±0.016**	0.726±0.020	0.739±0.020	**0.728±0.015**	**0.732±0.014**	**0.466±0.027**
	LR	0.604±0.008	0.607±0.018	0.602±0.020	0.603±0.009	0.603±0.007	0.209±0.016

	NB	0.631±0.026	0.657±0.033	0.625±0.009	0.637±0.033	0.640±0.016	0.283±0.037
	KNN	0.695±0.005	**0.746±0.008**	0.674±0.010	0.718±0.004	0.709±0.004	0.422±0.006
	MLP	0.733±0.016	0.570±0.032	**0.791±0.016**	0.639±0.022	0.680±0.014	0.371±0.028
SA	RF	0.611±0.005	0.642±0.010	0.590±0.005	0.623±0.008	**0.613±0.006**	**0.232±0.013**
	SVM	0.604±0.010	0.606±0.022	0.600±0.022	0.601±0.013	0.600±0.010	0.206±0.018
	LR	0.585±0.014	0.591±0.015	0.581±0.016	0.585±0.012	0.583±0.012	0.172±0.026
	NB	0.543±0.006	**0.911±0.011**	0.207±0.007	**0.672±0.006**	0.560±0.007	0.179±0.015
	KNN	**0.633±0.014**	0.498±0.007	**0.711±0.019**	0.555±0.008	0.603±0.010	0.214±0.020
	MLP	0.576±0.019	0.449±0.036	0.671±0.017	0.502±0.030	0.560±0.014	0.123±0.032
SS	RF	0.560±0.022	0.535±0.030	0.579±0.016	0.544±0.025	0.555±0.022	**0.115±0.046**
	SVM	**0.562±0.021**	0.463±0.043	0.634±0.023	0.492±0.034	0.540±0.021	0.102±0.037
	LR	0.536±0.017	0.542±0.022	0.531±0.018	0.537±0.019	0.536±0.018	0.073±0.037
	NB	0.543±0.007	**0.673±0.018**	0.432±0.010	**0.597±0.012**	**0.555±0.007**	0.111±0.015
	KNN	0.530±0.017	0.493±0.018	0.564±0.020	0.505±0.017	0.524±0.016	0.057±0.032
	MLP	0.535±0.024	0.361±0.029	**0.688±0.032**	0.428±0.025	0.525±0.018	0.052±0.037
Group 1	RF	0.823±0.008	0.800±0.007	0.829±0.008	0.810±0.007	**0.813±0.006**	**0.629±0.012**
	SVM	**0.829±0.008**	0.758±0.008	**0.844±0.009**	0.790±0.006	0.800±0.005	0.604±0.009
	LR	0.797±0.013	0.805±0.016	0.796±0.015	0.799±0.013	0.799±0.012	0.601±0.023
	NB	0.800±0.004	**0.827±0.005**	0.795±0.004	**0.812±0.003**	0.810±0.002	0.622±0.006
	KNN	0.788±0.005	0.809±0.007	0.783±0.007	0.797±0.004	0.795±0.004	0.591±0.009
	MLP	0.802±0.016	0.702±0.023	0.825±0.014	0.746±0.017	0.762±0.015	0.531±0.029
Group 2	RF	0.912±0.003	**0.900±0.003**	0.914±0.003	**0.905±0.003**	**0.906±0.002**	**0.813±0.005**
	SVM	**0.940±0.003**	0.862±0.007	**0.945±0.004**	0.899±0.003	0.904±0.002	0.810±0.004
	LR	0.801±0.018	0.859±0.008	0.787±0.026	0.828±0.010	0.823±0.012	0.647±0.024
	NB	0.887±0.003	0.896±0.004	0.887±0.003	0.891±0.003	0.891±0.002	0.782±0.005
	KNN	0.927±0.003	0.881±0.004	0.932±0.004	0.903±0.004	0.906±0.003	0.813±0.007
	MLP	0.932±0.008	0.854±0.009	0.939±0.009	0.891±0.002	0.896±0.002	0.796±0.003
Group 3	RF	0.715±0.009	0.733±0.017	0.708±0.009	0.722±0.012	0.720±0.010	0.441±0.020

	SVM	0.736±0.016	0.726±0.020	0.739±0.020	0.728±0.015	**0.732±0.014**	**0.466±0.027**
	LR	0.602±0.010	0.607±0.014	0.599±0.019	0.602±0.009	0.602±0.010	0.206±0.022
	NB	0.631±0.026	0.657±0.033	0.625±0.009	0.637±0.033	0.640±0.016	0.283±0.037
	KNN	0.704±0.008	**0.760±0.005**	0.682±0.011	**0.729±0.005**	0.720±0.006	0.443±0.013
	MLP	**0.737±0.020**	0.581±0.018	**0.791±0.023**	0.647±0.013	0.685±0.012	0.381±0.024
All features	RF	0.815±0.004	0.781±0.008	0.824±0.005	**0.796±0.005**	**0.801±0.005**	0.605±0.010
	SVM	0.856±0.009	0.727±0.016	0.878±0.007	0.785±0.012	0.801±0.009	**0.612±0.018**
	LR	0.686±0.017	0.697±0.018	0.681±0.019	0.690±0.017	0.688±0.018	0.378±0.036
	NB	0.657±0.003	**0.870±0.002**	0.545±0.008	0.747±0.002	0.707±0.003	0.439±0.007
	KNN	0.816±0.003	0.763±0.003	0.828±0.004	0.787±0.004	0.794±0.003	0.591±0.007
	MLP	**0.871±0.011**	0.669±0.045	**0.907±0.011**	0.754±0.030	0.793±0.022	0.599±0.039

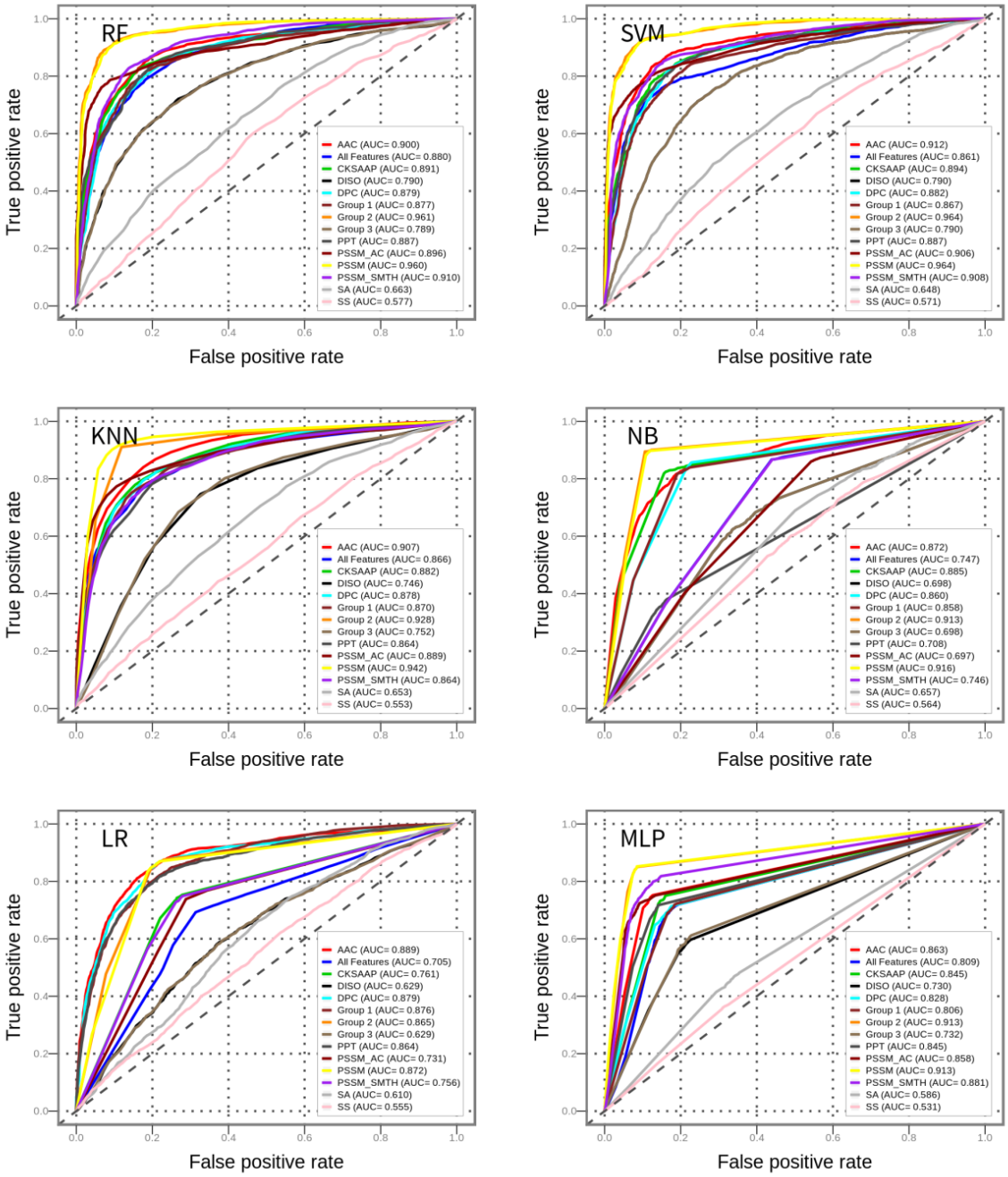


Figure S1. ROC curves of RF, SVM, NB, KNN, LR and MLP predictors of type IV secreted effectors with different feature encodings. Group 1 denotes the amino acid composition group (AAC, DPC, CKSAAP and PPT); Group 2 denotes the PSSM group (PSSM, PSSM_AC and PSSM_SMTH); Group 3 denotes the structure group (SA, SS and DISO), while All features include all the 10 feature types and are used as a whole group. For each group, corresponding features were combined into a vector to train the model.

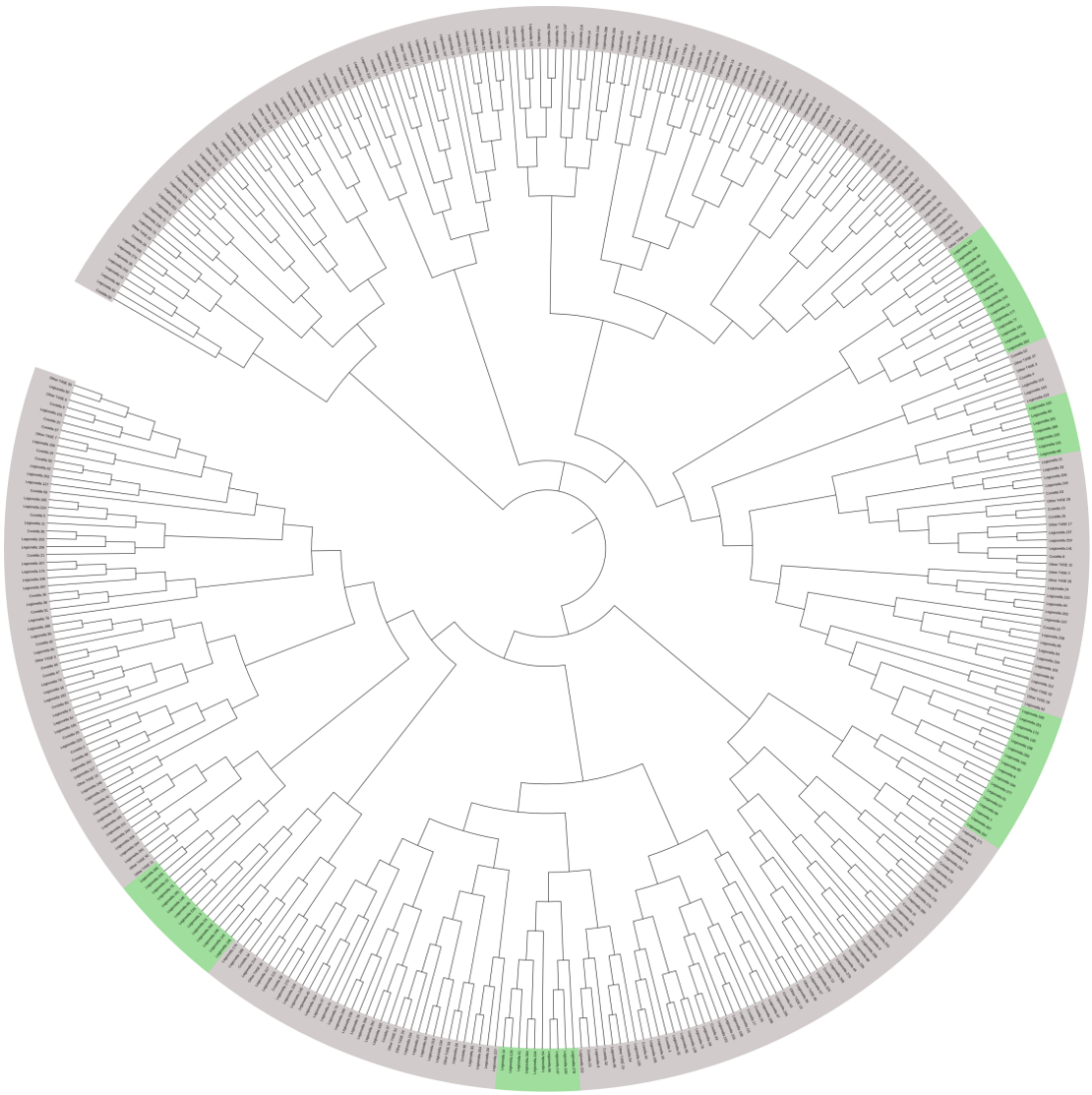


Fig. S2. Phylogenetic tree of T4SS effectors in the training dataset. Multiple sequence alignment for all the included proteins was carried out using Clustal Omega [1]. The phylogenetic tree was generated using iTOL [2].

References

1. Li W, Cowley A, Uludag M et al. The EMBL-EBI bioinformatics web and programmatic tools framework, Nucleic Acids Res 2015;43:W580-584.
2. Letunic I, Bork P. Interactive tree of life (iTOL) v3: an online tool for the display and annotation of phylogenetic and other trees, Nucleic Acids Res 2016;44:W242-245.

Appendix 2 - Supplementary information for Chapter 2.2

Bastion6: a bioinformatics approach for the accurate prediction of type VI secreted effector proteins

Supplementary Material

SUPPLEMENTAL INFORMATION

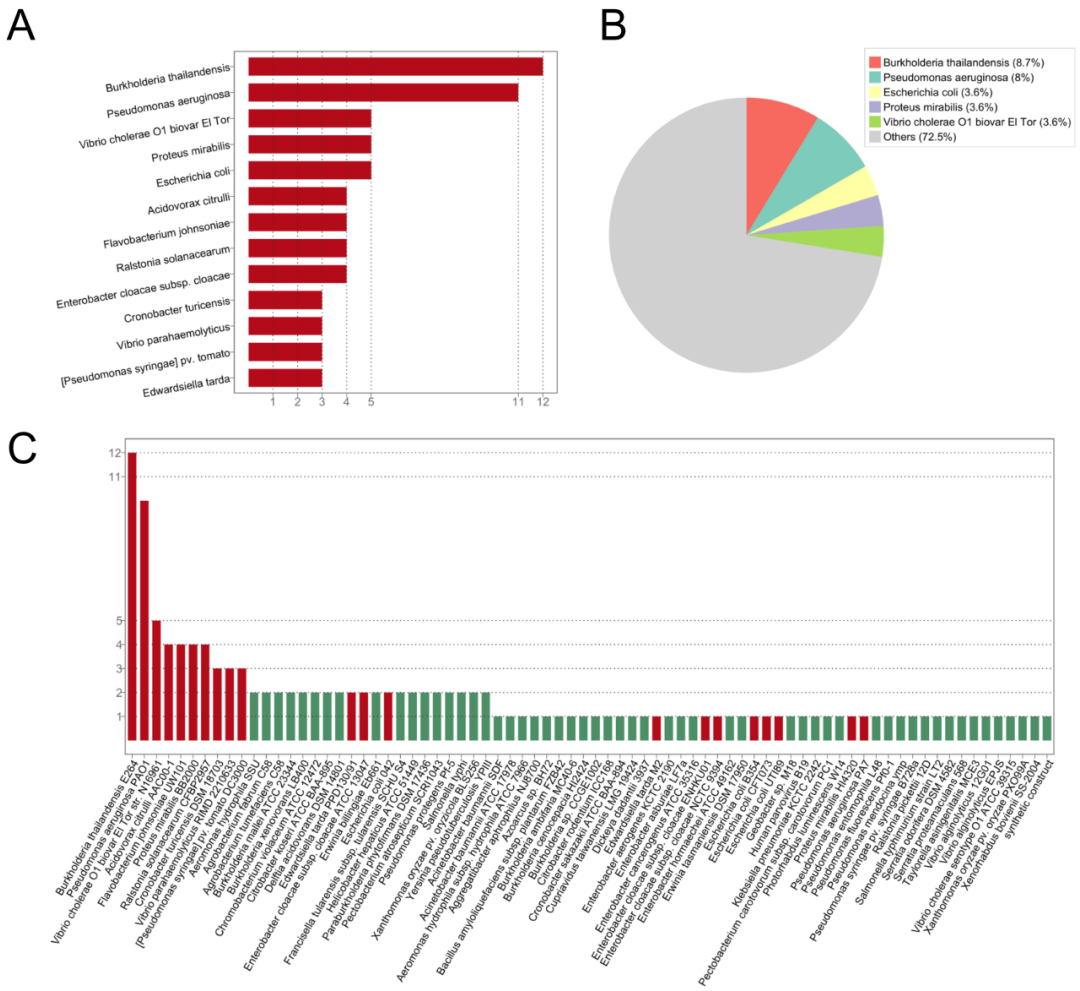


Fig. 1. Distribution of proteins in the training dataset by organism. (A) Bar chart listing the numbers of proteins in the training dataset for the top 13 species; (B) Pie chart showing the percentages of major protein associated species; and (C) Bar chart detailing the numbers of proteins per organism.

Table 1. Detailed information regarding positive samples in the independent dataset.

Effector ID	Effector name	Species	Reference	Comments
1	Hcp-effector	*Desulfobacterium autotrophicum*	(Wang, et al., 2015)	
2	Hcp3	*Pseudomonas fluorescens*	(Brunet, et al., 2015)	
3	EvpP	*Edwardsiella tarda*	(Durand, et al., 2014)	
4	VgrG3	*Pseudomonas fluorescens*	(Durand, et al., 2014)	
5	Tse1	*Pseudomonas aeruginosa*	(Durand, et al., 2014)	
6	Tae4	*Enterobacter cloacae*	(Durand, et al., 2014)	
7	TecA	*Burkholderia cenocepacia*	(Aubert, et al., 2016)	
8	VgrG2b	*Pseudomonas aeruginosa*	(Sana, et al., 2015)	
9	KatN	*Pseudomonas aeruginosa*	(Wan, et al., 2017)	
10	Tle1	*Escherichia coli*	(Flaugnatti, et al., 2016)	
-	RhsA	*Escherichia coli*	(Koskiniemi, et al., 2013)	Removed due to high similarity with RhsB
11	RhsB	*Escherichia coli*	(Koskiniemi, et al., 2013)	
12	Hcp-ET1	*Escherichia coli*	(Ma, et al., 2017)	
13	Hcp-ET2	*Escherichia coli*	(Ma, et al., 2017)	
14	Hcp-ET3 (1)	*Salmonella bongori*	(Ma, et al., 2017)	
-	Hcp-ET3 (2)	*Escherichia coli*	(Ma, et al., 2017)	Removed due to high similarity with Hcp-ET3+4
-	Hcp-ET3 (3)	*Escherichia coli*	(Ma, et al., 2017)	Removed due to high similarity with Hcp-ET3+4
15	Hcp-ET3 (4)	*Escherichia coli*	(Ma, et al., 2017)	
16	Hcp-ET3+4	*Escherichia coli*	(Ma, et al., 2017)	
17	Hcp-ET5	*Salmonella enterica*	(Ma, et al., 2017)	
18	Unclear	*Escherichia coli*	(Ma, et al., 2017)	
19	MIX-effector1	*Vibrio proteolyticus*	(Salomon, 2016)	
20	MIX-effector2	*Vibrio proteolyticus*	(Salomon, 2016)	

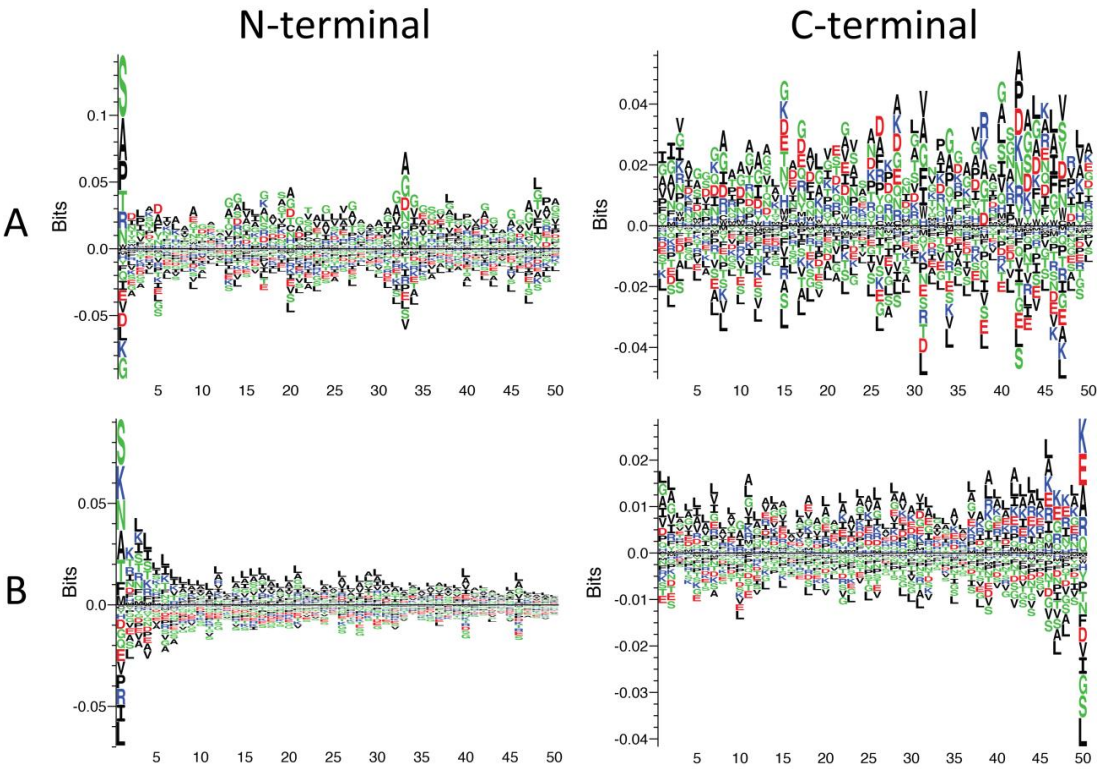


Fig. 2. Position-specific amino acid sequence profiles of 138 T6SEs and 1112 non-effectors, for 50 different N- and C-terminal positions. Images were generated with Seq2Logo (Thomsen and Nielsen, 2012) using the default settings. The positive y-axis depicts enriched amino acids in terms of amount of information in bits, while negative y-axis depicts corresponding depleted amino acids. The horizontal axis represents the N-/C-terminal position number. For the N terminal sequences, the methionine (M) at position 1 of each sequence is removed to improve readability. Here, the height of the stack represents the conservation level at each position, while the size of the letters depicts the relative frequency of each amino acid. (A) and (B) illustrate sequence logo representations for T6SEs and non-effectors, respectively.

Table 2. Details of distributions of T6SEs and non-T6SEs in each cluster.

Encoding	Cluster1			Cluster2		
	Total	T6SEs	non-T6SEs	Total	T6SEs	non-T6SEs
AAC	115	96 (83.5%)	19 (16.5)	161	42 (26.1%)	119 (73.9%)
DPC	105	92 (87.6%)	13 (12.4%)	171	46 (26.9%)	125 (73.1%)
QSO	68	49 (72.1%)	19 (27.9%)	208	89 (42.8%)	119 (57.2%)
BLOSUM	244	107 (43.9%)	137 (56.1%)	32	31 (96.9%)	1 (3.1%)
DPC-PSSM	114	1 (0.9%)	113 (99.1%)	162	137 (84.6%)	25 (15.4%)
S-FPSSM	79	54 (68.4%)	25 (31.6%)	197	84 (42.6%)	113 (57.4%)
Pse-PSSM	118	91 (77.1%)	27 (22.9%)	158	47 (29.7%)	111 (70.3%)
CTDC	132	78 (59.1%)	54 (40.9%)	144	60 (41.7%)	84 (58.3%)
CTDT	238	124 (52.1%)	114 (47.9%)	38	14 (36.8%)	24 (63.2%)

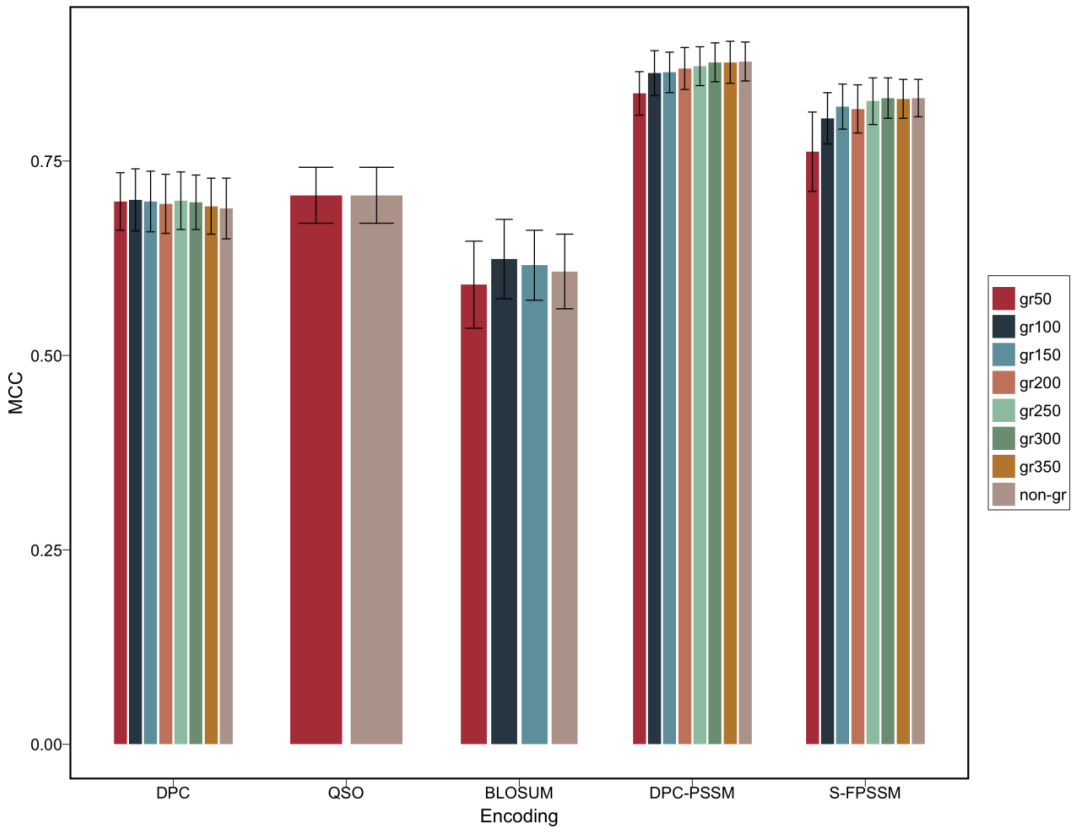


Fig. 3. Comparisons of the performance of various feature encoding methods with different numbers of top features, selected by GainRatio based on 5-fold cross-validation tests. grX (X=50,100,150,200,250,300,350) means top X features as ranked by GainRatio, while non-gr means full features without feature selection by GainRatio.

Table 3. The detailed prediction performance of various models in the independent test.

	Model	SN	SP	ACC	F-value	MCC
Single feature-based models	AAC	0.900±0.000	0.875±0.075	0.887±0.038	0.890±0.033	0.777±0.074
	DPC	0.800±0.000	0.865±0.097	0.833±0.049	0.829±0.041	0.670±0.101
	QSO	0.850±0.000	0.875±0.072	0.863±0.036	0.862±0.031	0.727±0.074
	BLOSUM	0.800±0.000	0.830±0.101	0.815±0.050	0.814±0.042	0.634±0.105
	DPC-PSSM	0.950±0.000	0.745±0.101	0.848±0.051	0.863±0.039	0.712±0.088
	S-FPSSM	0.750±0.000	0.770±0.116	0.760±0.058	0.760±0.042	0.523±0.115
	Pse-PSSM	0.950±0.000	0.780±0.111	0.865±0.056	0.878±0.044	0.743±0.098
	CTDC	0.900±0.000	0.850±0.094	0.875±0.047	0.880±0.040	0.753±0.090
	CTDT	0.850±0.000	0.795±0.064	0.823±0.032	0.828±0.026	0.647±0.063
Ensemble model	Group 1	0.850±0.000	0.880±0.079	0.865±0.039	0.864±0.034	0.733±0.081
	Group 2	**1.000±0.000**	0.825±0.072	0.912±0.036	0.920±0.030	0.839±0.062
	Group 3	0.950±0.000	0.840±0.088	0.895±0.044	0.902±0.038	0.797±0.082
Final ensemble model	Bastion6	**1.000±0.000**	**0.885±0.053**	**0.943±0.026**	**0.946±0.024**	**0.892±0.049**

Table 4. Prediction results of positive samples from the independent dataset using single encoding method-based models, group based ensemble models and Bastion6. Here, samples with a prediction score larger than 0.5 are recognized as T6SS effectors, and otherwise as non-T6SS effectors (marked in grey).

Effector ID	Effector Name	Single encoding method based model									Ensemble Model			Bastion6
		AAC	DPC	QSO	BLOSUM	DPC-PSSM	S-FPSSM	Pse-PSSM	CTDC	CTDT	Group 1	Group 2	Group 3	
1	Hcp	0.960	0.919	0.938	0.970	0.950	0.939	0.976	0.913	0.815	0.939	0.959	0.864	0.921
2	Hcp3	0.658	0.854	0.819	0.482	0.980	0.841	0.961	0.440	0.715	0.777	0.816	0.577	0.723
3	EvpP	0.752	0.850	0.838	0.787	0.966	0.864	0.905	0.568	0.585	0.813	0.880	0.576	0.757
4	VgrG3	0.921	0.940	0.977	0.931	0.987	0.961	0.948	0.719	0.750	0.946	0.957	0.735	0.879
5	Tse1	0.831	0.708	0.843	0.710	0.963	0.939	0.981	0.905	0.781	0.794	0.898	0.843	0.845
6	Tae4	0.842	0.923	0.848	0.948	0.992	0.968	0.968	0.897	0.798	0.871	0.969	0.848	0.896
7	TecA	0.575	0.423	0.313	0.299	0.987	0.495	0.819	0.693	0.768	0.437	0.650	0.730	0.606
8	VgrG2b	0.934	0.936	0.941	0.845	0.969	0.993	0.959	0.838	0.810	0.937	0.941	0.824	0.901
9	KatN	0.684	0.838	0.746	0.643	0.830	0.186	0.361	0.803	0.929	0.756	0.505	0.866	0.709
10	Tle1	0.406	0.572	0.536	0.482	0.336	0.450	0.824	0.746	0.460	0.505	0.523	0.603	0.543
11	RhsB	0.987	0.979	0.987	0.935	0.957	0.707	0.950	0.961	0.929	0.985	0.888	0.945	0.939
12	Hcp-ET1	0.584	0.375	0.210	0.491	0.956	0.909	0.848	0.632	0.413	0.390	0.801	0.522	0.571
13	Hcp-ET2	0.507	0.475	0.572	0.829	0.767	0.391	0.570	0.588	0.386	0.518	0.639	0.487	0.548
14	Hcp-ET3 (1)	0.878	0.909	0.939	0.956	0.928	0.725	0.754	0.842	0.675	0.909	0.841	0.758	0.836
15	Hcp-ET3 (4)	0.926	0.932	0.971	0.940	0.968	0.967	0.923	0.903	0.792	0.943	0.949	0.848	0.913
16	Hcp-ET3+4	0.707	0.741	0.889	0.793	0.969	0.777	0.812	0.484	0.647	0.779	0.838	0.565	0.728
17	Hcp-ET5	0.679	0.501	0.514	0.893	0.973	0.242	0.745	0.662	0.713	0.565	0.713	0.688	0.655
18	Unclear	0.432	0.446	0.374	0.850	0.980	0.847	0.873	0.851	0.641	0.417	0.887	0.746	0.684
19	MIX-effector1	0.989	0.965	0.976	0.707	0.950	0.982	0.993	0.953	0.906	0.976	0.908	0.930	0.938
20	MIX-effector2	0.987	0.920	0.953	0.783	0.945	0.976	0.983	0.888	0.689	0.953	0.922	0.789	0.888

Table 5. Detailed prediction results of Bastion 6 and of two motif-based methods for positive samples in the independent dataset. Misclassified proteins are marked in grey.

Effector ID	Effector Name	Bastion6		Motif Methods	
				MIX	SAVC
1	Hcp	0.921	√	×	×
2	Hcp3	0.723	√	×	×
3	EvpP	0.757	√	×	×
4	VgrG3	0.879	√	×	×
5	Tse1	0.845	√	×	×
6	Tae4	0.896	√	×	×
7	TecA	0.606	√	×	×
8	VgrG2b	0.901	√	×	×
9	KatN	0.709	√	×	×
10	Tle1	0.543	√	×	×
11	RhsB	0.939	√	×	√
12	Hcp-ET1	0.571	√	×	×
13	Hcp-ET2	0.548	√	×	√
14	Hcp-ET3 (1)	0.836	√	×	×
15	Hcp-ET3 (4)	0.913	√	×	×
16	Hcp-ET3+4	0.728	√	×	×
17	Hcp-ET5	0.655	√	×	×
18	Unclear	0.684	√	×	×
19	MIX-effector1	0.938	√	×	×
20	MIX-effector2	0.888	√	×	×

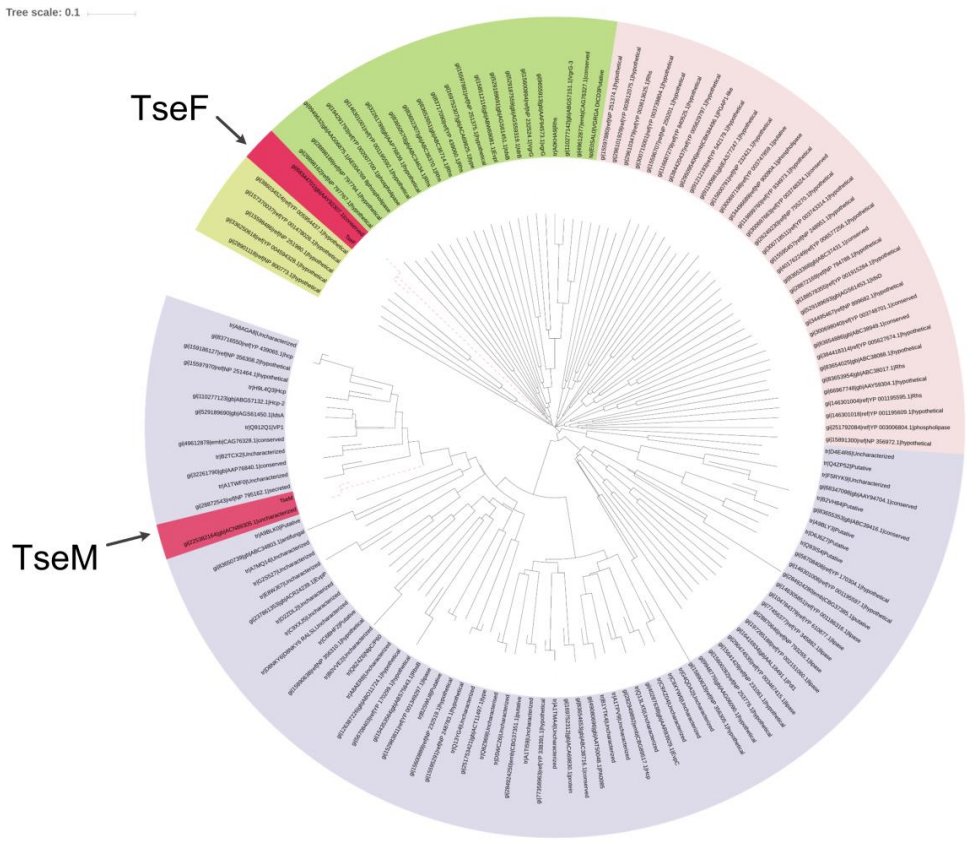


Fig. 4. Phylogenetic tree of all T6SEs in the training dataset and the two case study proteins TseM and TseF. Multiple sequence alignment was constructed for all the included proteins using Clustal Omega (Li, et al., 2015), with the phylogenetic tree generated using iTOL (Letunic and Bork, 2016).

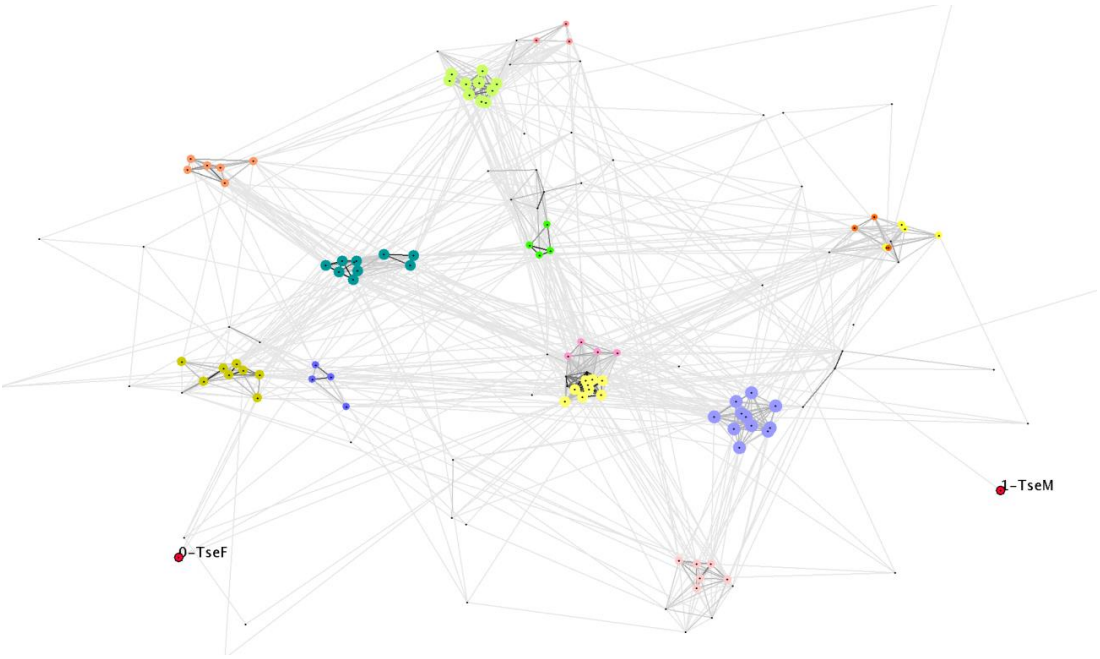


Fig. 5. Graphical two-dimensional representation of sequence similarities between the T6SS effectors of the training dataset and two case study effectors using the software CLANS. To draw a three-dimensional graph (projected here onto two dimensions), we performed all-against-all BLAST searches and used all significant high-scoring segment pairs (HSPs). In the graph, each node represents a T6SS effector protein and each edge (shaded according to p-value) represents a significant HSP with a p-value lower than 0.05 Each cluster is highlighted, while TseM and TseF are marked in the graph.

Table 6. Detailed prediction results of single encoding method based models, group based ensemble models and Bastion6, for two case study T6SS effector sequences.

Effector Name	Single encoding method based model									Ensemble Model			Bastion6
	AAC	DPC	QSO	BLOSUM	DPC-PSSM	S-FPSSM	Pse-PSSM	CTDC	CTDT	Group 1	Group 2	Group 3	
TseM	0.351	0.497	0.390	0.267	0.728	0.445	0.734	0.585	0.763	0.413	0.544	0.674	0.544
TseF	0.809	0.808	0.606	0.396	0.168	0.902	0.500	0.815	0.805	0.741	0.491	0.810	0.681

Table 7. Statistics of T6SE prediction results from 54,212 sequences of 12 bacterial species scanned by Bastion6. We list results using different thresholds, noting all results were filtered by readily validated T6SS effectors.

Species	Total number	>=0.5	>=0.6	>=0.7	>=0.8	>=0.9
Acidovorax citrulli strain AAC00-1	4652	925	495	225	83	29
Klebsiella pneumoniae AJ218	5108	524	299	142	47	4
Klebsiella pneumoniae B5055	5198	552	308	131	34	3
Burkholderia thailandensis E264	5763	954	497	240	89	14
Cronobacter turicensis z3032	3987	556	303	173	68	11
Flavobacterium johnsoniae UW101	5101	1009	594	284	70	6
Legionella pneumoniae Phi1	2943	212	88	34	6	0
Klebsiella pneumoniae MGH78578	4859	495	274	132	36	3
Proteus mirabilis BB2000	3325	364	211	111	40	1
Pseudomonas aeruginosa PAO1	5558	690	388	198	94	12
Ralstonia solanacearum CFBP2957	3174	481	216	90	17	3
Vibrio parahaemolyticus RIMD 2210633	4544	530	309	162	67	8

Reference

Aubert, D.F., *et al*. A Burkholderia Type VI Effector Deamidates Rho GTPases to Activate the Pyrin Inflammasome and Trigger Inflammation. *Cell host & microbe* 2016;19(5):664-674.

Brunet, Y.R., *et al*. The Type VI secretion TssEFGK-VgrG phage-like baseplate is recruited to the TssJLM membrane complex via multiple contacts and serves as assembly platform for tail tube/sheath polymerization. *PLoS Genet* 2015;11(10):e1005545.

Durand, E., *et al*. VgrG, Tae, Tle, and beyond: the versatile arsenal of Type VI secretion effectors. *Trends in microbiology* 2014;22(9):498-507.

Flaugnatti, N., *et al*. A phospholipase A1 antibacterial Type VI secretion effector interacts directly with the C-terminal domain of the VgrG spike protein for delivery. *Molecular microbiology* 2016;99(6):1099-1118.

Koskiniemi, S., *et al*. Rhs proteins from diverse bacteria mediate intercellular competition. *Proceedings of the National Academy of Sciences of the United States of America* 2013;110(17):7032-7037.

Letunic, I. and Bork, P. Interactive tree of life (iTOL) v3: an online tool for the display and annotation of phylogenetic and other trees. *Nucleic acids research* 2016;44(W1):W242-245.

Li, W., *et al*. The EMBL-EBI bioinformatics web and programmatic tools framework. *Nucleic acids research* 2015;43(W1):W580-584.

Ma, J., *et al*. The Hcp proteins fused with diverse extended-toxin domains represent a novel pattern of antibacterial effectors in type VI secretion systems. *Virulence* 2017:1-14.

Salomon, D. MIX and match: mobile T6SS MIX-effectors enhance bacterial fitness. *Mobile genetic elements* 2016;6(1):e1123796.

Sana, T.G., *et al*. Internalization of Pseudomonas aeruginosa Strain PAO1 into Epithelial Cells Is Promoted by Interaction of a T6SS Effector with the Microtubule Network. *mBio* 2015;6(3):e00712.

Thomsen, M.C. and Nielsen, M. Seq2Logo: a method for construction and visualization of amino acid binding motifs and sequence profiles including sequence weighting, pseudo counts and two-sided representation of amino acid enrichment and depletion. *Nucleic acids research* 2012;40(Web Server issue):W281-287.

Wan, B., *et al*. Type VI secretion system contributes to Enterohemorrhagic Escherichia coli virulence by secreting catalase against host reactive oxygen species (ROS). *PLoS pathogens* 2017;13(3):e1006246.

Wang, N., *et al*. Protective efficacy of recombinant hemolysin co-regulated protein (Hcp) of Aeromonas hydrophila in common carp (Cyprinus carpio). *Fish & shellfish immunology* 2015;46(2):297-304.

Appendix 3 - Supplementary information for Chapter 2.3

Bastion3: a two-layer ensemble predictor of type III secreted effectors

Supplementary Material

Table S1. A summary of the key aspects of previously developed T3SE prediction methods in comparison with Bastion3.

Method	Algorithm	Sequence length used for extracting features	Features	Species	Dataset size: Number of proteins	Ensemble learning strategy used	Web server or software accessibility	Reference
SIEVE	SVM	N30	AAC, SEQ, GC, CONS and PHYL	*S. Typhimurium P. syringae*	T3SE: 65 non-T3SE: -	N	http://cbb.pnnl.gov/portal/tools/sieve.html/	(Samudrala, et al., 2009)
EffectiveT3	NB	N100	AAC, SEQ, GC, CONS and PHYL	5 species	T3SE: 100 non-T3SE: 200	N	http://www.effectors.org/method/effectivet3/	(Arnold, et al., 2009)
T3SS_prediction	ANN SVM	N30	N-AAC	*P. syringae* and other species	T3SE: 575 non-T3SE: 685	N	Unavailable	(Löwer and Schneider, 2009)
T3Edb	NB	N100	PP	28 species	T3SE: 100 non-T3SE: 100	N	Unavailable	(Tay, et al., 2010)
SSE-ACC	SVM	N100	AAC, SS and ACC	*P. syringae*	T3SE: 108 non-T3SE: 3424	N	Unavailable	(Yang, et al., 2010)
BPBAac	SVM	N100	AAC	unclear	T3SE: 154 non-T3SE: 308	N	http://biocomputer.bio.cuhk.edu.hk/T3DB/BPBAac.php/	(Wang, et al., 2011)
T3SPs	RF	N100	AAC, SS, ACC, PP	16 species	T3SE: 189 non-T3SE: 385	N	Unavailable	(Yang, et al., 2013)
T3_MM	Markov model	N100	Conditional dependence of AAC	unclear	T3SE: 154 non-T3SE: 308	N	http://biocomputer.bio.cuhk.edu.hk/T3DB/T3_MM.php/	(Wang, et al., 2013)
T3SEpre	SVM	N100	AAC, SS and ACC	unclear	T3SE: 189 non-T3SE: 385	N	http://biocomputer.bio.cuhk.edu.hk/T3DB/T3SEpre.php/	(Wang, et al., 2013)
BEAN2.0	BLAST SVM	N120 C50	S, D, AAC PSSM	unclear	T3SE: 243 non- T3SE: 486	N	http://systbio.cau.edu.cn/bean/	(Dong, et al., 2015)
pEffect	BLAST SVM	Full sequence	S, PSSM	43 species	T3SE: 115 non-T3SE: 3460	N	http://services.bromberglab.org/peffect/	(Goldberg, et al., 2016)
Bastion3	LightGBM	Full sequence	AAC, DPC, QSO, CTDC, CTDT, PSSM, S_FPSSM, Pse_FPSSM, SS_FPSSM	62 species	T3SE: 379 non-T3SE: 1112	Y	http://bastion3.erc.monash.edu/	Proposed in this work.

Abbreviations: T3SE, type III secreted effector; *S. typhimurium*, *Salmonella Typhimurium*; *P. syringae, Pseudomonas syringae*; N, No; Y, Yes.
Note: N*xxx* indicates the corresponding number of features was extracted based on the top *xxx* amino acids from the N-terminus of the sequence. In comparison, C*xxx* indicates the corresponding number of features was extracted based on the top *xxx* amino acids from the C-terminus of the sequence.

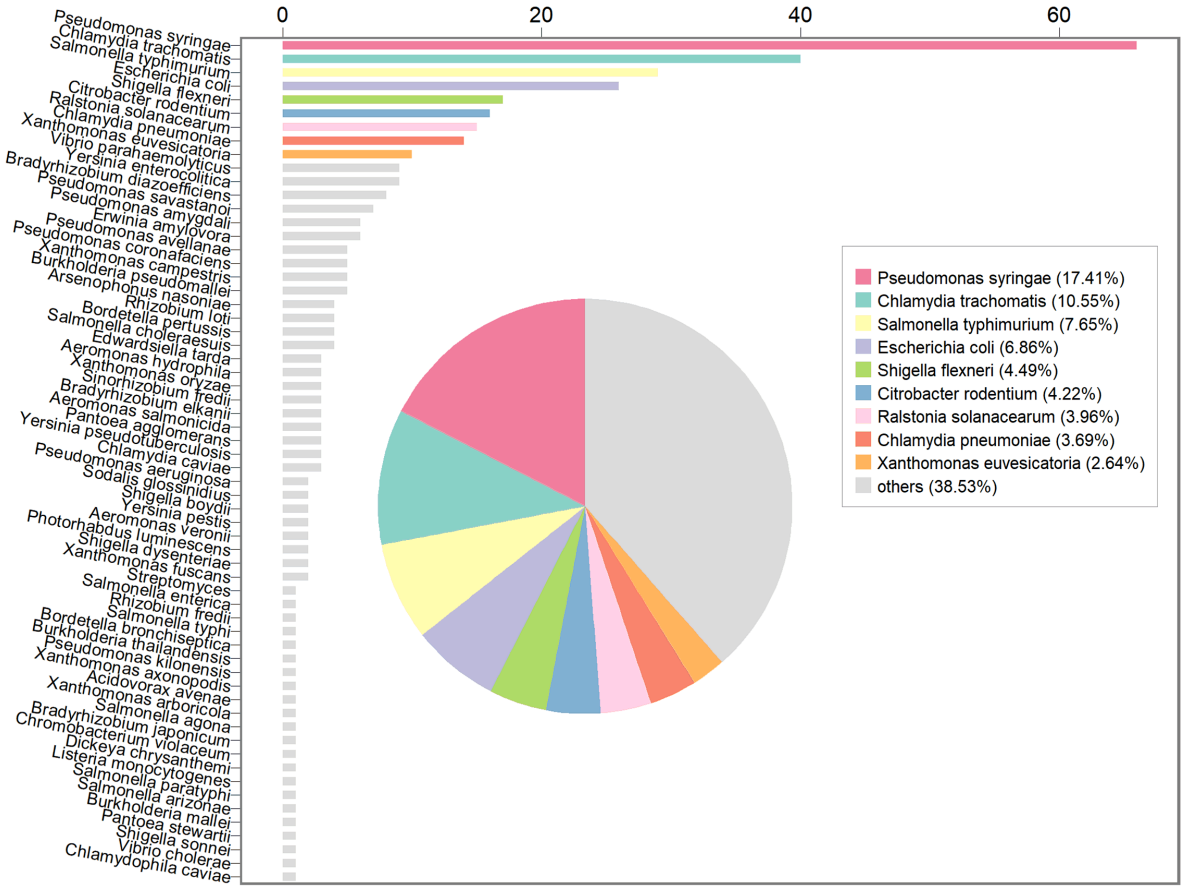


Fig. S1. Distribution of 379 T3SE proteins in the training dataset, by organism. Bar chart lists the numbers of proteins in the training dataset for all 62 species; Pie chart shows the percentages of major protein associated species.

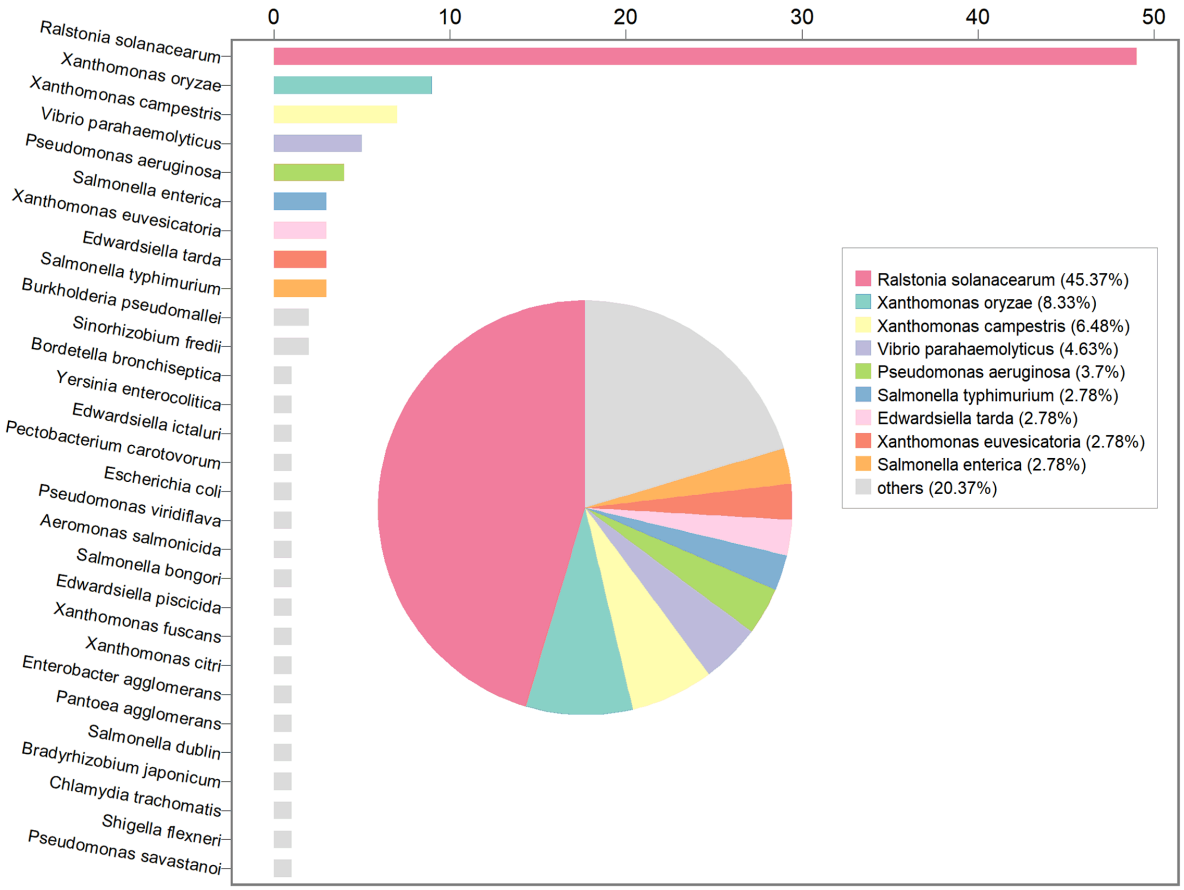


Fig. S2. Distribution of 108 T3SE proteins in the independent dataset, by species. Bar chart lists the numbers of proteins in the independent dataset for all 29 species; Pie chart shows the percentages of major protein associated species.

Table S2. Detailed information of positive samples in the independent dataset.

Effector ID	Effector/Gene name	Species	Reference
1	OrgC	*Salmonella enterica*	(Day and Lee, 2003)
2	ExoY	*Pseudomonas aeruginosa*	(Yahr, et al., 1998)
3	EspD	*Edwardsiella*	(Tejeda-Dominguez, et al., 2017)
4	sboH	*Salmonella bongori*	(Fookes, et al., 2011)
5	VPA0450	*Vibrio parahaemolyticus*	(Waddell, et al., 2014)
6	AopP	*Aeromonas salmonicida*	(Dorohonceanu and Nevill-Manning, 2000)
7	SlrP	*Salmonella enterica*	(Bernal-Bayard and Ramos-Morales, 2009)
8	ExoU	*Pseudomonas aeruginosa*	(McMorran, et al., 2003)
9	HopPsyA	*Pseudomonas viridiflava*	(van Dijk, et al., 2002)
10	GogB	*Escherichia coli*	(Tobe, et al., 2006)
11	DspE	*Pectobacterium carotovorum*	(Hogan, et al., 2013)
12	EseI	*Edwardsiella ictaluri*	(Rogge, et al., 2013)
13	YspI	*Yersinia enterocolitica*	(LeGrand, et al., 2015)
14	BspR	*Bordetella bronchiseptica*	(Abe, et al., 2015)
15	NopP	*Rhizobium*	(Ausmees, et al., 2004)
16	SteB	*Salmonella typhimurium*	(Geddes, et al., 2005)
17	HrpZ	*Pseudomonas savastanoi*	(Li, et al., 2002)
18	NopM	*Rhizobium*	(Xin, et al., 2012)
19	BapA	*Burkholderia pseudomallei*	(Treerat, et al., 2015)
20	EseH	*Edwardsiella tarda*	(Hou, et al., 2017)
21	PA14_16720	*Pseudomonas aeruginosa*	(Burstein, et al., 2015)
22	PA14_44480	*Pseudomonas aeruginosa*	(Burstein, et al., 2015)
23	ORF13	*Shigella flexneri*	(Pinaud, et al., 2017)
24	RSp0213	*Ralstonia solanacearum*	(Lonjon, et al., 2016)
25	XopAD	*Xanthomonas campestris*	(Teper, et al., 2016)
26	XopAP	*Xanthomonas campestris*	(Teper, et al., 2016)
27	XopAK	*Xanthomonas campestris*	(Teper, et al., 2016)
28	XopAU	*Xanthomonas campestris*	(Teper, et al., 2016)
29	XopAW	*Xanthomonas campestris*	(Teper, et al., 2016)
30	eseJ	*Edwardsiella tarda*	(Xie, et al., 2015)
31	CT_695	*Chlamydia trachomatis*	(Mueller and Fields, 2015)
32	MA20_12780	*Bradyrhizobium japonicum*	(Tsurumaru, et al., 2015)

33	xopG	*Xanthomonas campestris*	(Teper, et al., 2015)
34	xopH	*Xanthomonas campestris*	(Teper, et al., 2015)
35	XopI	*Xanthomonas oryzae*	(Teper, et al., 2015)
36	XopK	*Xanthomonas oryzae*	(Teper, et al., 2015)
37	XopV	*Xanthomonas oryzae*	(Teper, et al., 2015)
38	XopZ1	*Xanthomonas oryzae*	(Teper, et al., 2015)
39	XopAK	*Xanthomonas oryzae*	(Teper, et al., 2015)
40	AvrBsT	*Xanthomonas euvesicatoria*	(Teper, et al., 2015)
41	avrXv3	*Xanthomonas euvesicatoria*	(Teper, et al., 2015)
42	AvrXv4	*Xanthomonas euvesicatoria*	(Teper, et al., 2015)
43	XopAE	*Xanthomonas oryzae*	(Teper, et al., 2015)
44	Pthxo3	*Xanthomonas oryzae*	(Hutin, et al., 2015)
45	VopO	*Vibrio parahaemolyticus*	(Hiyoshi, et al., 2015)
46	SpvD	*Salmonella dublin*	(Patton, et al., 2016)
47	cigR	*Salmonella enterica*	(Yin, et al., 2016)
48	RipAY	*Ralstonia solanacearum*	(Fujiwara, et al., 2016)
49	VPA1336	*Vibrio parahaemolyticus*	(Zhou, et al., 2013)
50	VPA1350	*Vibrio parahaemolyticus*	(Zhou, et al., 2013)
51	vopA	*Vibrio parahaemolyticus*	(Zhou, et al., 2013)
52	hpx2	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
53	hpx4	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
54	hpx8	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
55	hpx9	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
56	hpx11	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
57	hpx18	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
58	hpx23	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
59	hpx25	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
60	hpx26	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
61	hpx27	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
62	hpx28	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
63	hpx30	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
64	hpx32	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
65	hpx33	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
66	hpx34	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
67	hpx35	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
68	hpx36	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)

69	hpx13	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
70	hpx14	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
71	hpx16	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
72	rip53	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
73	rip28	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
74	rip1	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
75	rip3	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
76	rip4	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
77	rip6	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
78	rip10	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
79	rip15	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
80	rip16	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
81	rip22	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
82	rip30	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
83	rip31	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
84	rip32	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
85	rip36	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
86	rip41	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
87	rip42	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
88	rip51	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
89	rip57	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
90	rip59	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
91	rip61	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
92	rip62	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
93	rip67	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
94	rip69	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
95	rip71	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
96	rip72	*Ralstonia solanacearum*	(Mukaihara, et al., 2010)
97	PseB	*Pantoea agglomerans*	(Nissan, et al., 2018)
98	HrpK	*Enterobacter/Pantoea agglomerans*	(Nissan, et al., 2018)
99	pip	*Xanthomonas citri*	(Kan, et al., 2018)
100	RSp0672	*Ralstonia solanacearum*	(Angot, et al., 2006)
101	RSc1800	*Ralstonia solanacearum*	(Peeters, et al., 2013)
102	XOO3803	*Xanthomonas oryzae*	(Fan, et al., 2017)
103	PXO_03702	*Xanthomonas oryzae*	(Fan, et al., 2017)
104	EseG	*Edwardsiella tarda*	(Xie, et al., 2010)

105	BPSS1385	*Burkholderia pseudomallei*	(Vander Broek and Stevens, 2017)
106	SopF	*Salmonella typhimurium*	(Cheng, et al., 2017)
107	SsrB	*Salmonella typhimurium*	(Cordero-Alba, et al., 2012)
108	avrGf2	*Xanthomonas fuscans*	(Gochez, et al., 2017)

Table S3. Detailed information of T3SE proteins used in the case study.

Effector ID	Effector/Gene name	Species	Reference
1	XCV1197 (XopAV)	*Xanthomonas campestris*	(Teper, et al., 2016)
2	HaRxL23	*Hyaloperonospora arabidopsidis*	(Deb, et al., 2018)
3	YggG	*Salmonella Typhimurium*	(Li, et al., 2018)

Algorithm S1: Gradient-based One-Side Sampling

Input: *I*: training data, *d*: iterations
Input: *a*: sampling ratio of large gradient data
Input: *b*: sampling ratio of small gradient data
Input: *loss*: loss function, *L*: weak learner
models←(Day and Lee, 2003), fact← $(1 - a)/b$
topN←$a \times$ len(*I*) , randN←$b \times$ len(*I*)
for i = 1 **to** d **do**
    preds←models.predict(*I*)
    g←*loss*(I, preds), w←(Dorohonceanu and Nevill-Manning, 2000)
    sorted←GetSortedIndices(abs(g))
    topSet←sorted[1:topN]
    randSet←RandomPick(sorted[topN:len(I)], randN)
    usedSet←topSet + randSet
    w[randSet] ×= fact ▷ Assign weight *fact* to the small gradient data.
    newModel←L(*I*[usedSet], − g[usedSet], w[usedSet])
    models.append(newModel)

Algorithm S2: Greedy Bundling

Input: F: features, K: max conflict count
Construct graph G
searchOrder←*G*.sortByDegree()
bundles←{}, bundlesConflict←{}
for *i* **in** *searchOrder* **do**
    needNew←True
    for *j* = 1 **to** *len(bundles)* **do**
        cnt←ConflictCnt(bundles[j],F[i])
        if *cnt* + *bundlesConflict[i]* $\leqslant$ K **then**
            bundles[j].add(F[i]), needNew←False
            break
    if *needNew* **then**
        Add F[*i*] as a new bundle to *bundles*
Output: *bundles*

Note: The above algorithms come from (Ke, et al., 2017), where LightGBM was proposed. Please refer to such paper for more detailed description for the algorithms.

Table S4. Explanation of the 12 parameters on LightGBM.

Parameters	**Declaration**[a]	**Parameter tuning range**[b]
weight (alias: *weight_column*)	used to specify the weight column	[1,10], step: 1
learning_rate	shrinkage rate	$[2^{-10}, 2^{-1}]$, step: 2^1
num_leaves	number of leaves in one tree	[50, 800], step: 50
max_depth	max depth of the tree	[5,10], step: 1
min_data_in_leaf	minimal number of data in one leaf	$[2^1, 2^6]$, step: 2^1
max_bin	max number of bins that feature values will be bucketed in	$[2^5, 2^{10}]$, step: 2^1
feature_fraction	percentage of features selected before training each tree	[0.5,1], step: 0.02
min_sum_hessian	minimal sum hessian in one leaf	[0,0.02], step: 0.001
lambda_l1	L1 regularization	[0,0.01], step: 0.002
lambda_l2	L2 regularization	[0,0.01], step: 0.002
drop_rate	only used in dart	[0,1], step: 0.1
max_drop	max number of dropped trees on one iteration	[1,30], step: 2

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

[b]Parameter tuning range represents the tuning range and step length in the process of one-by-one parameter tuning (step 1).

Table S5. Detailed performance of using two-step parameter optimization for various feature encoding methods compared with first-step-only parameter optimization and initial parameter setting, based on 100-time 5-fold cross-validation.

Model	AAC	DPC	QSO	CTDC	CTDT	PSSM-composition	RPM-PSSM	D-FPSSM	TPC	DP-PSSM
default	0.613±0.012	0.515±0.016	0.625±0.014	0.611±0.010	0.488±0.011	0.820±0.008	0.731±0.012	0.720±0.012	0.824±0.008	0.836±0.008
step1	0.692±0.011	0.654±0.014	0.690±0.011	0.611±0.011	0.540±0.013	0.847±0.008	0.815±0.009	0.808±0.010	0.840±0.008	0.853±0.007
step1+step2	**0.694±0.011**	**0.656±0.013**	**0.705±0.012**	**0.619±0.010**	**0.549±0.013**	**0.857±0.006**	**0.828±0.009**	**0.809±0.011**	**0.843±0.008**	**0.858±0.007**

Note: Values were expressed as mean±standard deviation. To ease understanding, the best performance value for each encoding method across different models appears in bold font.

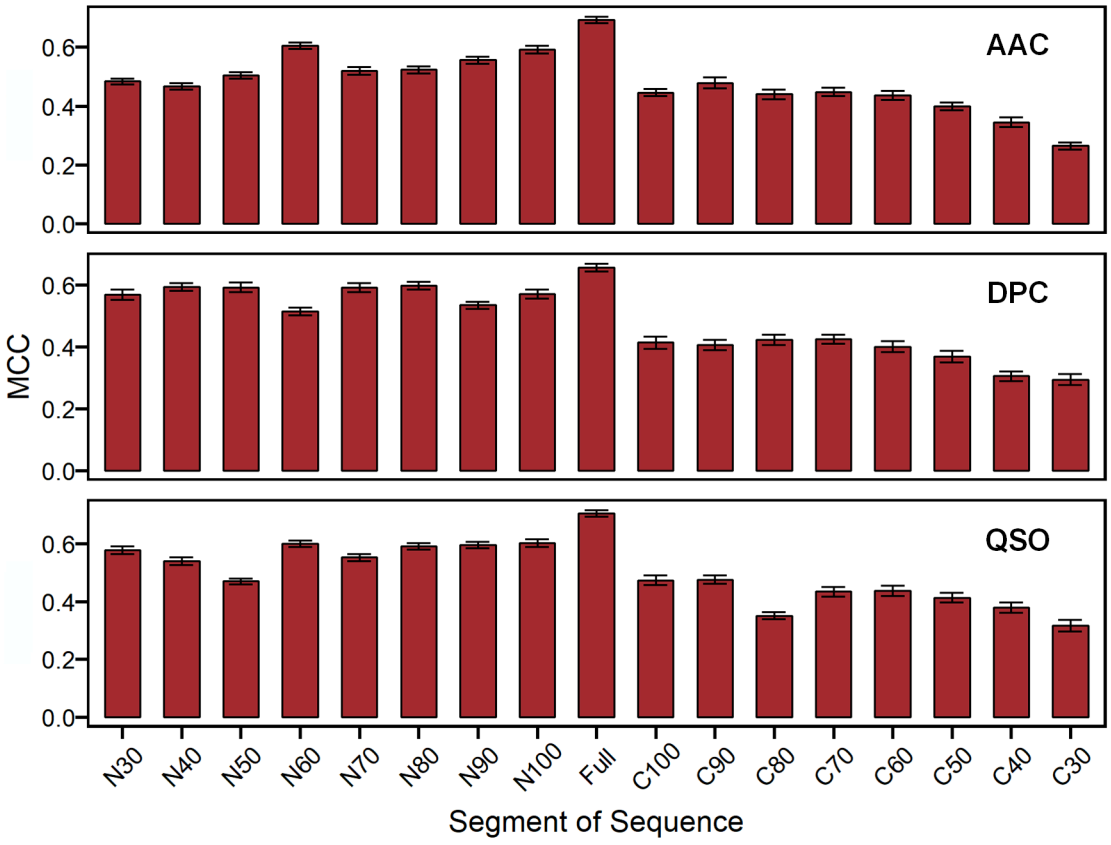


Fig. S3. Performance comparison of LightGBM models trained using different sizes of sequence segments based on three feature encoding methods. *Nxx* and *Cxx* represent the features extracted based on the top *xx* (*xx* denotes the length of the sequence segments) N-terminus and C-terminus of the sequences, respectively, while Full represents features extracted based on full protein sequences.

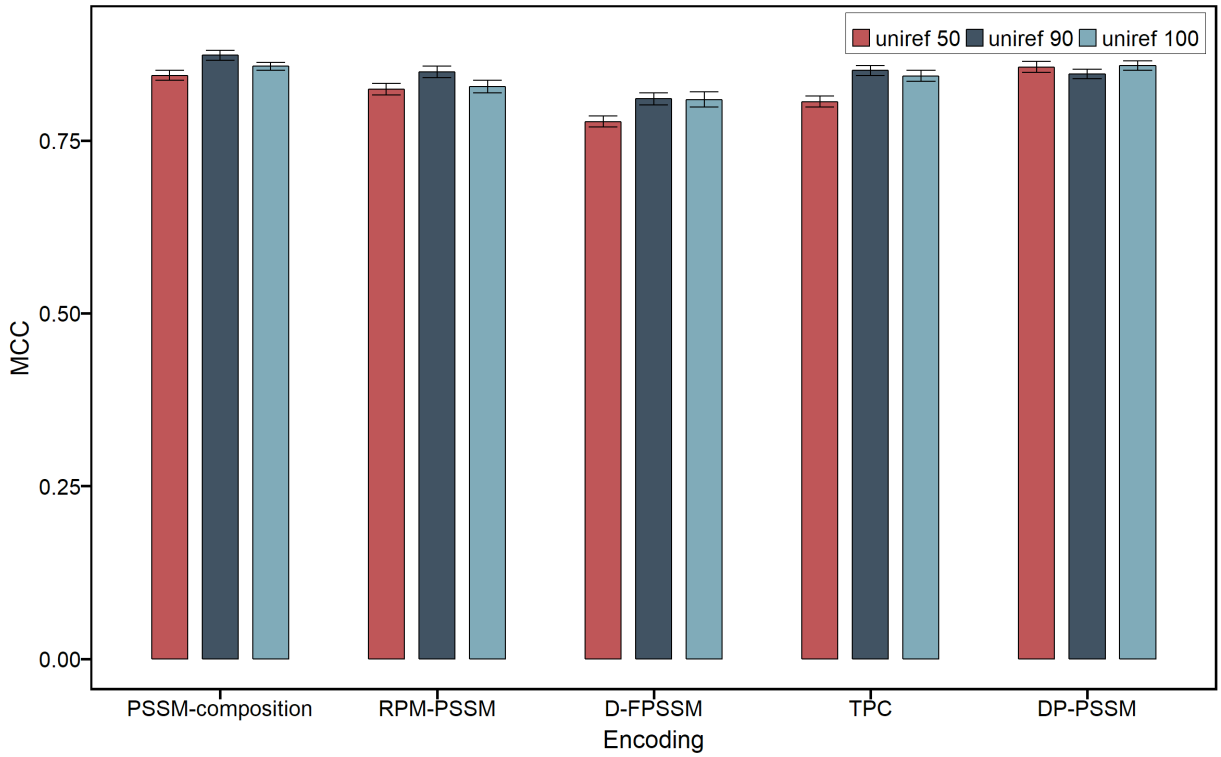


Fig. S4. Performance comparison of multiple PSSM-based feature encoding methods when using different uniref databases (*i.e.* uniref50, uniref90 and uniref100) based on 100-time 5-fold cross-validation test.

Table S6. Performance comparison of single feature-based models, group-based one-layer ensemble model and the final two-layer ensemble model on the independent test.

Model	SN	SP	ACC	F-value	MCC
AAC	0.870±0.036	0.838±0.089	0.856±0.049	0.860±0.041	0.711±0.099
DPC	0.806±0.031	0.886±0.045	0.848±0.025	0.842±0.028	0.696±0.050
QSO	0.862±0.061	0.863±0.083	0.865±0.038	0.865±0.040	0.729±0.077
CTDC	0.897±0.039	0.737±0.085	0.815±0.048	0.829±0.051	0.641±0.087
CTDT	0.787±0.132	0.867±0.064	0.821±0.075	0.814±0.080	0.654±0.146
PSSM-composition	**0.954±0.034**	0.900±0.058	0.930±0.014	0.932±0.018	0.861±0.027
RPM-PSSM	0.945±0.053	0.846±0.080	0.900±0.042	0.904±0.043	0.802±0.080
D-FPSSM	0.915±0.041	0.932±0.063	0.927±0.028	0.928±0.022	0.855±0.059
TPC	0.918±0.052	0.957±0.042	0.939±0.033	0.939±0.031	0.880±0.066
DP-PSSM	0.927±0.054	0.856±0.052	0.896±0.034	0.898±0.035	0.792±0.067
Group 1	0.886±0.046	0.890±0.071	0.890±0.040	0.892±0.031	0.780±0.083
Group 2	0.898±0.061	0.855±0.081	0.874±0.048	0.876±0.050	0.753±0.094
Group 3	**0.954±0.034**	0.935±0.051	0.948±0.012	0.949±0.008	0.897±0.023
Final model	**0.954±0.034**	**0.958±0.025**	**0.959±0.008**	**0.958±0.010**	**0.917±0.017**

Note: Values are expressed as mean±standard deviation. To facilitate understanding, the best performance value for each metric across different encoding methods is shown in bold font.

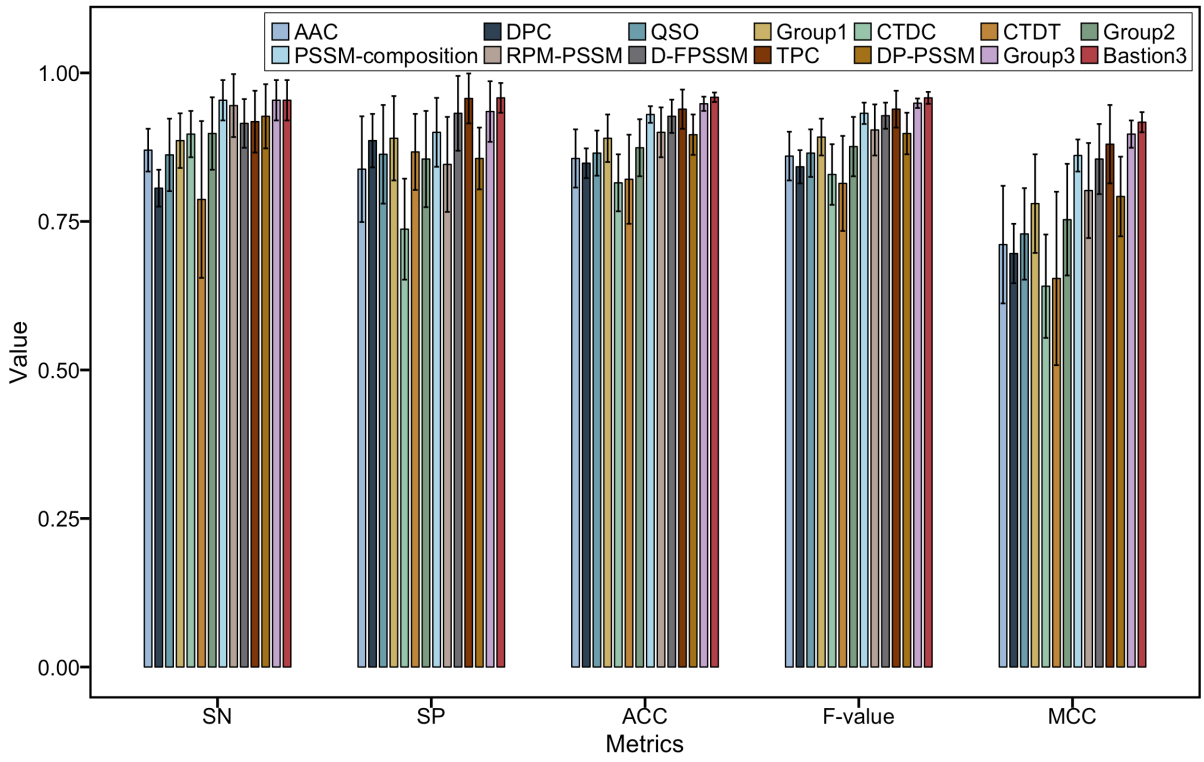


Fig. S5. Performance evaluation of single feature-based models, group-based one-layer ensemble and the final two-layer ensemble models based on the independent test.

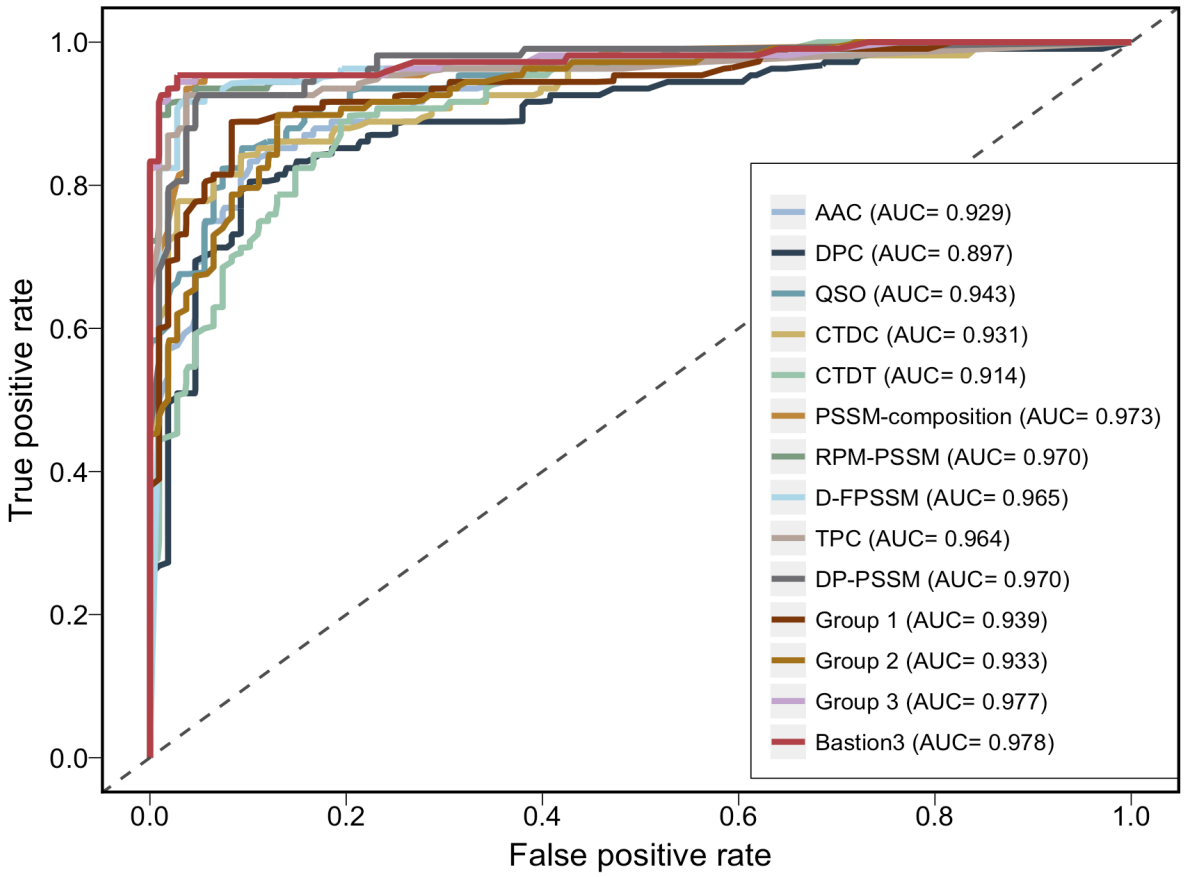


Fig. S6. ROC curves of single feature-based models, group-based one-layer ensemble models, and the final two-layer ensemble model used by Bastion3 based on the independent test. Results for each are color coded, and respective AUC values are also presented.

Table S7. Detailed prediction performance of Bastion3 (using the final two-layer ensemble model) and other existing state-of-the-art toolkits based on the independent test.

Toolkit	SN	SP	ACC	F-value	MCC
Bastion3	**0.954±0.034**	0.958±0.025	**0.959±0.008**	**0.958±0.010**	**0.917±0.017**
BEAN 2.0	0.891±0.060	0.917±0.069	0.908±0.015	0.906±0.024	0.816±0.033
pEffect	0.878±0.088	0.909±0.041	0.895±0.048	0.889±0.066	0.790±0.100
EffectiveT3	0.741±0.086	0.873±0.037	0.809±0.038	0.794±0.051	0.623±0.068
T3_MM	0.804±0.040	0.783±0.054	0.795±0.031	0.797±0.043	0.588±0.066
BPBAac	0.288±0.067	0.978±0.031	0.629±0.062	0.437±0.082	0.371±0.072
SIEVE	0.122±0.057	**1.000±0.000**	0.557±0.048	0.214±0.091	0.247±0.063

Note: Values are expressed as mean±standard deviation. To facilitate understanding, the best performance value for each metric across different toolkits is shown in bold font.

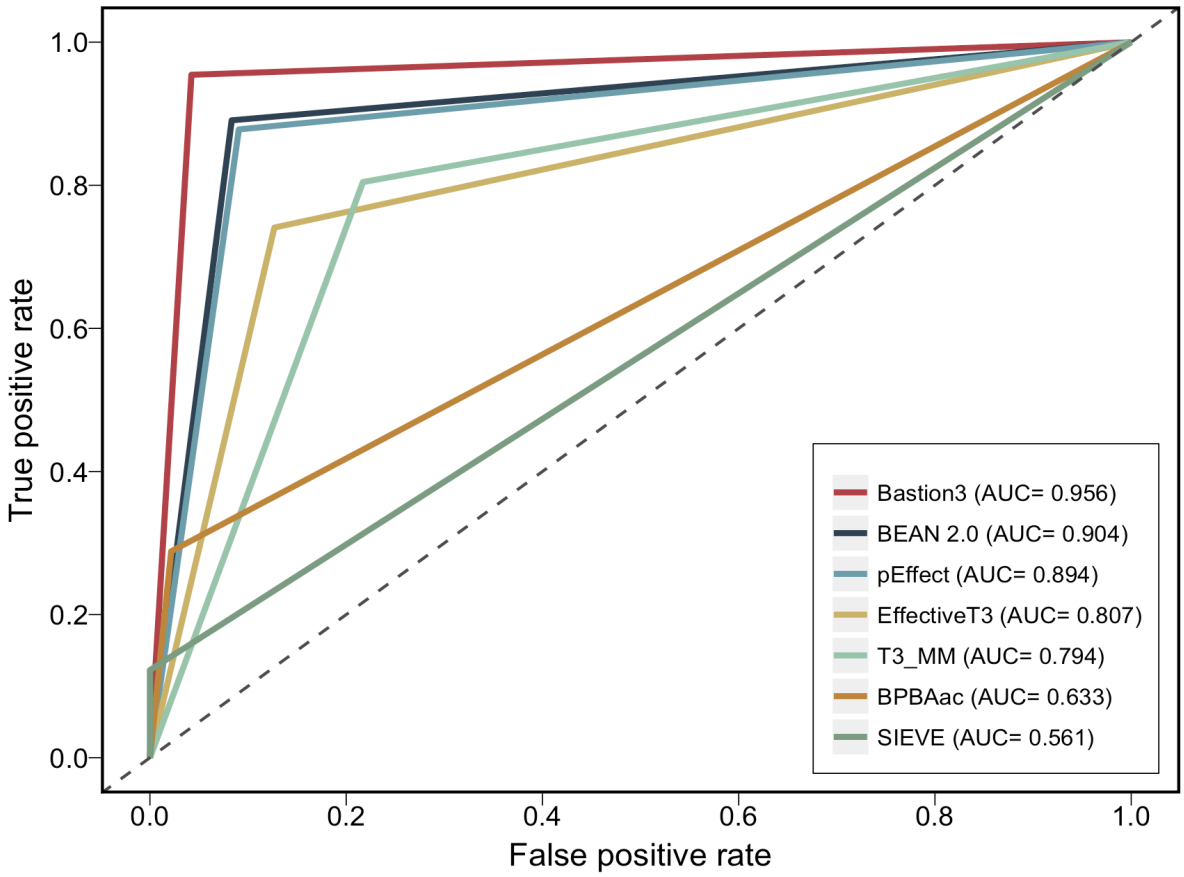


Fig. S7. ROC curves of Bastion3 (using the final two-layer ensemble model) and other existing state-of-the-art toolkits based on the independent test. Results for each are color coded, and respective AUC values are also presented. Curves composed by connected lines in the figure were obtained due to the fact that most of the existing toolkits predicted samples with a true or false label, but without giving the detailed probability score. Therefore, to make a fair comparison, we first transformed the prediction probabilities of Bastion3 and SIEVE into predictive labels (true or false), and then uniformly used the predictive labels to generate the ROC curves for all the toolkits under comparison in this study.

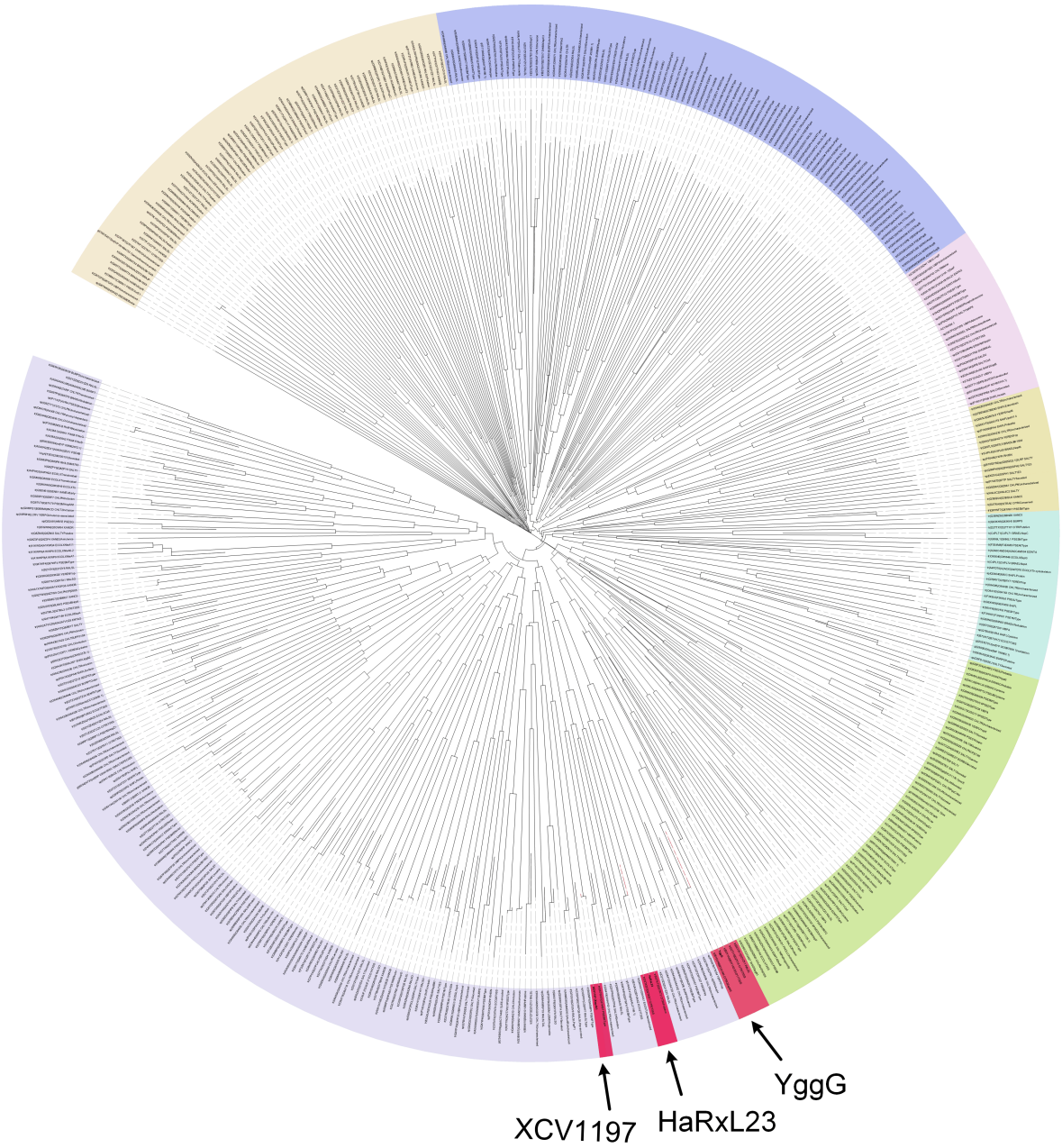


Fig. S8. Phylogenetic tree of all T3SEs in the training dataset, independent dataset and the three case study proteins XCV1197 (XopAV), HaRxL23 and YggG. We used Clustal Omega (Li, et al., 2015) to generate multiple sequence alignment for all these proteins, based on which we plotted the phylogenetic tree using iTOL (Letunic and Bork, 2016). T3SEs under the same red area indicates a relatively closer relationship with the case study protein in terms of sequence similarity.

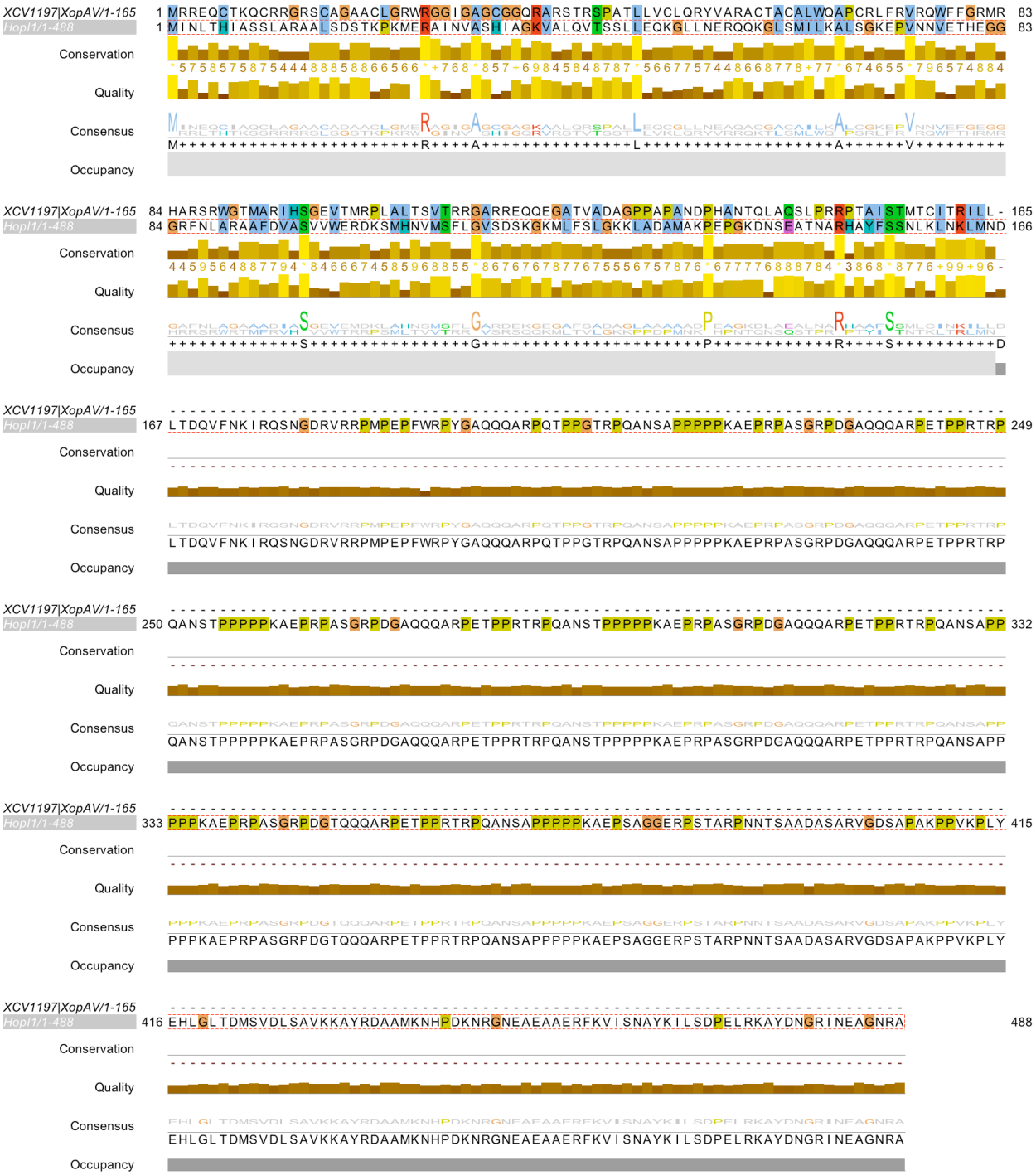


Fig. S9. Visualization of the pair-wise sequence alignment between the case study protein XCV1197 (XopAV) and its closest sequence homologue (HopI1; UniProt ID: Q87W07; located in training dataset). We used Jalview (Clamp, et al., 2004) to visualize the pair-wise sequence alignment result, which was generated by T-Coffee (Notredame, et al., 2000) with default settings per Jalview's built-in function.

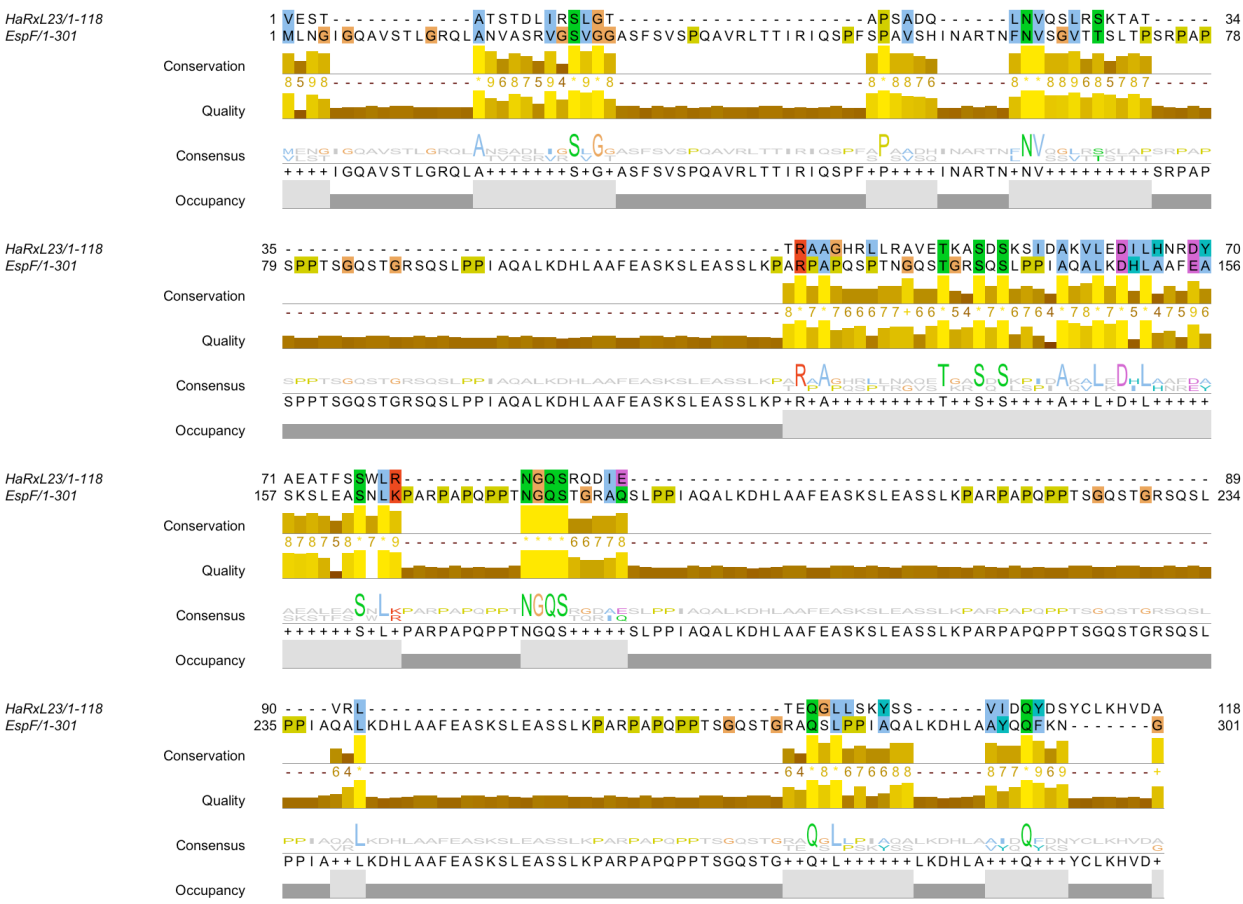


Fig. S10. Visualization of the pair-wise sequence alignment between the case study protein HaRxL23 and its closest sequence homologue (EspF; UniProt ID: D2TKD7; located in training dataset). We used Jalview (Clamp, et al., 2004) to visualize the pair-wise sequence alignment result, which was generated by T-Coffee (Notredame, et al., 2000) with default settings per Jalview's built-in function.

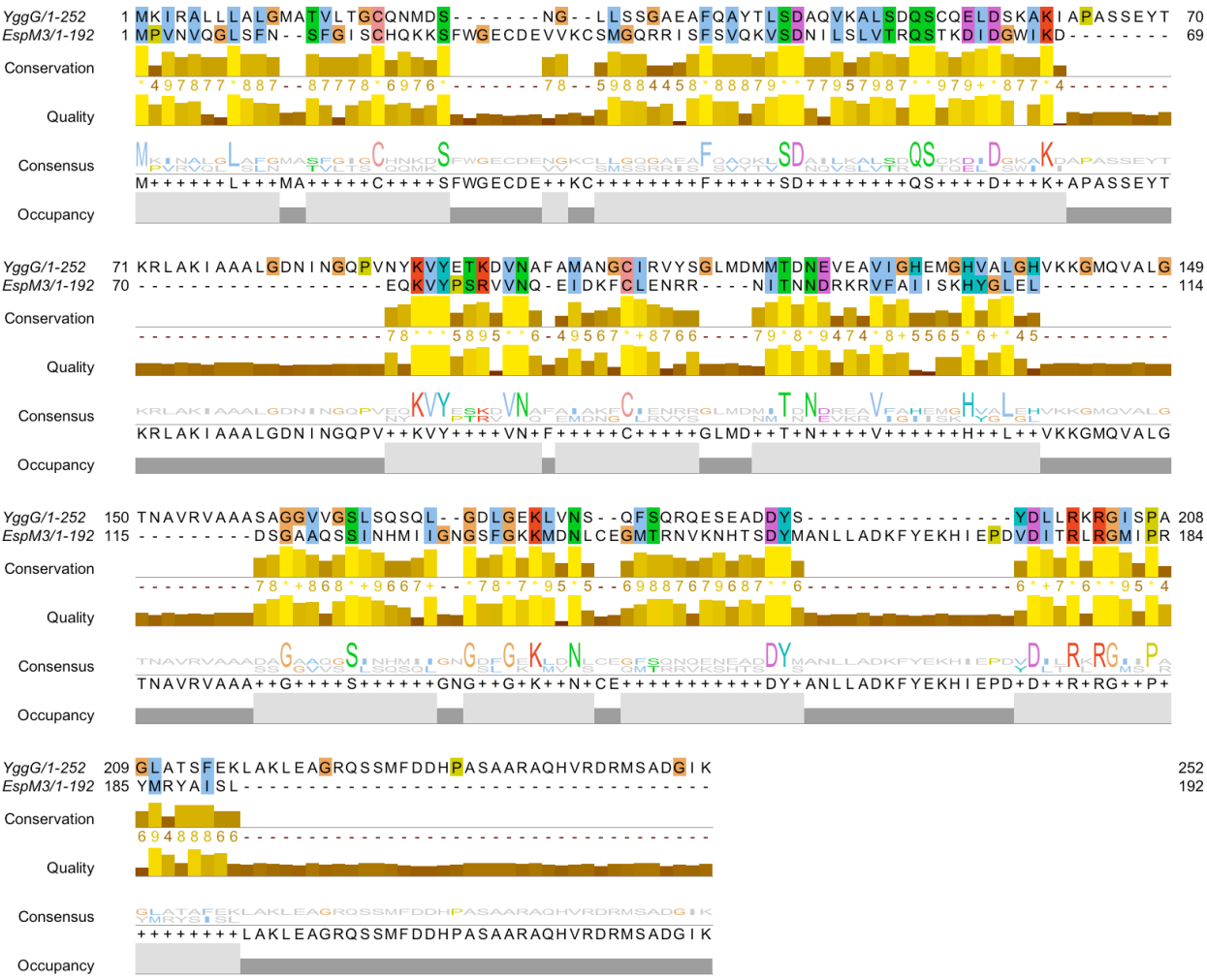


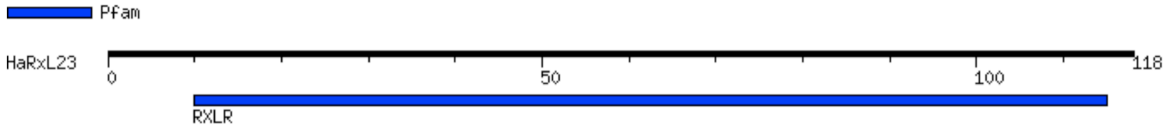
Fig. S11. Visualization of the pair-wise sequence alignment between the case study protein YggG and its closest sequence homologue (EspM3; UniProt ID: B1GVN9; located in training dataset). We used Jalview (Clamp, et al., 2004) to visualize the pair-wise sequence alignment result, which was generated by T-Coffee (Notredame, et al., 2000) with default settings per Jalview's built-in function.

A **Result of MotifFinder**

Number of found motif: 0

No motif was found in Pfam.

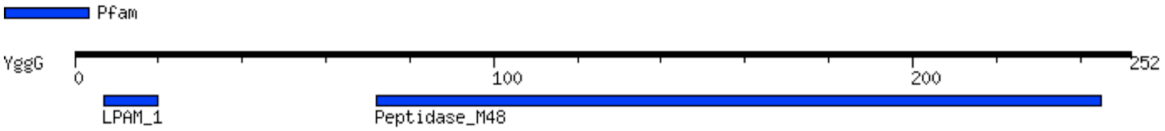
B **Number of found motif: 1**



Pfam (1 motif)

Pfam	Position(Independent E-value)		Description
RXLR	10..115(1.4e-20)	Detail	PF16810, RXLR phytopathogen effector protein, Avirulence activity

C **Number of found motifs: 2**



Pfam (2 motifs)

Pfam	Position(Independent E-value)		Description
Peptidase_M48	72..245(1.5e-25)	Detail	PF01435, Peptidase family M48
LPAM_1	7..20(0.032)	Detail	PF08139, Prokaryotic membrane lipoprotein lipid attachment site

Fig. S12. Visualization of the conserved sequence motifs in the three case study proteins: (A) XCV1197 (XopAV), (B) HaRxL23 and (C) YggG. The MOTIF Search service in the GenomeNet resources (Kanehisa, 1997) was used to search each of the three respective case study proteins against the Pfam database (Bateman, et al., 2002). As a result, 0, 1 and 2 motifs were detected in the proteins XCV1197 (XopAV), HaRxL23 and YggG, respectively.

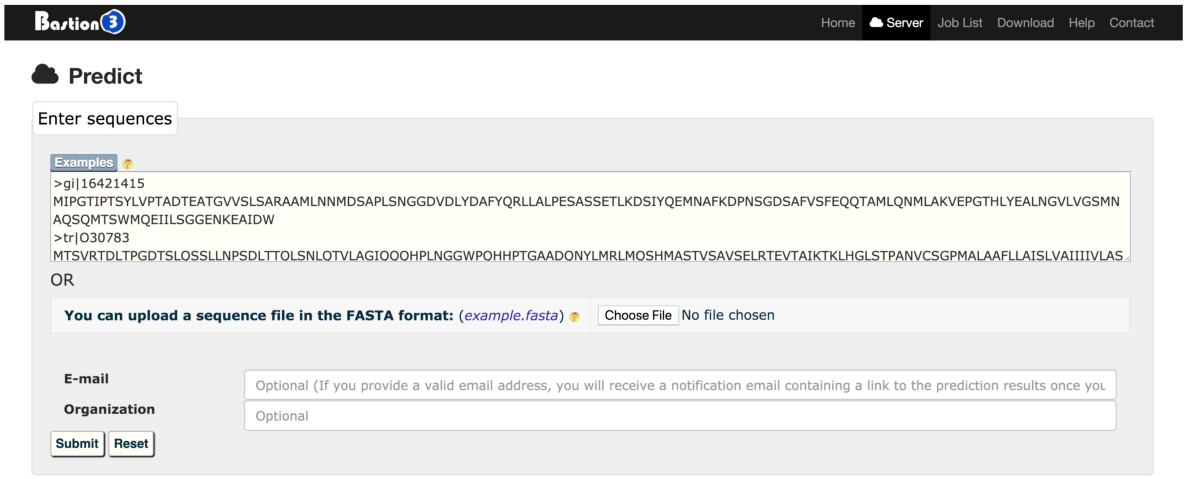
Table S8. Detailed prediction results of the three T3SE samples used in the case study using single encoding method-based models, group-based one-layer ensemble models and the final two-layer ensemble model. Here, samples with a prediction score larger than 0.5 are recognized as T3SS effectors, and otherwise as non-T3SS effectors (marked in grey).

Effector ID	Effector name	Single encoding method based model										Ensemble model			Final model
		AAC	DPC	QSO	CTDC	CTDT	PSSM-composition	RPM-PSSM	D-FPSSM	TPC	DP-PSSM	Group 1	Group 2	Group 3	
1	XopAV	0.400	0.413	0.205	0.317	0.281	0.947	0.856	0.474	0.835	0.657	0.339	0.299	0.754	0.536
2	HaRxL23	0.991	0.764	0.825	0.989	0.755	0.984	0.838	0.113	0.995	0.942	0.860	0.872	0.774	0.820
3	YggG	0.902	0.941	0.749	0.984	0.880	0.501	0.329	0.998	0.967	0.865	0.864	0.932	0.732	0.815

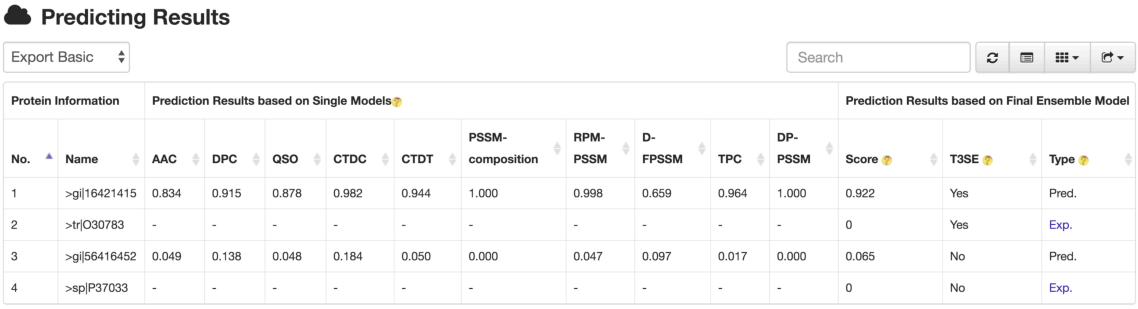
Table S9. Detailed prediction results of Bastion3 (using the final two-layer ensemble model) and other existing state-of-the-art toolkits for T3SE samples regarding the case study. Misclassified protein samples are marked in grey.

Effector ID	Effector Name	Bastion3		BEAN 2.0	pEffect	EffectiveT3	T3_MM	BPBAac	SIEVE
1	XopAV	0.536	√	√	×	×	×	×	×
2	HaRxL23	0.820	√	√	×	√	√	×	×
3	YggG	0.815	√	×	×	√	×	×	×

A



B



Protein Information		Prediction Results based on Single Models										Prediction Results based on Final Ensemble Model		
No.	Name	AAC	DPC	QSO	CTDC	CTDT	PSSM-composition	RPM-PSSM	D-FPSSM	TPC	DP-PSSM	Score	T3SE	Type
1	>gi|16421415	0.834	0.915	0.878	0.982	0.944	1.000	0.998	0.659	0.964	1.000	0.922	Yes	Pred.
2	>tr|O30783	-	-	-	-	-	-	-	-	-	-	0	Yes	Exp.
3	>gi|56416452	0.049	0.138	0.048	0.184	0.050	0.000	0.047	0.097	0.017	0.000	0.065	No	Pred.
4	>sp|P37033	-	-	-	-	-	-	-	-	-	-	0	No	Exp.

Fig. S13. Screenshot of the Bastion3 online web server: (A) user submission interface and (B) predicted result for a case study using 4 protein sequences as input. The result is marked as '*Pred.*' for a computationally predicted protein, and '*Exp.*' if the predicted protein belongs to a known (experimentally verified) effector.

References

Abe, A.*, et al.* The Bordetella Secreted Regulator BspR Is Translocated into the Nucleus of Host Cells via Its N-Terminal Moiety: Evaluation of Bacterial Effector Translocation by the Escherichia coli Type III Secretion System. *PloS one* 2015;10(8):e0135140.

Angot, A.*, et al.* Ralstonia solanacearum requires F-box-like domain-containing type III effectors to promote disease on several host plants. *Proceedings of the National Academy of Sciences of the United States of America* 2006;103(39):14620-14625.

Arnold, R.*, et al.* Sequence-based prediction of type III secreted proteins. *PLoS pathogens* 2009;5(4):e1000376.

Ausmees, N.*, et al.* Characterization of NopP, a type III secreted effector of Rhizobium sp. strain NGR234. *Journal of bacteriology* 2004;186(14):4774-4780.

Bateman, A.*, et al.* The Pfam protein families database. *Nucleic acids research* 2002;30(1):276-280.

Bernal-Bayard, J. and Ramos-Morales, F. Salmonella type III secretion effector SlrP is an E3 ubiquitin ligase for mammalian thioredoxin. *The Journal of biological chemistry* 2009;284(40):27587-27595.

Burstein, D.*, et al.* Novel type III effectors in Pseudomonas aeruginosa. *mBio* 2015;6(2):e00161.

Cheng, S.*, et al.* Identification of a Novel Salmonella Type III Effector by Quantitative Secretome Profiling. *Molecular & cellular proteomics : MCP* 2017;16(12):2219-2228.

Clamp, M.*, et al.* The Jalview Java alignment editor. *Bioinformatics* 2004;20(3):426-427.

Cordero-Alba, M., Bernal-Bayard, J. and Ramos-Morales, F. SrfJ, a Salmonella type III secretion system effector regulated by PhoP, RcsB, and IolR. *Journal of bacteriology* 2012;194(16):4226-4236.

Day, J.B. and Lee, C.A. Secretion of the orgC gene product by Salmonella enterica serovar Typhimurium. *Infection and immunity* 2003;71(11):6680-6685.

Deb, D.*, et al.* Application of alignment-free bioinformatics methods to identify an oomycete protein with structural and functional similarity to the bacterial AvrE effector protein. *PloS one* 2018;13(4):e0195559.

Dong, X., Lu, X. and Zhang, Z. BEAN 2.0: an integrated web resource for the identification and functional analysis of type III secreted effectors. *Database : the journal of biological databases and curation* 2015;2015:bav064.

Dorohonceanu, B. and Nevill-Manning, C.G. Accelerating protein classification using suffix trees. *Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology* 2000;8:128-133.

Fan, S.*, et al.* Identification of phenolic compounds that suppress the virulence of Xanthomonas oryzae on rice via the type III secretion system. *Mol Plant Pathol* 2017;18(4):555-568.

Fookes, M.*, et al.* Salmonella bongori provides insights into the evolution of the Salmonellae. *PLoS pathogens* 2011;7(8):e1002191.

Fujiwara, S., *et al.* RipAY, a Plant Pathogen Effector Protein, Exhibits Robust gamma-Glutamyl Cyclotransferase Activity When Stimulated by Eukaryotic Thioredoxins. *The Journal of biological chemistry* 2016;291(13):6813-6830.

Geddes, K., *et al.* Identification of new secreted effectors in Salmonella enterica serovar Typhimurium. *Infection and immunity* 2005;73(10):6260-6271.

Gochez, A.M., *et al.* Molecular characterization of XopAG effector AvrGf2 from Xanthomonas fuscans ssp. aurantifolii in grapefruit. *Mol Plant Pathol* 2017;18(3):405-419.

Goldberg, T., Rost, B. and Bromberg, Y. Computational prediction shines light on type III secretion origins. *Scientific reports* 2016;6:34516.

Hiyoshi, H., *et al.* Interaction between the type III effector VopO and GEF-H1 activates the RhoA-ROCK pathway. *PLoS pathogens* 2015;11(3):e1004694.

Hogan, C.S., *et al.* The type III secreted effector DspE is required early in solanum tuberosum leaf infection by Pectobacterium carotovorum to cause cell death, and requires Wx(3-6)D/E motifs. *PloS one* 2013;8(6):e65534.

Hou, M., *et al.* Identification and functional characterization of EseH, a new effector of the type III secretion system of Edwardsiella piscicida. *Cell Microbiol* 2017;19(1).

Hutin, M., *et al.* A knowledge-based molecular screen uncovers a broad-spectrum OsSWEET14 resistance allele to bacterial blight from wild rice. *Plant J* 2015;84(4):694-703.

Kan, J., *et al.* A dual role for proline iminopeptidase in the regulation of bacterial motility and host immunity. *Mol Plant Pathol* 2018.

Kanehisa, M. Linking databases and organisms: GenomeNet resources in Japan. *Trends Biochem Sci* 1997;22(11):442-444.

Ke, G., *et al.* LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems* 2017:3149-3157.

LeGrand, K., Matsumoto, H. and Young, G.M. A novel type 3 secretion system effector, YspI of Yersinia enterocolitica, induces cell paralysis by reducing total focal adhesion kinase. *Cell Microbiol* 2015;17(5):688-701.

Letunic, I. and Bork, P. Interactive tree of life (iTOL) v3: an online tool for the display and annotation of phylogenetic and other trees. *Nucleic acids research* 2016;44(W1):W242-245.

Li, C.M., *et al.* The Hrp pilus of Pseudomonas syringae elongates from its tip and acts as a conduit for translocation of the effector protein HrpZ. *EMBO J* 2002;21(8):1909-1915.

Li, M., *et al.* YggG is a Novel SPI-1 Effector Essential for Salmonella Virulence. *bioRxiv* 2018.

Li, W., *et al.* The EMBL-EBI bioinformatics web and programmatic tools framework. *Nucleic acids research* 2015;43(W1):W580-584.

Lonjon, F., *et al.* Comparative Secretome Analysis of Ralstonia solanacearum Type 3 Secretion-Associated Mutants Reveals a Fine Control of Effector Delivery, Essential for Bacterial Pathogenicity. *Molecular & cellular proteomics : MCP* 2016;15(2):598-613.

Löwer, M. and Schneider, G. Prediction of Type III Secretion Signals in Genomes of Gram-Negative Bacteria. *PloS one* 2009;4(6):e5917.

McMorran, B., *et al.* Effector ExoU from the type III secretion system is an important modulator of gene expression in lung epithelial cells in response to Pseudomonas aeruginosa infection. *Infection and immunity* 2003;71(10):6035-6044.

Mueller, K.E. and Fields, K.A. Application of beta-lactamase reporter fusions as an indicator of effector protein secretion during infections with the obligate intracellular pathogen Chlamydia trachomatis. *PloS one* 2015;10(8):e0135295.

Mukaihara, T., Tamura, N. and Iwabuchi, M. Genome-wide identification of a large repertoire of Ralstonia solanacearum type III effector proteins by a new functional screen. *Mol Plant Microbe Interact* 2010;23(3):251-262.

Nissan, G.*, et al.* Revealing the inventory of type III effectors in Pantoea agglomerans gall-forming pathovars using draft genome sequences and a machine-learning approach. *Mol Plant Pathol* 2018;19(2):381-392.

Notredame, C., Higgins, D.G. and Heringa, J. T-Coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of molecular biology* 2000;302(1):205-217.

Patton, M.J.*, et al.* Chlamydial Protease-Like Activity Factor and Type III Secreted Effectors Cooperate in Inhibition of p65 Nuclear Translocation. *mBio* 2016;7(5).

Peeters, N.*, et al.* Repertoire, unified nomenclature and evolution of the Type III effector gene set in the Ralstonia solanacearum species complex. *BMC Genomics* 2013;14:859.

Pinaud, L.*, et al.* Identification of novel substrates of Shigella T3SA through analysis of its virulence plasmid-encoded secretome. *PloS one* 2017;12(10):e0186920.

Rogge, M.L.*, et al.* Comparison of Vietnamese and US isolates of Edwardsiella ictaluri. *Dis Aquat Organ* 2013;106(1):17-29.

Samudrala, R., Heffron, F. and McDermott, J.E. Accurate prediction of secreted substrates and identification of a conserved putative secretion signal for type III secretion systems. *PLoS pathogens* 2009;5(4):e1000375.

Tay, D.M.*, et al.* T3SEdb: data warehousing of virulence effectors secreted by the bacterial Type III Secretion System. *BMC bioinformatics* 2010;11 Suppl 7:S4.

Tejeda-Dominguez, F.*, et al.* A Novel Mechanism for Protein Delivery by the Type 3 Secretion System for Extracellularly Secreted Proteins. *mBio* 2017;8(2).

Teper, D.*, et al.* Identification of novel Xanthomonas euvesicatoria type III effector proteins by a machine-learning approach. *Mol Plant Pathol* 2016;17(3):398-411.

Teper, D.*, et al.* Five Xanthomonas type III effectors suppress cell death induced by components of immunity-associated MAP kinase cascades. *Plant Signal Behav* 2015;10(10):e1064573.

Tobe, T.*, et al.* An extensive repertoire of type III secretion effectors in Escherichia coli O157 and the role of lambdoid phages in their dissemination. *Proceedings of the National Academy of Sciences of the United States of America* 2006;103(40):14941-14946.

Treerat, P.*, et al.* The Burkholderia pseudomallei Proteins BapA and BapC Are Secreted TTSS3 Effectors and BapB Levels Modulate Expression of BopE. *PloS one* 2015;10(12):e0143916.

Tsurumaru, H.*, et al.* A Putative Type III Secretion System Effector Encoded by the MA20_12780 Gene in Bradyrhizobium japonicum Is-34 Causes Incompatibility with Rj4 Genotype Soybeans. *Appl Environ Microbiol* 2015;81(17):5812-5819.

van Dijk, K.*, et al.* The ShcA protein is a molecular chaperone that assists in the secretion of the HopPsyA effector from the type III (Hrp) protein secretion system of Pseudomonas syringae. *Molecular microbiology* 2002;44(6):1469-1481.

Vander Broek, C.W. and Stevens, J.M. Type III Secretion in the Melioidosis Pathogen Burkholderia pseudomallei. *Front Cell Infect Microbiol* 2017;7:255.

Waddell, B.*, et al.* Identification of VPA0451 as the specific chaperone for the Vibrio parahaemolyticus chromosome 1 type III-secreted effector VPA0450. *FEMS Microbiol Lett* 2014;353(2):141-150.

Wang, Y.*, et al.* T3_MM: a Markov model effectively classifies bacterial type III secretion signals. *PloS one* 2013;8(3):e58173.

Wang, Y.*, et al.* Effective identification of bacterial type III secretion signals using joint element features. *PloS one* 2013;8(4):e59754.

Wang, Y.*, et al.* High-accuracy prediction of bacterial type III secreted effectors based on position-specific amino acid composition profiles. *Bioinformatics* 2011;27(6):777-784.

Xie, H.X.*, et al.* Identification and functional characterization of the novel Edwardsiella tarda effector EseJ. *Infection and immunity* 2015;83(4):1650-1660.

Xie, H.X.*, et al.* EseG, an effector of the type III secretion system of Edwardsiella tarda, triggers microtubule destabilization. *Infection and immunity* 2010;78(12):5011-5021.

Xin, D.W.*, et al.* Functional analysis of NopM, a novel E3 ubiquitin ligase (NEL) domain effector of Rhizobium sp. strain NGR234. *PLoS pathogens* 2012;8(5):e1002707.

Yahr, T.L.*, et al.* ExoY, an adenylate cyclase secreted by the Pseudomonas aeruginosa type III system. *Proceedings of the National Academy of Sciences of the United States of America* 1998;95(23):13899-13904.

Yang, X.*, et al.* Effective identification of Gram-negative bacterial type III secreted effectors using position-specific residue conservation profiles. *PloS one* 2013;8(12):e84439.

Yang, Y.*, et al.* Computational prediction of type III secreted proteins from gram-negative bacteria. *BMC bioinformatics* 2010;11 Suppl 1:S47.

Yin, J.*, et al.* Construction and characterization of a cigR deletion mutant of Salmonella enterica serovar Pullorum. *Avian Pathol* 2016;45(5):569-575.

Zhou, X.*, et al.* A Vibrio parahaemolyticus T3SS effector mediates pathogenesis by independently enabling intestinal colonization and inhibiting TAK1 activation. *Cell reports* 2013;3(5):1690-1702.

Appendix 4 - Supplementary information for Chapter 3.1

BastionX: Systematic and accurate prediction of secreted substrates in Gram-negative bacteria within a distributed framework

Supplementary file

Table S1. Statistical summary of the datasets collected and curated in this study.

Datasets Number of the substrates (All/Less than 70% similarity)		Type I 196/161	Type II 81/79	Type III 511/504	Type IV 420/414	Type VI 160/148
Training set	Positive samples	132	62	410	339	119
	Negative samples	1112	1112	1112	1112	1112
Independent set	Positive samples	29	17	94	75	29
	Negative samples	29	17	94	75	29

Table S2. The detailed information of adopted feature encoding methods in this study.

Group	Method	Description	Dimension	Ref
Group 1	AAC	The frequency of the amino acids	20	(Liu, et al., 2008)
	DPC	The frequency of the dipeptides	400	(Liu, et al., 2008)
	DP	The PseAAC using the distance-pairs and reduced alphabet	$n+dn^2$ (n=14, d=3)	(Liu, et al., 2014)
	DDE	The dipeptide deviation from expected mean	400	(Chen, et al., 2018)
Group 2	QSOrder	The quasi-sequence-order feature	20+20+lag*2 (lag=30)	(Chou, 2000)
	CTDC	The composition among CTD (composition, transition and distribution)	N*3 (N=13)	(Lin, et al., 2007)
	PDT	Feature based on the physicochemical distance transformation	531*lamada (lamada=1)	(Liu, et al., 2012)
Group 13	RPSSM	Feature by calculating the correlation between two adjacent residues via importing the transition probability matrix into the PSSM profile	110	(Ding, et al., 2014)
	TPC-PSSM	Feature based on the transition probability matrix (TPM), which is extended from the PSSM to avoid complete loss of the sequence-order information	400	(Zhang, et al., 2012)
	RPM-PSSM	Feature based a 'filtered' PSSM that is generated by a residue probing method	400	(Jeong, et al., 2011)
	DP-PSSM	Extension of the Pse-PSSM feature encoding method to describe the relationship of an amino acid and its α-th succeeding amino acid	(α+1)* 40 (α=5)	(Juan, et al., 2009)
	Pse-PSSM	Feature based on a set of PSSM transformations and dimension normalization	40	(Chou and Shen, 2007)
	AAC-PSSM	Extension of the concept of traditional AAC feature encoding method from the primary sequence to the PSSM profile	20	(Liu, et al., 2010)
	AB-PSSM	Feature based on averaged blocks of the PSSM profile	400	(Jeong, et al., 2011)
	EEDP	Feature based on an ED-PSSM that is generated by an evolutionary formula (EDF)	400	(Zhang, et al., 2014)
	PSSM-composition	Feature by converting the original PSSM profile into a 20*20 matrix through summing up all rows of the same amino acid	400	(Zou, et al., 2013)

Note: Groups 1, 2 and 3 represent sequence based features, physicochemical property based features and evolutionary information based features, respectively.

Table S3. The detailed information of machine learning algorithms adopted in this study.

Machine learning	**Parameters**	**Parameter tuning range**
K-nearest neighbor (KNN)	*k*	[1,100], step: 1
Naïve Bayes (NB)	-	-
support vector machine (SVM)	*gamma*	$[2^{-10}, 2^{10}]$, step: 2^1
	cost	$[2^{-10}, 2^{10}]$, step: 2^1
random forest (RF)	*mtry*	1000
eXtreme Gradient Boosting (XGBoost)	*max_depth*	[4,8], step: 1
	eta	[0.01, 0.03], step: random
	gamma	[0, 0.2], step: random
	subsample	[0.6, 0.9], step: random
	colsample_bytree	[0.5, 0.8], step: random
	min_child_weight	[1, 40], step: 1
	max_delta_step	[1,10], step: 1
Light Gradient Boosting Machine (LightGBM)	*weight* (alias: *weight_column*)	[1,10], step: 1
	learning_rate	$[2^{-10}, 2^{-1}]$, step: 2^1
	num_leaves	[50, 800], step: 50
	max_depth	[3,10], step: 1
	min_data_in_leaf	$[2^1, 2^5]$, step: 2^1
	max_bin	$[2^5, 2^{10}]$, step: 2^1
	feature_fraction	[0.5,1], step: 0.02
	min_sum_hessian	[0,0.02], step: 0.001
	lambda_l1	[0,0.01], step: 0.002
	lambda_l2	[0,0.01], step: 0.002
	drop_rate	[0,1], step: 0.1
	max_drop	[1,30], step: 2

Table S4. Performance comparison of single-method-based models for predicting type I secreted substrates based on the 5-fold cross-validation test.

Model	Encoding	SN	SP	ACC	F-value	MCC
KNN	AAC	0.890±0.014	0.981±0.008	0.934±0.006	0.931±0.007	0.872±0.012
	DPC	**0.945±0.010**	0.835±0.053	0.887±0.025	0.893±0.020	0.782±0.044
	DDE	0.856±0.026	0.972±0.024	0.914±0.013	0.907±0.014	0.834±0.024
	QSOrder	0.896±0.012	0.976±0.007	0.935±0.007	0.932±0.008	0.873±0.014
	CTDC	0.898±0.010	0.961±0.021	0.928±0.009	0.925±0.009	0.858±0.020
	RPSSM	0.903±0.020	0.971±0.025	0.937±0.010	0.934±0.009	0.876±0.022
	TPC-PSSM	0.839±0.022	0.984±0.014	0.911±0.014	0.903±0.016	0.831±0.025
	RPM-PSSM	0.899±0.007	**0.988±0.010**	0.943±0.007	0.940±0.005	0.889±0.014
	DP-PSSM	0.913±0.017	0.979±0.012	**0.946±0.008**	**0.944±0.009**	**0.893±0.015**
	Pse-PSSM	0.899±0.012	0.982±0.011	0.940±0.007	0.937±0.007	0.883±0.015
RF	AAC	0.916±0.012	0.968±0.012	0.941±0.007	0.939±0.007	0.883±0.014
	DPC	0.918±0.014	**0.986±0.014**	0.952±0.007	0.950±0.007	0.905±0.015
	DDE	0.910±0.019	0.975±0.013	0.941±0.014	0.939±0.013	0.884±0.027
	QSOrder	0.910±0.012	0.979±0.014	0.944±0.007	0.942±0.007	0.890±0.015
	CTDC	0.925±0.008	0.959±0.012	0.941±0.009	0.940±0.008	0.882±0.018
	RPSSM	0.921±0.014	0.973±0.017	0.948±0.010	0.945±0.010	0.896±0.021
	TPC-PSSM	0.920±0.004	0.979±0.014	0.950±0.008	0.948±0.008	0.901±0.016
	RPM-PSSM	0.937±0.009	0.972±0.013	0.953±0.010	0.952±0.010	0.907±0.020
	DP-PSSM	**0.947±0.011**	0.969±0.017	**0.959±0.010**	**0.958±0.011**	**0.917±0.021**
	Pse-PSSM	0.938±0.010	0.974±0.015	0.956±0.012	0.954±0.012	0.912±0.024
LightGBM	AAC	0.910±0.012	0.942±0.014	0.924±0.009	0.922±0.010	0.850±0.019
	DPC	0.924±0.013	0.940±0.019	0.932±0.008	0.929±0.008	0.864±0.017
	DDE	0.923±0.017	0.961±0.022	0.943±0.015	0.941±0.015	0.886±0.029
	QSOrder	0.909±0.018	0.949±0.018	0.929±0.014	0.926±0.016	0.858±0.031
	CTDC	0.923±0.012	0.952±0.019	0.937±0.013	0.936±0.013	0.874±0.028
	RPSSM	0.919±0.014	0.962±0.015	0.940±0.011	0.939±0.012	0.881±0.023
	TPC-PSSM	0.920±0.014	**0.972±0.015**	0.946±0.011	0.944±0.011	0.893±0.022
	RPM-PSSM	0.938±0.015	0.967±0.013	0.952±0.010	0.951±0.010	0.905±0.021
	DP-PSSM	**0.945±0.010**	0.963±0.018	**0.953±0.010**	**0.953±0.011**	**0.906±0.021**
	Pse-PSSM	0.927±0.017	0.958±0.017	0.943±0.013	0.941±0.013	0.886±0.027

NB	AAC	**0.911±0.010**	0.961±0.010	0.934±0.008	0.932±0.008	0.871±0.015
	DPC	0.834±0.013	0.972±0.010	0.902±0.007	0.894±0.008	0.812±0.012
	DDE	0.810±0.019	**0.981±0.012**	0.895±0.013	0.883±0.016	0.801±0.025
	QSOrder	0.904±0.013	0.966±0.008	0.934±0.010	0.932±0.009	0.870±0.018
	CTDC	0.904±0.006	0.934±0.013	0.919±0.006	0.916±0.007	0.838±0.012
	RPSSM	0.893±0.045	0.744±0.103	0.819±0.038	0.830±0.027	0.653±0.066
	TPC-PSSM	0.891±0.027	0.953±0.017	0.921±0.016	0.918±0.017	0.843±0.031
	RPM-PSSM	0.841±0.012	0.979±0.011	0.910±0.007	0.902±0.008	0.828±0.012
	DP-PSSM	0.910±0.014	0.959±0.018	0.935±0.011	0.931±0.012	0.870±0.023
	Pse-PSSM	0.909±0.017	0.972±0.016	**0.941±0.013**	**0.939±0.014**	**0.883±0.027**
SVM	AAC	0.933±0.007	0.962±0.013	0.946±0.008	0.945±0.008	0.893±0.017
	DPC	0.878±0.015	0.962±0.017	0.919±0.011	0.915±0.012	0.841±0.023
	DDE	0.891±0.020	0.953±0.019	0.921±0.014	0.919±0.014	0.845±0.028
	QSOrder	0.943±0.006	0.966±0.014	0.954±0.009	0.954±0.009	0.909±0.017
	CTDC	0.935±0.008	0.958±0.012	0.945±0.007	0.944±0.008	0.892±0.014
	RPSSM	0.923±0.016	**0.967±0.014**	0.945±0.013	0.944±0.014	0.892±0.027
	TPC-PSSM	0.891±0.033	0.926±0.021	0.905±0.017	0.903±0.019	0.816±0.030
	RPM-PSSM	0.943±0.010	0.963±0.017	0.953±0.009	0.952±0.010	0.906±0.019
	DP-PSSM	**0.958±0.007**	0.965±0.015	**0.962±0.008**	**0.962±0.008**	**0.923±0.017**
	Pse-PSSM	0.949±0.007	0.965±0.011	0.957±0.008	0.956±0.009	0.914±0.017
XGBoost	AAC	0.911±0.011	0.944±0.021	0.927±0.009	0.925±0.008	0.854±0.018
	DPC	0.915±0.008	0.934±0.019	0.925±0.012	0.922±0.011	0.849±0.025
	DDE	0.921±0.015	0.961±0.016	0.941±0.014	0.939±0.014	0.883±0.027
	QSOrder	0.898±0.015	0.952±0.015	0.924±0.011	0.922±0.011	0.850±0.024
	CTDC	0.925±0.008	0.958±0.013	0.941±0.008	0.940±0.008	0.882±0.018
	RPSSM	0.916±0.013	0.964±0.015	0.939±0.010	0.937±0.010	0.879±0.020
	TPC-PSSM	0.922±0.009	0.973±0.011	0.948±0.005	0.946±0.006	0.897±0.011
	RPM-PSSM	**0.943±0.010**	**0.975±0.012**	**0.959±0.009**	**0.958±0.009**	**0.918±0.019**
	DP-PSSM	0.929±0.013	0.959±0.016	0.944±0.012	0.943±0.011	0.887±0.024
	Pse-PSSM	0.921±0.015	0.959±0.015	0.940±0.013	0.937±0.013	0.880±0.026

Note: Values are expressed as mean±standard deviation. The best performance value for each metric across different encoding methods within the same machine learning algorithm is highlighted in bold.

Table S5. Performance comparison of single method-based models for predicting T1SE based on independent test.

Model	Encoding	SN	SP	ACC	F-value	MCC
KNN	AAC	**0.966**	0.966	0.966	0.966	0.931
	DPC	**0.966**	0.897	0.931	0.933	0.864
	DDE	**0.966**	0.966	0.966	0.966	0.931
	QSOrder	**0.966**	0.966	0.966	0.966	0.931
	CTDC	**0.966**	0.931	0.948	0.949	0.897
	RPSSM	**0.966**	**1**	**0.983**	0.982	0.966
	TPC-PSSM	0.931	0.966	0.948	0.947	0.897
	RPM-PSSM	**0.966**	0.966	0.966	0.966	0.931
	DP-PSSM	**0.966**	0.966	0.966	0.966	0.931
	Pse-PSSM	**0.966**	**1**	**0.983**	**0.982**	**0.966**
RF	AAC	**0.966**	0.966	0.966	0.966	0.931
	DPC	**0.966**	0.966	0.966	0.966	0.931
	DDE	0.931	0.966	0.948	0.947	0.897
	QSOrder	**0.966**	**1**	0.983	**0.982**	**0.966**
	CTDC	**0.966**	0.931	0.948	0.949	0.897
	RPSSM	**0.966**	0.966	0.966	0.966	0.931
	TPC-PSSM	**0.966**	**1**	**0.983**	**0.982**	**0.966**
	RPM-PSSM	**0.966**	**1**	**0.983**	**0.982**	**0.966**
	DP-PSSM	**0.966**	**1**	**0.983**	**0.982**	**0.966**
	Pse-PSSM	**0.966**	**1**	**0.983**	**0.982**	**0.966**
LightGBM	AAC	**0.966**	0.931	0.948	0.949	0.897
	DPC	**0.966**	0.897	0.931	0.933	0.864
	DDE	0.931	0.966	0.948	0.947	0.897
	QSOrder	**0.966**	**1**	0.983	**0.982**	**0.966**
	CTDC	**0.966**	0.931	0.948	0.949	0.897
	RPSSM	**0.966**	0.966	0.966	0.966	0.931
	TPC-PSSM	**0.966**	**1**	**0.983**	0.982	**0.966**
	RPM-PSSM	**0.966**	0.966	0.966	0.966	0.931
	DP-PSSM	**0.966**	0.931	0.948	0.949	0.897
	Pse-PSSM	**0.966**	**1**	**0.983**	**0.982**	0.966

NB	AAC	**0.966**	0.966	0.966	0.966	0.931
	DPC	**0.966**	**1**	**0.983**	**0.982**	**0.966**
	DDE	0.931	0.966	0.948	0.947	0.897
	QSOrder	**0.966**	0.966	0.966	0.966	0.931
	CTDC	**0.966**	0.897	0.931	0.933	0.864
	RPSSM	**0.966**	0.862	0.914	0.918	0.832
	TPC-PSSM	0.931	0.966	0.948	0.947	0.897
	RPM-PSSM	0.897	**1**	0.948	0.945	0.901
	DP-PSSM	**0.966**	**1**	**0.983**	**0.982**	**0.966**
	Pse-PSSM	**0.966**	**1**	**0.983**	**0.982**	**0.966**
SVM	AAC	**0.966**	0.931	0.948	0.949	0.897
	DPC	**0.966**	0.931	0.948	0.949	0.897
	DDE	**0.966**	0.931	0.948	0.949	0.897
	QSOrder	**0.966**	0.966	0.966	0.966	0.931
	CTDC	**0.966**	0.931	0.948	0.949	0.897
	RPSSM	**0.966**	**1**	**0.983**	0.982	**0.966**
	TPC-PSSM	**0.966**	0.966	0.966	0.966	0.931
	RPM-PSSM	**0.966**	**1**	**0.983**	**0.982**	**0.966**
	DP-PSSM	**0.966**	0.966	0.966	0.966	0.931
	Pse-PSSM	**0.966**	**1**	**0.983**	**0.982**	**0.966**
XGBoost	AAC	**0.966**	0.931	0.948	0.949	0.897
	DPC	**0.966**	0.897	0.931	0.933	0.864
	DDE	**0.966**	0.966	0.966	0.966	0.931
	QSOrder	0.931	0.862	0.897	0.9	0.795
	CTDC	**0.966**	0.931	0.948	0.949	0.897
	RPSSM	**0.966**	0.966	0.966	0.966	0.931
	TPC-PSSM	**0.966**	**1**	**0.983**	**0.982**	**0.966**
	RPM-PSSM	**0.966**	0.966	0.966	0.966	0.931
	DP-PSSM	**0.966**	**1**	**0.983**	**0.982**	**0.966**
	Pse-PSSM	**0.966**	**1**	**0.983**	**0.982**	**0.966**

Note: The best performance value for each metric across different encoding methods in each model is highlighted in bold.

Table S6. Performance comparison of single-method-based models for predicting type II secreted substrates based on the 5-fold cross-validation test.

Model	Encoding	SN	SP	ACC	F-value	MCC
KNN	DDE	0.814±0.074	0.880±0.082	0.836±0.026	0.845±0.022	0.704±0.046
	DPC	0.856±0.095	0.717±0.135	0.784±0.044	0.796±0.041	0.595±0.075
	DP	0.878±0.074	0.796±0.051	0.836±0.023	0.839±0.029	0.682±0.050
	QSOrder	0.857±0.048	0.837±0.039	0.845±0.016	0.844±0.019	0.699±0.035
	PDT	0.865±0.050	0.714±0.074	0.788±0.030	0.801±0.023	0.591±0.051
	AAC-PSSM	0.863±0.055	0.857±0.054	0.860±0.020	0.858±0.023	0.725±0.037
	TPC-PSSM	0.697±0.067	0.907±0.039	0.801±0.033	0.770±0.045	0.624±0.066
	RPM-PSSM	**0.904±0.031**	0.839±0.045	**0.870±0.016**	**0.874±0.012**	**0.748±0.031**
	DP-PSSM	0.845±0.043	0.815±0.055	0.828±0.021	0.827±0.023	0.664±0.041
	Pse-PSSM	0.863±0.043	0.870±0.047	0.863±0.024	0.861±0.025	0.734±0.044
RF	DDE	0.803±0.037	0.788±0.036	0.794±0.030	0.792±0.031	0.600±0.058
	DPC	0.788±0.027	0.878±0.038	0.832±0.022	0.820±0.022	0.674±0.046
	DP	0.796±0.020	0.892±0.020	0.841±0.014	0.829±0.015	0.693±0.027
	QSOrder	0.836±0.035	0.856±0.039	0.845±0.027	0.843±0.028	0.699±0.054
	PDT	0.823±0.033	0.860±0.035	0.840±0.022	0.834±0.023	0.687±0.047
	AAC-PSSM	0.828±0.029	0.899±0.028	0.865±0.017	0.854±0.019	0.738±0.031
	TPC-PSSM	0.832±0.022	0.898±0.034	0.864±0.016	0.857±0.013	0.734±0.030
	RPM-PSSM	0.824±0.033	0.904±0.039	0.862±0.027	0.854±0.026	0.734±0.053
	DP-PSSM	0.841±0.025	0.888±0.046	0.861±0.024	0.856±0.024	0.733±0.049
	Pse-PSSM	**0.859±0.036**	**0.909±0.022**	**0.883±0.019**	**0.879±0.021**	**0.771±0.040**
LightGBM	DDE	0.772±0.062	0.759±0.065	0.766±0.054	0.764±0.058	0.537±0.102
	DPC	0.769±0.045	0.794±0.044	0.783±0.036	0.775±0.039	0.572±0.072
	DP	0.787±0.054	0.818±0.030	0.802±0.032	0.793±0.039	0.609±0.064
	QSOrder	0.817±0.060	0.809±0.038	0.814±0.046	0.812±0.053	0.632±0.096
	PDT	0.834±0.044	0.842±0.045	0.839±0.035	0.833±0.039	0.681±0.073
	AAC-PSSM	0.843±0.045	**0.879±0.031**	0.862±0.024	**0.855±0.026**	**0.729±0.049**
	TPC-PSSM	0.819±0.029	0.863±0.034	0.840±0.024	0.833±0.023	0.685±0.051
	RPM-PSSM	0.811±0.029	0.825±0.049	0.815±0.037	0.811±0.035	0.639±0.072
	DP-PSSM	0.843±0.061	0.859±0.031	0.850±0.040	0.845±0.044	0.707±0.076
	Pse-PSSM	**0.847±0.047**	0.872±0.037	**0.860±0.030**	0.854±0.033	0.727±0.063

NB	DDE	0.802±0.054	0.820±0.061	0.812±0.047	0.805±0.046	0.630±0.093
	DPC	0.771±0.041	0.753±0.050	0.764±0.035	0.759±0.035	0.532±0.070
	DP	0.782±0.019	0.837±0.040	0.809±0.019	0.801±0.018	0.625±0.037
	QSOrder	0.807±0.030	0.834±0.048	0.820±0.027	0.814±0.027	0.649±0.050
	PDT	0.775±0.034	0.794±0.041	0.785±0.030	0.778±0.029	0.571±0.060
	AAC-PSSM	0.847±0.030	0.617±0.054	0.733±0.025	0.757±0.023	0.481±0.047
	TPC-PSSM	0.776±0.142	0.681±0.189	0.726±0.053	0.732±0.055	0.480±0.100
	RPM-PSSM	0.823±0.029	0.889±0.029	0.856±0.022	0.846±0.027	0.717±0.047
	DP-PSSM	0.817±0.042	0.875±0.032	0.845±0.029	0.835±0.035	0.695±0.063
	Pse-PSSM	**0.870±0.032**	**0.916±0.015**	**0.893±0.018**	**0.890±0.021**	**0.790±0.036**
SVM	DDE	0.829±0.030	0.868±0.044	0.848±0.029	0.843±0.029	0.701±0.059
	DPC	0.807±0.027	0.869±0.023	0.839±0.019	0.829±0.020	0.684±0.032
	DP	0.812±0.034	0.871±0.030	0.842±0.022	0.832±0.025	0.693±0.040
	QSOrder	0.815±0.030	0.877±0.031	0.846±0.020	0.836±0.023	0.698±0.037
	PDT	0.837±0.027	0.885±0.030	0.859±0.025	0.854±0.027	0.726±0.049
	AAC-PSSM	0.929±0.018	**0.919±0.026**	0.922±0.019	0.923±0.017	0.849±0.037
	TPC-PSSM	0.835±0.035	0.839±0.048	0.837±0.030	0.835±0.028	0.679±0.064
	RPM-PSSM	0.859±0.026	0.911±0.025	0.884±0.018	0.879±0.020	0.775±0.037
	DP-PSSM	0.876±0.036	0.905±0.037	0.889±0.032	0.886±0.034	0.784±0.064
	Pse-PSSM	**0.900±0.017**	0.915±0.032	**0.907±0.017**	**0.906±0.016**	**0.818±0.035**
XGBoost	DDE	0.688±0.080	0.713±0.081	0.698±0.074	0.687±0.083	0.404±0.146
	DPC	0.679±0.112	0.740±0.089	0.704±0.095	0.683±0.113	0.426±0.181
	DP	0.706±0.082	0.747±0.106	0.726±0.083	0.711±0.087	0.459±0.175
	QSOrder	0.748±0.062	0.781±0.049	0.760±0.050	0.748±0.060	0.535±0.097
	PDT	0.794±0.048	0.790±0.074	0.790±0.053	0.787±0.053	0.586±0.106
	AAC-PSSM	0.756±0.085	0.837±0.088	0.799±0.071	0.778±0.081	0.605±0.145
	TPC-PSSM	0.719±0.112	0.805±0.107	0.763±0.091	0.742±0.105	0.537±0.183
	RPM-PSSM	0.742±0.071	0.805±0.101	0.770±0.076	0.760±0.076	0.551±0.149
	DP-PSSM	0.770±0.096	**0.836±0.072**	0.799±0.066	0.786±0.076	0.615±0.120
	Pse-PSSM	**0.802±0.062**	0.834±0.068	**0.817±0.053**	**0.812±0.056**	**0.640±0.107**

Note: Values are expressed as mean±standard deviation. The best performance value for each metric across different encoding methods within the same machine learning algorithm is highlighted in bold.

Table S7. Performance comparison of single-method-based models for predicting type II secreted substrates based on independent test.

Model	Encoding	SN	SP	ACC	F-value	MCC
KNN	DDE	0.882	**0.941**	**0.912**	0.909	**0.825**
	DPC	0.941	0.647	0.794	0.821	0.615
	DP	**1**	0.765	0.882	0.895	0.787
	QSOrder	0.941	0.588	0.765	0.8	0.566
	PDT	**1**	0.824	**0.912**	**0.919**	0.837
	AAC-PSSM	0.941	0.765	0.853	0.865	0.717
	TPC-PSSM	0.765	**0.941**	0.853	0.839	0.717
	RPM-PSSM	0.941	0.824	0.882	0.889	0.77
	DP-PSSM	0.882	0.706	0.794	0.811	0.598
	Pse-PSSM	0.882	**0.941**	**0.912**	0.909	**0.825**
RF	DDE	**0.941**	**0.941**	**0.941**	**0.941**	**0.882**
	DPC	0.882	0.882	0.882	0.882	0.765
	DP	0.882	0.824	0.853	0.857	0.707
	QSOrder	0.824	0.765	0.794	0.8	0.589
	PDT	0.882	0.882	0.882	0.882	0.765
	AAC-PSSM	0.765	0.882	0.824	0.812	0.652
	TPC-PSSM	0.765	**0.941**	0.853	0.839	0.717
	RPM-PSSM	0.824	0.882	0.853	0.848	0.707
	DP-PSSM	**0.941**	0.824	0.882	0.889	0.77
	Pse-PSSM	**0.941**	0.882	0.912	0.914	0.825
LightGBM	DDE	0.882	**1**	**0.941**	**0.938**	**0.889**
	DPC	**0.941**	0.882	0.912	0.914	0.825
	DP	**0.941**	0.824	0.882	0.889	0.77
	QSOrder	0.882	0.765	0.824	0.833	0.652
	PDT	0.882	0.824	0.853	0.857	0.707
	AAC-PSSM	0.765	0.882	0.824	0.812	0.652
	TPC-PSSM	0.706	0.941	0.824	0.8	0.666
	RPM-PSSM	0.882	0.882	0.882	0.882	0.765
	DP-PSSM	**0.941**	0.882	0.912	0.914	0.825
	Pse-PSSM	**0.941**	0.824	0.882	0.889	0.77
NB	DDE	0.824	**0.882**	0.853	0.848	0.707
	DPC	**0.941**	0.941	0.941	0.941	**0.882**
	DP	**0.941**	0.824	**0.882**	**0.889**	0.77
	QSOrder	0.765	0.824	0.794	0.788	0.589
	PDT	**0.941**	0.765	0.853	0.865	0.717
	AAC-PSSM	0.882	0.412	0.647	0.714	0.333
	TPC-PSSM	**0.941**	0.824	**0.882**	**0.889**	0.77
	RPM-PSSM	0.882	0.824	0.853	0.857	0.707
	DP-PSSM	**0.941**	0.824	**0.882**	**0.889**	0.77
	Pse-PSSM	0.824	**0.882**	0.853	0.848	0.707

SVM	DDE	0.882	**0.941**	**0.912**	0.909	**0.825**
	DPC	0.882	**0.941**	**0.912**	0.909	**0.825**
	DP	0.882	0.824	0.853	0.857	0.707
	QSOrder	0.882	0.765	0.824	0.833	0.652
	PDT	**0.941**	0.882	**0.912**	**0.914**	**0.825**
	AAC-PSSM	0.882	0.882	0.882	0.882	0.765
	TPC-PSSM	0.882	0.882	0.882	0.882	0.765
	RPM-PSSM	**0.941**	0.824	0.882	0.889	0.77
	DP-PSSM	**0.941**	0.824	0.882	0.889	0.77
	Pse-PSSM	**0.941**	0.882	**0.912**	**0.914**	**0.825**
XGBoost	DDE	0.882	**0.941**	**0.912**	0.909	**0.825**
	DPC	0.824	0.882	0.853	0.848	0.707
	DP	**0.941**	0.882	**0.912**	**0.914**	**0.825**
	QSOrder	0.824	0.824	0.824	0.824	0.647
	PDT	0.882	0.824	0.853	0.857	0.707
	AAC-PSSM	0.824	0.824	0.824	0.824	0.647
	TPC-PSSM	0.765	0.882	0.824	0.812	0.652
	RPM-PSSM	0.882	0.882	0.882	0.882	0.765
	DP-PSSM	**0.941**	0.882	**0.912**	**0.914**	**0.825**
	Pse-PSSM	0.882	0.824	0.853	0.857	0.707

Note: The best performance value for each metric across different encoding methods within the same machine learning algorithm is highlighted in bold.

Table S8. Performance comparison of single-method-based models for predicting type III secreted substrates based on the 5-fold cross-validation test.

Model	Encoding	SN	SP	ACC	F-value	MCC
KNN	AAC	0.816±0.022	0.876±0.017	0.845±0.008	0.840±0.010	0.693±0.016
	DP	0.750±0.037	0.884±0.019	0.817±0.014	0.802±0.019	0.641±0.026
	DPC	0.747±0.035	0.776±0.037	0.761±0.010	0.756±0.013	0.526±0.019
	QSOrder	0.819±0.020	0.876±0.014	0.847±0.007	0.842±0.009	0.697±0.013
	CTDC	0.777±0.019	0.858±0.020	0.817±0.007	0.809±0.008	0.638±0.014
	AAC-PSSM	0.907±0.009	0.886±0.019	0.896±0.009	0.897±0.008	0.793±0.018
	TPC-PSSM	0.815±0.032	0.883±0.024	0.849±0.011	0.842±0.014	0.700±0.021
	AB-PSSM	**0.946±0.007**	0.739±0.016	0.843±0.007	0.857±0.006	0.700±0.012
	DP-PSSM	0.913±0.018	0.874±0.020	0.893±0.009	0.895±0.009	0.787±0.018
	Pse-PSSM	0.914±0.013	**0.897±0.015**	**0.905±0.008**	**0.906±0.008**	**0.811±0.015**
RF	AAC	0.856±0.017	0.860±0.011	0.858±0.008	0.857±0.009	0.717±0.016
	DP	0.819±0.011	0.886±0.009	0.852±0.008	0.846±0.008	0.706±0.014
	DPC	0.774±0.017	0.886±0.016	0.829±0.010	0.818±0.010	0.664±0.019
	QSOrder	0.850±0.013	0.862±0.009	0.856±0.007	0.854±0.008	0.712±0.015
	CTDC	0.823±0.010	0.846±0.013	0.834±0.010	0.832±0.010	0.670±0.019
	AAC-PSSM	0.883±0.008	0.923±0.011	0.903±0.008	0.901±0.008	0.806±0.016
	TPC-PSSM	0.875±0.009	0.902±0.011	0.888±0.005	0.887±0.005	0.777±0.010
	AB-PSSM	0.884±0.006	**0.937±0.009**	0.910±0.005	0.907±0.006	**0.822±0.011**
	DP-PSSM	**0.919±0.005**	0.903±0.009	**0.911±0.004**	**0.912±0.004**	**0.822±0.009**
	Pse-PSSM	0.914±0.008	0.902±0.008	0.907±0.007	0.908±0.006	0.815±0.014
LightGBM	AAC	0.842±0.011	0.851±0.012	0.847±0.008	0.846±0.008	0.694±0.014
	DP	0.828±0.010	0.858±0.011	0.843±0.007	0.840±0.007	0.687±0.013
	DPC	0.807±0.017	0.838±0.014	0.822±0.012	0.818±0.012	0.646±0.024
	QSOrder	0.844±0.011	0.859±0.010	0.851±0.005	0.849±0.006	0.702±0.012
	CTDC	0.822±0.012	0.846±0.015	0.834±0.011	0.831±0.012	0.668±0.023
	AAC-PSSM	0.898±0.010	0.911±0.012	0.904±0.008	0.903±0.008	0.809±0.016
	TPC-PSSM	0.890±0.008	0.906±0.011	0.898±0.007	0.897±0.007	0.796±0.014
	AB-PSSM	0.898±0.007	**0.923±0.009**	0.910±0.005	0.909±0.005	0.821±0.010
	DP-PSSM	**0.917±0.009**	0.914±0.010	**0.916±0.008**	**0.916±0.008**	**0.831±0.016**
	Pse-PSSM	0.912±0.010	0.913±0.012	0.912±0.009	0.912±0.009	0.825±0.017

NB	AAC	0.790±0.016	0.828±0.010	0.809±0.007	0.805±0.008	0.619±0.013
	DP	0.789±0.012	0.819±0.018	0.804±0.012	0.800±0.011	0.609±0.024
	DPC	0.764±0.011	0.694±0.025	0.729±0.014	0.737±0.011	0.460±0.028
	QSOrder	0.743±0.013	0.841±0.009	0.792±0.006	0.780±0.008	0.587±0.013
	CTDC	0.739±0.009	0.777±0.012	0.758±0.006	0.753±0.006	0.517±0.013
	AAC-PSSM	**0.886±0.005**	0.676±0.016	0.781±0.007	0.801±0.005	0.575±0.012
	TPC-PSSM	0.505±0.045	0.932±0.013	0.718±0.020	0.638±0.037	0.484±0.034
	AB-PSSM	0.867±0.004	0.789±0.014	0.828±0.007	0.834±0.006	0.658±0.012
	DP-PSSM	0.792±0.011	**0.880±0.011**	0.836±0.009	0.828±0.010	0.675±0.019
	Pse-PSSM	0.908±0.005	0.835±0.010	**0.871±0.004**	**0.875±0.004**	**0.744±0.008**
SVM	AAC	0.877±0.007	0.876±0.012	0.876±0.007	0.876±0.006	0.753±0.013
	DP	0.857±0.011	0.867±0.010	0.861±0.009	0.860±0.009	0.724±0.017
	DPC	0.838±0.010	0.871±0.017	0.854±0.012	0.851±0.012	0.710±0.024
	QSOrder	0.866±0.012	0.866±0.009	0.866±0.008	0.865±0.009	0.732±0.017
	CTDC	0.851±0.016	0.872±0.008	0.861±0.008	0.859±0.009	0.723±0.015
	AAC-PSSM	0.916±0.014	0.915±0.016	0.915±0.006	0.915±0.006	0.831±0.013
	TPC-PSSM	0.881±0.023	0.859±0.020	0.870±0.010	0.871±0.011	0.741±0.020
	AB-PSSM	0.913±0.009	0.923±0.017	0.918±0.011	0.918±0.011	0.837±0.021
	DP-PSSM	0.924±0.010	**0.927±0.010**	**0.925±0.009**	**0.925±0.009**	**0.851±0.017**
	Pse-PSSM	**0.927±0.006**	0.919±0.008	0.923±0.007	0.923±0.007	0.846±0.013
XGBoost	AAC	0.844±0.016	0.855±0.010	0.850±0.010	0.848±0.011	0.700±0.020
	DP	0.832±0.008	0.858±0.011	0.845±0.007	0.842±0.007	0.691±0.014
	DPC	0.810±0.022	0.849±0.018	0.829±0.018	0.825±0.019	0.660±0.036
	QSOrder	0.850±0.010	0.855±0.009	0.852±0.006	0.851±0.007	0.704±0.013
	CTDC	0.825±0.015	0.848±0.015	0.836±0.011	0.834±0.011	0.673±0.022
	AAC-PSSM	0.892±0.007	0.912±0.012	0.901±0.007	0.900±0.007	0.803±0.015
	TPC-PSSM	0.884±0.010	0.900±0.009	0.892±0.007	0.891±0.007	0.784±0.014
	AB-PSSM	0.900±0.013	**0.921±0.013**	0.910±0.009	0.909±0.009	0.821±0.017
	DP-PSSM	**0.914±0.012**	0.912±0.016	**0.913±0.012**	**0.913±0.012**	**0.826±0.025**
	Pse-PSSM	0.911±0.011	0.904±0.011	0.907±0.008	0.907±0.008	0.814±0.016

Note: Values are expressed as mean±standard deviation. The best performance value for each metric across different encoding methods within the same machine learning algorithm is highlighted in bold.

Table S9. Performance comparison of single-method-based models for predicting type III secreted substrates based on independent test.

Model	Encoding	SN	SP	ACC	F-value	MCC
KNN	AAC	0.851	0.713	0.796	0.782	0.569
	DP	0.84	0.755	0.806	0.798	0.598
	DPC	0.83	0.5	0.712	0.665	0.349
	QSOrder	0.851	0.734	0.804	0.793	0.589
	CTDC	0.84	0.862	0.849	0.851	0.702
	AAC-PSSM	0.947	**0.894**	**0.922**	**0.92**	**0.842**
	TPC-PSSM	0.862	**0.894**	0.876	0.878	0.756
	AB-PSSM	**0.957**	0.734	0.861	0.846	0.709
	DP-PSSM	**0.957**	0.862	0.914	0.91	0.823
	Pse-PSSM	0.947	0.862	0.908	0.904	0.811
RF	AAC	0.904	0.819	0.867	0.862	0.726
	DP	0.872	0.851	0.863	0.862	0.724
	DPC	0.883	0.755	0.83	0.819	0.644
	QSOrder	0.883	0.83	0.86	0.856	0.714
	CTDC	0.883	0.894	0.888	0.888	0.777
	AAC-PSSM	0.915	0.926	0.92	0.92	0.84
	TPC-PSSM	0.904	0.904	0.904	0.904	0.809
	AB-PSSM	**0.947**	**0.968**	**0.957**	**0.957**	**0.915**
	DP-PSSM	0.936	0.883	0.912	0.91	0.82
	Pse-PSSM	0.915	0.883	0.901	0.899	0.798
LightGBM	AAC	0.904	0.809	0.863	0.856	0.716
	DP	0.883	0.83	0.86	0.856	0.714
	DPC	0.883	0.734	0.822	0.809	0.624
	QSOrder	0.872	0.851	0.863	0.862	0.724
	CTDC	0.894	0.872	0.884	0.883	0.766
	AAC-PSSM	0.947	0.894	0.922	0.92	0.842
	TPC-PSSM	0.936	0.915	0.926	0.926	0.851
	AB-PSSM	**0.947**	**0.947**	**0.947**	**0.947**	**0.894**
	DP-PSSM	0.947	0.894	0.922	0.92	0.842
	Pse-PSSM	0.926	0.894	0.911	0.91	0.82

NB	AAC	0.851	0.84	0.847	0.846	0.692
	DP	0.851	0.755	0.812	0.803	0.609
	DPC	0.798	0.702	0.761	0.75	0.502
	QSOrder	0.809	0.809	0.809	0.809	0.617
	CTDC	0.819	0.766	0.798	0.793	0.586
	AAC-PSSM	**0.904**	0.745	0.837	0.824	0.657
	TPC-PSSM	0.468	**0.926**	0.607	0.697	0.443
	AB-PSSM	0.883	0.851	**0.869**	**0.867**	**0.734**
	DP-PSSM	0.809	0.755	0.788	0.782	0.565
	Pse-PSSM	0.926	0.787	0.866	0.856	0.72
SVM	AAC	0.872	0.851	0.863	0.862	0.724
	DP	0.904	0.83	0.872	0.867	0.736
	DPC	0.926	0.766	0.857	0.846	0.7
	QSOrder	0.894	0.809	0.857	0.851	0.705
	CTDC	0.883	0.872	0.878	0.878	0.755
	AAC-PSSM	0.947	0.894	0.922	0.92	0.842
	TPC-PSSM	0.915	0.883	0.901	0.899	0.798
	AB-PSSM	0.947	**0.926**	**0.937**	**0.936**	**0.873**
	DP-PSSM	**0.957**	0.862	0.914	0.91	0.823
	Pse-PSSM	0.936	0.894	0.917	0.915	0.831
XGBoost	AAC	0.894	0.84	0.87	0.867	0.735
	DP	0.894	0.872	0.884	0.883	0.766
	DPC	0.883	0.755	0.83	0.819	0.644
	QSOrder	0.883	0.819	0.856	0.851	0.704
	CTDC	0.894	0.862	0.88	0.878	0.756
	AAC-PSSM	0.936	0.851	0.898	0.894	0.79
	TPC-PSSM	0.926	0.915	0.921	0.92	0.84
	AB-PSSM	**0.947**	**0.926**	**0.937**	**0.936**	**0.873**
	DP-PSSM	**0.947**	0.904	0.927	0.926	0.852
	Pse-PSSM	0.926	0.872	0.902	0.899	0.799

Note: The best performance value for each metric across different encoding methods within the same machine learning algorithm is highlighted in bold.

Table S10. Performance comparison of single-method-based models for predicting type IV secreted substrates based on the 5-fold cross-validation test.

Model	Encoding	SN	SP	ACC	F-value	MCC
KNN	AAC	0.803±0.021	0.795±0.030	0.799±0.015	0.799±0.013	0.599±0.028
	DPC	0.867±0.021	0.636±0.061	0.751±0.024	0.777±0.015	0.519±0.040
	PDT	0.790±0.025	0.755±0.036	0.771±0.011	0.775±0.009	0.546±0.022
	QSOrder	0.793±0.009	0.800±0.034	0.796±0.015	0.795±0.012	0.594±0.031
	CTDC	0.786±0.019	0.785±0.023	0.785±0.012	0.785±0.012	0.572±0.024
	AB-PSSM	**0.916±0.015**	0.622±0.053	0.769±0.021	0.798±0.014	0.563±0.032
	EEDP	0.848±0.021	**0.903±0.026**	**0.875±0.008**	**0.871±0.008**	**0.753±0.017**
	DP-PSSM	0.846±0.021	0.900±0.016	0.873±0.008	0.869±0.009	0.747±0.015
	PSSM-composition	0.879±0.019	0.826±0.031	0.853±0.012	0.856±0.010	0.707±0.023
	Pse-PSSM	0.855±0.012	0.888±0.020	0.871±0.011	0.868±0.011	0.743±0.022
RF	AAC	0.784±0.008	0.818±0.022	0.800±0.012	0.797±0.011	0.602±0.025
	DPC	0.769±0.013	0.826±0.016	0.797±0.012	0.791±0.012	0.596±0.025
	PDT	0.782±0.020	0.819±0.014	0.800±0.012	0.795±0.014	0.602±0.025
	QSOrder	0.782±0.012	0.821±0.022	0.801±0.014	0.797±0.014	0.603±0.029
	CTDC	0.788±0.015	0.819±0.016	0.803±0.011	0.800±0.012	0.608±0.023
	AB-PSSM	0.842±0.008	0.917±0.011	0.879±0.007	0.874±0.007	0.761±0.014
	EEDP	0.832±0.007	0.929±0.011	0.881±0.007	0.874±0.007	0.765±0.014
	DP-PSSM	**0.849±0.005**	0.914±0.009	0.882±0.005	0.877±0.005	0.765±0.011
	PSSM-composition	0.825±0.006	**0.933±0.009**	0.879±0.003	0.871±0.003	0.762±0.006
	Pse-PSSM	0.850±0.012	0.921±0.012	**0.885±0.008**	**0.880±0.008**	**0.773±0.016**
LightGBM	AAC	0.794±0.017	0.801±0.014	0.797±0.011	0.795±0.011	0.595±0.021
	DPC	0.776±0.021	0.795±0.024	0.785±0.019	0.783±0.019	0.571±0.037
	PDT	0.806±0.004	0.804±0.019	0.804±0.010	0.804±0.009	0.610±0.021
	QSOrder	0.800±0.018	0.799±0.018	0.799±0.015	0.799±0.015	0.600±0.029
	CTDC	0.781±0.019	0.801±0.028	0.790±0.020	0.788±0.018	0.584±0.039
	AB-PSSM	**0.866±0.010**	0.883±0.016	0.875±0.009	0.873±0.009	0.750±0.018
	EEDP	0.848±0.009	0.904±0.012	0.876±0.007	0.872±0.008	0.753±0.015
	DP-PSSM	0.859±0.014	0.913±0.005	**0.886±0.008**	**0.882±0.009**	**0.773±0.014**
	PSSM-composition	0.848±0.008	**0.915±0.016**	0.881±0.009	0.877±0.008	0.765±0.018
	Pse-PSSM	**0.866±0.005**	0.900±0.015	0.883±0.009	0.880±0.009	0.767±0.018

NB	AAC	0.821±0.007	0.728±0.014	0.774±0.010	0.784±0.008	0.552±0.018
	DPC	0.793±0.015	0.685±0.015	0.738±0.011	0.751±0.011	0.480±0.022
	PDT	0.763±0.007	0.704±0.019	0.733±0.011	0.741±0.009	0.468±0.022
	QSOrder	0.827±0.007	0.707±0.018	0.767±0.011	0.780±0.009	0.538±0.020
	CTDC	0.794±0.009	0.674±0.024	0.734±0.014	0.749±0.012	0.472±0.028
	AB-PSSM	0.807±0.005	0.715±0.012	0.762±0.007	0.771±0.006	0.525±0.012
	EEDP	0.825±0.007	0.479±0.024	0.653±0.010	0.703±0.005	0.324±0.018
	DP-PSSM	0.810±0.009	**0.875±0.008**	**0.842±0.007**	**0.836±0.008**	**0.686±0.014**
	PSSM-composition	0.825±0.007	0.678±0.030	0.752±0.018	0.769±0.014	0.510±0.034
	Pse-PSSM	**0.847±0.006**	0.798±0.015	0.823±0.009	0.827±0.008	0.646±0.017
SVM	AAC	0.813±0.010	0.837±0.019	0.824±0.010	0.822±0.009	0.651±0.020
	DPC	0.794±0.023	0.831±0.012	0.812±0.013	0.808±0.014	0.626±0.024
	PDT	0.807±0.016	0.832±0.021	0.818±0.010	0.816±0.010	0.639±0.020
	QSOrder	0.815±0.015	0.826±0.014	0.820±0.010	0.819±0.011	0.641±0.020
	CTDC	0.803±0.011	0.803±0.018	0.803±0.012	0.803±0.011	0.607±0.025
	AB-PSSM	0.869±0.011	0.895±0.016	0.882±0.011	0.879±0.012	0.764±0.022
	EEDP	0.848±0.014	**0.927±0.020**	0.887±0.013	0.882±0.013	0.777±0.026
	DP-PSSM	**0.895±0.012**	0.906±0.014	**0.900±0.008**	**0.899±0.008**	**0.801±0.015**
	PSSM-composition	0.866±0.014	0.912±0.017	0.889±0.008	0.886±0.008	0.779±0.017
	Pse-PSSM	0.870±0.009	0.913±0.011	0.892±0.007	0.889±0.007	0.785±0.014
XGBoost	AAC	0.789±0.018	0.804±0.018	0.796±0.015	0.794±0.016	0.593±0.031
	DPC	0.791±0.016	0.807±0.016	0.799±0.011	0.797±0.011	0.599±0.021
	PDT	0.798±0.015	0.819±0.023	0.807±0.016	0.805±0.016	0.617±0.033
	QSOrder	0.797±0.015	0.804±0.022	0.800±0.016	0.798±0.016	0.601±0.032
	CTDC	0.781±0.016	0.815±0.021	0.797±0.015	0.793±0.015	0.597±0.031
	AB-PSSM	**0.860±0.010**	0.883±0.013	0.872±0.010	0.870±0.010	0.743±0.020
	EEDP	0.822±0.043	0.908±0.017	0.864±0.022	0.857±0.026	0.733±0.041
	DP-PSSM	0.849±0.015	0.903±0.016	0.876±0.014	0.872±0.014	0.753±0.027
	PSSM-composition	0.846±0.009	**0.915±0.010**	**0.880±0.006**	**0.876±0.006**	**0.763±0.012**
	Pse-PSSM	0.854±0.007	0.904±0.014	0.879±0.008	**0.876±0.009**	0.760±0.017

Note: Values are expressed as mean±standard deviation. The best performance value for each metric across different encoding methods within the same machine learning algorithm is highlighted in bold.

Table S11. Performance comparison of single-method-based models for predicting type IV secreted substrates based on independent test.

Model	Encoding	SN	SP	ACC	F-value	MCC
KNN	AAC	0.893	0.827	0.865	0.86	0.722
	DPC	**0.973**	0.747	0.874	0.86	0.739
	PDT	0.867	0.827	0.85	0.847	0.694
	QSOrder	0.92	**0.92**	0.92	0.92	0.84
	CTDC	0.867	0.88	0.872	0.873	0.747
	AB-PSSM	1	0.6	0.833	0.8	0.655
	EEDP	0.933	0.893	0.915	0.913	0.827
	DP-PSSM	**0.973**	0.88	**0.93**	**0.927**	**0.857**
	PSSM-composition	0.987	0.853	0.925	0.92	0.848
	Pse-PSSM	0.96	0.853	0.911	0.907	0.818
RF	AAC	0.933	0.907	0.921	0.92	0.84
	DPC	0.907	0.933	0.919	0.92	0.84
	PDT	0.893	0.907	0.899	0.9	0.8
	QSOrder	0.907	**0.947**	0.925	0.927	0.854
	CTDC	0.907	0.907	0.907	0.907	0.813
	AB-PSSM	0.947	0.907	0.928	0.927	0.854
	EEDP	0.92	0.92	0.92	0.92	0.84
	DP-PSSM	**0.96**	0.893	0.929	0.927	0.855
	PSSM-composition	0.947	**0.947**	**0.947**	**0.947**	**0.893**
	Pse-PSSM	**0.96**	0.907	0.935	0.933	0.868
LightGBM	AAC	0.933	0.88	0.909	0.907	0.814
	DPC	0.92	0.907	0.914	0.913	0.827
	PDT	0.92	0.853	0.89	0.887	0.775
	QSOrder	0.893	**0.933**	0.912	0.913	0.827
	CTDC	0.88	**0.933**	0.904	0.907	0.814
	AB-PSSM	0.947	0.893	0.922	0.92	0.841
	EEDP	0.92	0.88	0.902	0.9	0.801
	DP-PSSM	0.96	**0.933**	**0.947**	**0.947**	**0.894**
	PSSM-composition	0.947	0.893	0.922	0.92	0.841
	Pse-PSSM	**0.96**	0.893	0.929	0.927	0.855

NB	AAC	0.947	0.853	0.904	0.9	0.804
	DPC	0.853	0.88	0.865	0.867	0.734
	PDT	0.853	0.76	0.815	0.807	0.616
	QSOrder	0.933	**0.893**	**0.915**	**0.913**	**0.827**
	CTDC	0.893	0.8	0.854	0.847	0.696
	AB-PSSM	0.867	0.813	0.844	0.84	0.681
	EEDP	0.88	0.653	0.79	0.767	0.548
	DP-PSSM	0.893	0.8	0.854	0.847	0.696
	PSSM-composition	**0.96**	0.8	0.889	0.88	0.77
	Pse-PSSM	0.947	0.76	0.866	0.853	0.719
SVM	AAC	0.96	0.947	0.954	0.953	0.907
	DPC	0.867	0.907	0.884	0.887	0.774
	PDT	0.92	0.893	0.908	0.907	0.814
	QSOrder	0.893	0.947	0.918	0.92	0.841
	CTDC	0.96	**0.96**	**0.96**	**0.96**	**0.92**
	AB-PSSM	0.947	0.893	0.922	0.92	0.841
	EEDP	0.947	0.92	0.934	0.933	0.867
	DP-PSSM	**0.987**	0.893	0.943	0.94	0.884
	PSSM-composition	0.987	0.893	0.943	0.94	0.884
	Pse-PSSM	0.947	0.907	0.928	0.927	0.854
XGBoost	AAC	0.933	0.867	0.903	0.9	0.802
	DPC	0.933	0.92	0.927	0.927	0.853
	PDT	0.92	0.84	0.885	0.88	0.762
	QSOrder	0.893	**0.947**	0.918	0.92	0.841
	CTDC	0.907	0.933	0.919	0.92	0.84
	AB-PSSM	0.947	0.907	0.928	0.927	0.854
	EEDP	0.92	0.88	0.902	0.9	0.801
	DP-PSSM	**0.973**	0.92	**0.948**	**0.947**	**0.895**
	PSSM-composition	**0.973**	0.88	0.93	0.927	0.857
	Pse-PSSM	0.96	0.893	0.929	0.927	0.855

Note: The best performance value for each metric across different encoding methods within the same machine learning algorithm is highlighted in bold.

Table S12. Performance comparison of single-method-based models for predicting type VI secreted substrates based on the 5-fold cross-validation test.

Model	Encoding	SN	SP	ACC	F-value	MCC
KNN	AAC	0.903±0.036	0.699±0.073	0.800±0.035	0.818±0.028	0.615±0.065
	DPC	0.852±0.086	0.527±0.108	0.687±0.030	0.728±0.028	0.410±0.058
	DP	0.899±0.056	0.633±0.075	0.764±0.026	0.789±0.022	0.558±0.049
	QSOrder	0.895±0.031	0.693±0.045	0.793±0.029	0.811±0.025	0.600±0.055
	CTDC	0.849±0.039	0.762±0.034	0.805±0.029	0.811±0.027	0.614±0.056
	EEDP	**0.934±0.015**	0.794±0.044	0.863±0.022	0.871±0.018	0.735±0.040
	AAC-PSSM	0.913±0.019	**0.863±0.034**	**0.888±0.012**	**0.890±0.012**	**0.778±0.025**
	AB-PSSM	0.902±0.074	0.788±0.065	0.843±0.025	0.849±0.030	0.699±0.046
	DP-PSSM	0.892±0.029	0.777±0.054	0.832±0.020	0.841±0.015	0.674±0.034
	Pse-PSSM	0.892±0.014	0.810±0.034	0.849±0.015	0.855±0.014	0.704±0.028
RF	AAC	0.826±0.023	0.818±0.037	0.819±0.020	0.819±0.018	0.643±0.041
	DPC	0.834±0.016	0.850±0.024	0.836±0.017	0.834±0.015	0.683±0.033
	DP	0.858±0.017	0.849±0.031	0.851±0.021	0.851±0.018	0.706±0.040
	QSOrder	0.798±0.014	0.823±0.034	0.808±0.021	0.806±0.018	0.621±0.041
	CTDC	0.850±0.029	0.798±0.033	0.824±0.023	0.828±0.023	0.649±0.046
	EEDP	**0.902±0.013**	0.828±0.023	0.864±0.015	0.868±0.014	0.730±0.029
	AAC-PSSM	0.890±0.018	0.884±0.021	0.886±0.016	0.885±0.016	0.773±0.031
	AB-PSSM	0.893±0.018	0.855±0.024	0.872±0.017	0.873±0.017	0.747±0.032
	DP-PSSM	0.871±0.018	**0.904±0.019**	0.885±0.012	0.882±0.013	0.772±0.025
	Pse-PSSM	0.889±0.016	**0.904±0.018**	**0.895±0.016**	**0.893±0.016**	**0.790±0.033**
LightGBM	AAC	0.809±0.026	0.796±0.048	0.801±0.028	0.802±0.025	0.605±0.056
	DPC	0.803±0.025	0.753±0.022	0.773±0.021	0.777±0.021	0.558±0.042
	DP	0.817±0.025	0.766±0.034	0.789±0.022	0.793±0.020	0.582±0.046
	QSOrder	0.818±0.032	0.765±0.051	0.790±0.034	0.795±0.032	0.584±0.071
	CTDC	0.832±0.022	0.798±0.029	0.815±0.019	0.817±0.019	0.631±0.040
	EEDP	0.893±0.024	0.856±0.020	0.873±0.020	0.874±0.021	0.749±0.037
	AAC-PSSM	**0.907±0.015**	0.877±0.024	**0.891±0.017**	**0.892±0.017**	**0.783±0.033**
	AB-PSSM	0.898±0.022	0.859±0.013	0.877±0.016	0.878±0.018	0.757±0.030
	DP-PSSM	0.893±0.015	0.875±0.023	0.883±0.013	0.884±0.013	0.767±0.026
	Pse-PSSM	0.895±0.020	**0.878±0.020**	0.884±0.016	0.885±0.015	0.771±0.033

NB	AAC	0.861±0.022	0.741±0.041	0.800±0.019	0.811±0.017	0.605±0.038
	DPC	0.691±0.081	0.737±0.049	0.713±0.038	0.699±0.053	0.431±0.077
	DP	0.783±0.038	0.773±0.027	0.778±0.013	0.777±0.020	0.558±0.028
	QSOrder	0.842±0.021	0.685±0.050	0.762±0.021	0.779±0.015	0.534±0.042
	CTDC	0.898±0.019	0.666±0.050	0.782±0.025	0.803±0.019	0.578±0.047
	EEDP	0.897±0.026	0.442±0.046	0.671±0.034	0.729±0.024	0.380±0.075
	AAC-PSSM	**0.909±0.013**	0.556±0.045	0.734±0.023	0.771±0.017	0.499±0.044
	AB-PSSM	0.851±0.015	0.644±0.033	0.746±0.023	0.768±0.019	0.504±0.042
	DP-PSSM	0.866±0.034	**0.853±0.023**	**0.857±0.017**	**0.857±0.021**	**0.717±0.035**
	Pse-PSSM	0.876±0.013	0.829±0.021	0.850±0.013	0.852±0.011	0.704±0.025
SVM	AAC	0.861±0.019	0.836±0.034	0.847±0.022	0.849±0.020	0.697±0.043
	DPC	0.820±0.029	0.833±0.031	0.824±0.023	0.824±0.022	0.652±0.047
	DP	0.833±0.021	0.839±0.030	0.835±0.016	0.834±0.016	0.673±0.040
	QSOrder	0.823±0.028	0.824±0.036	0.821±0.028	0.821±0.027	0.645±0.057
	CTDC	0.822±0.032	0.816±0.034	0.819±0.021	0.818±0.022	0.640±0.038
	EEDP	**0.899±0.031**	0.899±0.029	0.899±0.015	0.899±0.015	0.798±0.030
	AAC-PSSM	0.893±0.027	0.927±0.023	**0.909±0.008**	**0.907±0.009**	**0.820±0.015**
	AB-PSSM	0.894±0.021	0.888±0.023	0.890±0.013	0.890±0.013	0.781±0.025
	DP-PSSM	0.888±0.035	**0.927±0.021**	0.905±0.014	0.902±0.017	0.814±0.024
	Pse-PSSM	0.884±0.017	0.910±0.018	0.895±0.013	0.893±0.013	0.792±0.024
XGBoost	AAC	0.816±0.021	0.793±0.034	0.804±0.019	0.804±0.017	0.610±0.039
	DPC	0.810±0.025	0.773±0.036	0.786±0.029	0.788±0.027	0.586±0.055
	DP	0.819±0.028	0.787±0.035	0.800±0.027	0.803±0.026	0.606±0.053
	QSOrder	0.813±0.028	0.765±0.053	0.788±0.033	0.793±0.030	0.579±0.070
	CTDC	0.831±0.032	0.784±0.050	0.807±0.034	0.811±0.031	0.616±0.068
	EEDP	0.899±0.026	0.838±0.026	0.868±0.023	0.871±0.025	0.739±0.045
	AAC-PSSM	0.859±0.022	0.844±0.021	0.850±0.019	0.849±0.021	0.702±0.035
	AB-PSSM	**0.904±0.030**	0.837±0.039	**0.869±0.034**	**0.873±0.034**	**0.741±0.066**
	DP-PSSM	0.856±0.035	0.838±0.022	0.844±0.023	0.844±0.025	0.694±0.041
	Pse-PSSM	0.859±0.018	**0.849±0.015**	0.852±0.015	0.853±0.016	0.707±0.032

Note: Values are expressed as mean±standard deviation. The best performance value for each metric across different encoding methods within the same machine learning algorithm is highlighted in bold.

Table S13. Performance comparison of single-method-based models for predicting type VI secreted substrates based on independent test.

Model	Encoding	SN	SP	ACC	F-value	MCC
KNN	AAC	**1**	0.586	0.793	0.829	0.644
	DPC	**1**	0.241	0.621	0.725	0.37
	DP	0.966	0.621	0.793	0.824	0.625
	QSOrder	**1**	0.586	0.793	0.829	0.644
	CTDC	0.966	0.897	0.931	0.933	0.864
	EEDP	**1**	0.828	0.914	0.921	0.84
	AAC-PSSM	**1**	0.897	0.948	0.951	0.901
	AB-PSSM	**1**	0.655	0.828	0.853	0.698
	DP-PSSM	**1**	0.724	0.862	0.879	0.753
	Pse-PSSM	**1**	**0.931**	**0.966**	**0.967**	**0.933**
RF	AAC	**1**	0.862	0.931	0.935	0.87
	DPC	0.897	0.897	0.897	0.897	0.793
	DP	**1**	**1**	**1**	**1**	**1**
	QSOrder	0.931	0.862	0.897	0.9	0.795
	CTDC	0.966	0.931	0.948	0.949	0.897
	EEDP	0.966	0.897	0.931	0.933	0.864
	AAC-PSSM	**1**	0.931	0.966	0.967	0.933
	AB-PSSM	**1**	0.862	0.931	0.935	0.87
	DP-PSSM	**1**	0.966	0.983	0.983	0.966
	Pse-PSSM	**1**	0.931	0.966	0.967	0.933
LightGBM	AAC	**1**	0.793	0.897	0.906	0.811
	DPC	0.897	0.793	0.845	0.852	0.693
	DP	**1**	0.966	0.983	0.983	0.966
	QSOrder	0.966	0.793	0.879	0.889	0.77
	CTDC	**1**	0.931	0.966	0.967	0.933
	EEDP	**1**	0.828	0.914	0.921	0.84
	AAC-PSSM	**1**	0.931	0.966	0.967	0.933
	AB-PSSM	**1**	0.828	0.914	0.921	0.84
	DP-PSSM	**1**	**1**	**1**	**1**	**1**
	Pse-PSSM	**1**	0.931	0.966	0.967	0.933

NB	AAC	0.931	0.724	0.828	0.844	0.67
	DPC	0.828	**0.862**	0.845	0.842	0.69
	DP	0.966	0.828	0.897	0.903	0.801
	QSOrder	0.931	0.69	0.81	0.831	0.64
	CTDC	0.966	0.759	0.862	0.875	0.74
	EEDP	0.862	0.586	0.724	0.758	0.466
	AAC-PSSM	**1**	0.724	0.862	0.879	0.753
	AB-PSSM	0.966	0.621	0.793	0.824	0.625
	DP-PSSM	**1**	0.931	**0.966**	**0.967**	**0.933**
	Pse-PSSM	**1**	**0.862**	0.931	0.935	0.87
SVM	AAC	**1**	0.862	0.931	0.935	0.87
	DPC	0.931	0.897	0.914	0.915	0.828
	DP	**1**	**0.931**	**0.966**	**0.967**	0.933
	QSOrder	0.966	0.862	0.914	0.918	0.832
	CTDC	0.966	0.828	0.897	0.903	0.801
	EEDP	**1**	0.897	0.948	0.951	0.901
	AAC-PSSM	**1**	**0.931**	**0.966**	**0.967**	**0.933**
	AB-PSSM	**1**	0.862	0.931	0.935	0.87
	DP-PSSM	**1**	**0.931**	**0.966**	**0.967**	**0.933**
	Pse-PSSM	**1**	0.897	0.948	0.951	0.901
XGBoost	AAC	**1**	0.828	0.914	0.921	0.84
	DPC	0.931	0.828	0.879	0.885	0.763
	DP	**1**	0.966	**0.983**	0.983	**0.966**
	QSOrder	0.966	0.828	0.897	0.903	0.801
	CTDC	**1**	0.897	0.948	0.951	0.901
	EEDP	0.966	0.897	0.931	0.933	0.864
	AAC-PSSM	**1**	0.931	0.966	**0.967**	0.933
	AB-PSSM	**1**	0.828	0.914	0.921	0.84
	DP-PSSM	**1**	**0.966**	**0.983**	0.983	**0.966**
	Pse-PSSM	**1**	0.931	0.966	**0.967**	0.933

Note: The best performance value for each metric across different encoding methods within the same machine learning algorithm is highlighted in bold.

Table S14. Performance comparison of ensemble models using different strategies for substrate prediction based on the 5-fold cross-validation test.

Substrate type	Ensemble	SN	SP	ACC	F-value	MCC
I	Stacking	**0.950±0.010**	0.978±0.013	**0.964±0.010**	**0.963±0.010**	**0.927±0.019**
	Averaging	0.933±0.008	0.983±0.013	0.957±0.008	0.956±0.008	0.916±0.018
	Majority voting	0.930±0.008	**0.984±0.013**	0.956±0.008	0.955±0.008	0.914±0.018
II	Stacking	**0.893±0.024**	0.917±0.017	**0.904±0.016**	**0.902±0.016**	**0.812±0.030**
	Averaging	0.882±0.023	0.922±0.024	0.901±0.013	0.898±0.014	0.806±0.026
	Majority voting	0.869±0.028	**0.929±0.019**	0.898±0.016	0.894±0.018	0.802±0.029
III	Stacking	**0.915±0.009**	**0.940±0.010**	**0.927±0.008**	**0.926±0.008**	**0.856±0.017**
	Averaging	0.912±0.006	0.928±0.010	0.919±0.006	0.919±0.006	0.840±0.012
	Majority voting	0.903±0.006	0.929±0.009	0.916±0.006	0.914±0.006	0.832±0.012
IV	Stacking	**0.873±0.012**	**0.942±0.010**	**0.907±0.007**	**0.903±0.008**	**0.817±0.015**
	Averaging	0.871±0.007	0.913±0.016	0.892±0.007	0.889±0.007	0.784±0.016
	Majority voting	0.858±0.011	0.906±0.020	0.882±0.010	0.879±0.010	0.766±0.022
VI	Stacking	0.905±0.019	**0.932±0.020**	**0.917±0.018**	**0.915±0.020**	**0.835±0.036**
	Averaging	**0.937±0.013**	0.886±0.019	0.910±0.015	0.912±0.015	0.822±0.029
	Majority voting	0.928±0.015	0.890±0.017	0.907±0.012	0.909±0.012	0.816±0.025

Note: Values are expressed as mean±standard deviation. The best performance value for each metric across different substrate types is highlighted in bold.

Table S15. Performance comparison of ensemble models using different strategies for substrate prediction based on the independent test.

Substrate type	Ensemble	SN	SP	ACC	F-value	MCC
I	Stacking	**0.966**	**1**	**0.983**	**0.982**	**0.966**
	Averaging	**0.966**	**1**	**0.983**	**0.982**	**0.966**
	Majority voting	**0.966**	**1**	**0.983**	**0.982**	**0.966**
II	Stacking	**0.941**	**1**	**0.971**	**0.97**	**0.943**
	Averaging	**0.941**	**1**	**0.971**	**0.97**	**0.943**
	Majority voting	**0.941**	**1**	**0.971**	**0.97**	**0.943**
III	Stacking	**0.947**	**1**	**0.973**	**0.973**	**0.948**
	Averaging	**0.947**	**1**	**0.973**	**0.973**	**0.948**
	Majority voting	0.936	**1**	0.968	0.967	0.938
IV	Stacking	**0.973**	**1**	**0.987**	**0.986**	**0.974**
	Averaging	**0.973**	**1**	**0.987**	**0.986**	**0.974**
	Majority voting	0.96	**1**	0.98	0.98	0.961
VI	Stacking	**1**	**1**	**1**	**1**	**1**
	Averaging	**1**	**1**	**1**	**1**	**1**
	Majority voting	**1**	**1**	**1**	**1**	**1**

Note: The best performance value for each metric across different substrate types is highlighted in bold.

Table S16. Detailed contributions of different single-method-based models to the final ensemble model for predicting type I secreted substrates.

Model[a]	Importance value	Model[a]	Importance value
DP-PSSM_SVM	6.674	Pse-PSSM_KNN	1.597
DP-PSSM_RF	6.007	CTDC_XGBoost	1.548
RPM-PSSM_XGBoost	5.811	RPSSM_LightGBM	1.460
Pse-PSSM_RF	5.038	AAC_RF	1.421
Pse-PSSM_SVM	4.950	RPSSM_KNN	1.314
QSOrder_SVM	4.917	RPSSM_XGBoost	1.234
RPM-PSSM_RF	4.560	QSOrder_LightGBM	1.177
DP-PSSM_LightGBM	4.308	AAC_KNN	1.066
DPC_RF	4.263	DP-PSSM_NB	1.024
RPM-PSSM_SVM	4.191	DPC_LightGBM	0.932
RPM-PSSM_LightGBM	4.009	QSOrder_NB	0.908
TPC-PSSM_RF	3.604	QSOrder_KNN	0.882
RPSSM_SVM	3.399	AAC_NB	0.821
RPSSM_RF	3.245	TPC-PSSM_NB	0.769
TPC-PSSM_XGBoost	3.063	DPC_XGBoost	0.755
AAC_SVM	2.844	DDE_SVM	0.505
TPC-PSSM_LightGBM	2.814	AAC_XGBoost	0.474
DP-PSSM_KNN	2.770	CTDC_KNN	0.472
DDE_LightGBM	2.590	QSOrder_XGBoost	0.420
QSOrder_RF	2.566	DDE_KNN	0.289
DP-PSSM_XGBoost	2.524	AAC_LightGBM	0.288
Pse-PSSM_LightGBM	2.492	DPC_KNN	0.237
RPM-PSSM_KNN	2.250	TPC-PSSM_SVM	0.226
CTDC_SVM	2.213	CTDC_NB	0.214
DDE_RF	2.177	RPM-PSSM_NB	0.170
Pse-PSSM_NB	2.047	TPC-PSSM_KNN	0.152
DDE_XGBoost	1.995	DPC_SVM	0.141
Pse-PSSM_XGBoost	1.850	RPSSM_NB	0.114
CTDC_RF	1.820	DDE_NB	0.107
CTDC_LightGBM	1.678	DPC_NB	0.098

Note: [a]Models are denoted as (feature encoding method)_(machine learning algorithm). For example, DP-PSSM_SVM denotes that the model is trained with the DP-PSSM feature based on the SVM algorithm.

Table S17. Detailed contributions of different single-method-based models to the final ensemble model for predicting type II secreted substrates.

Model[a]	Importance value	Model[a]	Importance value
AAC-PSSM_SVM	4.538	DP-PSSM_KNN	0.787
Pse-PSSM_SVM	3.659	Pse-PSSM_XGBoost	0.695
Pse-PSSM_NB	2.796	TPC-PSSM_LightGBM	0.692
DP-PSSM_SVM	2.656	TPC-PSSM_KNN	0.678
RPM-PSSM_SVM	2.020	DP_RF	0.628
Pse-PSSM_RF	1.939	QSOrder_LightGBM	0.613
AAC-PSSM_RF	1.681	DP-PSSM_XGBoost	0.612
Pse-PSSM_KNN	1.671	DDE_NB	0.581
RPM-PSSM_KNN	1.608	DPC_RF	0.547
AAC-PSSM_KNN	1.605	DPC_SVM	0.545
AAC-PSSM_LightGBM	1.600	RPM-PSSM_LightGBM	0.452
DP-PSSM_LightGBM	1.478	PDT_KNN	0.435
PDT_SVM	1.442	QSOrder_NB	0.429
TPC-PSSM_RF	1.340	PDT_XGBoost	0.427
Pse-PSSM_LightGBM	1.325	DP_NB	0.426
DP-PSSM_RF	1.319	DPC_LightGBM	0.419
RPM-PSSM_RF	1.281	DPC_KNN	0.415
RPM-PSSM_NB	1.164	DDE_LightGBM	0.407
TPC-PSSM_SVM	1.113	TPC-PSSM_XGBoost	0.382
DDE_SVM	1.069	DP_LightGBM	0.380
QSOrder_KNN	1.026	DDE_RF	0.340
PDT_LightGBM	1.020	RPM-PSSM_XGBoost	0.338
DP_SVM	0.954	DP_XGBoost	0.337
DDE_KNN	0.927	QSOrder_XGBoost	0.313
DP-PSSM_NB	0.919	AAC-PSSM_NB	0.310
PDT_RF	0.904	TPC-PSSM_NB	0.258
QSOrder_SVM	0.875	DDE_XGBoost	0.256
AAC-PSSM_XGBoost	0.848	DPC_XGBoost	0.254
QSOrder_RF	0.847	PDT_NB	0.254
DP_KNN	0.817	DPC_NB	0.254

Note: [a]Models are denoted as (feature encoding method)_(machine learning algorithm). For example, DP-PSSM_SVM denotes that the model is trained with the DP-PSSM feature based on the SVM algorithm.

Table S18. Detailed contributions of different single-method-based models to the final ensemble model for predicting type III secreted substrates.

Model[a]	Importance value	Model[a]	Importance value
DP-PSSM_SVM	34.124	AAC_KNN	2.083
Pse-PSSM_SVM	30.616	DP_SVM	2.043
AB-PSSM_SVM	26.960	TPC-PSSM_KNN	2.000
AAC-PSSM_SVM	20.379	QSOrder_XGBoost	1.894
DP-PSSM_LightGBM	19.249	QSOrder_KNN	1.885
DP-PSSM_XGBoost	18.648	QSOrder_LightGBM	1.879
Pse-PSSM_LightGBM	16.915	AAC_RF	1.862
AB-PSSM_XGBoost	16.443	DP_KNN	1.770
AB-PSSM_RF	15.676	AAC_LightGBM	1.763
AB-PSSM_LightGBM	14.790	QSOrder_RF	1.735
DP-PSSM_RF	14.503	AB-PSSM_NB	1.672
Pse-PSSM_KNN	12.622	DP_XGBoost	1.651
Pse-PSSM_XGBoost	12.400	AAC_XGBoost	1.624
Pse-PSSM_RF	11.367	CTDC_RF	1.543
AAC-PSSM_LightGBM	10.168	DPC_LightGBM	1.531
AAC-PSSM_RF	8.937	DPC_KNN	1.516
AAC-PSSM_XGBoost	8.649	DPC_XGBoost	1.489
DP-PSSM_KNN	7.799	DP_LightGBM	1.483
TPC-PSSM_LightGBM	7.155	CTDC_XGBoost	1.460
AAC-PSSM_KNN	6.547	DP_RF	1.407
AAC_SVM	4.733	CTDC_LightGBM	1.391
TPC-PSSM_XGBoost	4.481	AAC-PSSM_NB	1.376
TPC-PSSM_RF	3.962	CTDC_KNN	1.302
TPC-PSSM_SVM	3.337	DPC_NB	1.278
CTDC_SVM	2.831	CTDC_NB	1.258
QSOrder_SVM	2.684	DP_NB	1.161
Pse-PSSM_NB	2.648	QSOrder_NB	1.133
DP-PSSM_NB	2.528	DPC_RF	1.115
AB-PSSM_KNN	2.165	AAC_NB	1.099
DPC_SVM	2.143	TPC-PSSM_NB	1.075

Note: [a]Models are denoted as (feature encoding method)_(machine learning algorithm). For example, DP-PSSM_SVM denotes that the model is trained with the DP-PSSM feature based on the SVM algorithm.

Table S19. Detailed contributions of different single-method-based models to the final ensemble model for predicting type IV secreted substrates.

Model[a]	Importance value	Model[a]	Importance value
DP-PSSM_SVM	24.567	DPC_SVM	1.864
Pse-PSSM_SVM	17.322	EEDP_NB	1.799
PSSM-composition_SVM	15.274	PDT_KNN	1.700
EEDP_SVM	14.402	DPC_XGBoost	1.699
DP-PSSM_LightGBM	13.017	AB-PSSM_KNN	1.660
Pse-PSSM_RF	12.653	CTDC_NB	1.651
AB-PSSM_SVM	11.132	CTDC_RF	1.636
Pse-PSSM_LightGBM	10.837	DPC_KNN	1.600
PSSM-composition_XGBoost	10.693	CTDC_LightGBM	1.585
PSSM-composition_LightGBM	10.551	PSSM-composition_NB	1.581
AB-PSSM_RF	10.013	QSOrder_LightGBM	1.565
EEDP_RF	9.940	DPC_LightGBM	1.561
Pse-PSSM_XGBoost	9.517	QSOrder_KNN	1.531
DP-PSSM_RF	9.148	CTDC_SVM	1.521
PSSM-composition_RF	8.630	AAC_LightGBM	1.503
EEDP_LightGBM	8.168	CTDC_XGBoost	1.496
AB-PSSM_LightGBM	7.640	PDT_XGBoost	1.477
EEDP_KNN	7.516	DPC_NB	1.467
DP-PSSM_XGBoost	7.085	CTDC_KNN	1.439
Pse-PSSM_KNN	6.561	AAC_KNN	1.413
AB-PSSM_XGBoost	6.545	AB-PSSM_NB	1.409
DP-PSSM_KNN	6.385	PDT_LightGBM	1.380
EEDP_XGBoost	6.032	DPC_RF	1.366
PSSM-composition_KNN	4.256	QSOrder_XGBoost	1.301
DP-PSSM_NB	2.997	QSOrder_RF	1.259
AAC_SVM	2.532	PDT_NB	1.245
Pse-PSSM_NB	2.291	AAC_NB	1.230
PDT_SVM	2.174	QSOrder_NB	1.226
QSOrder_SVM	2.067	AAC_RF	1.162
PDT_RF	1.940	AAC_XGBoost	1.087

Note: [a]Models are denoted as (feature encoding method)_(machine learning algorithm). For example, DP-PSSM_SVM denotes that the model is trained with the DP-PSSM feature based on the SVM algorithm.

Table S20. Detailed contributions of different single-method-based models to the final ensemble model for predicting type VI secreted substrates.

Model[a]	Importance value	Model[a]	Importance value
AAC-PSSM_SVM	8.015	CTDC_RF	0.975
DP-PSSM_SVM	6.566	AAC_KNN	0.883
EEDP_SVM	5.667	DPC_SVM	0.872
Pse-PSSM_SVM	5.509	CTDC_SVM	0.858
Pse-PSSM_RF	5.423	DP_SVM	0.847
AB-PSSM_SVM	4.622	QSOrder_KNN	0.780
AAC-PSSM_KNN	4.568	QSOrder_SVM	0.743
AAC-PSSM_LightGBM	4.313	CTDC_LightGBM	0.743
AAC-PSSM_RF	4.089	CTDC_KNN	0.707
Pse-PSSM_LightGBM	3.947	CTDC_XGBoost	0.704
DP-PSSM_RF	3.709	AAC_RF	0.627
DP-PSSM_LightGBM	3.606	DPC_XGBoost	0.608
EEDP_KNN	3.362	DP_KNN	0.582
AB-PSSM_XGBoost	3.198	DPC_LightGBM	0.574
EEDP_LightGBM	3.053	AAC_LightGBM	0.567
AB-PSSM_LightGBM	3.000	CTDC_NB	0.537
AB-PSSM_RF	2.717	QSOrder_RF	0.521
EEDP_XGBoost	2.352	DP_XGBoost	0.500
EEDP_RF	2.291	QSOrder_LightGBM	0.471
AB-PSSM_KNN	2.002	AAC_XGBoost	0.467
Pse-PSSM_KNN	1.952	AAC_NB	0.457
DP-PSSM_NB	1.918	DP_LightGBM	0.453
DP_RF	1.655	QSOrder_XGBoost	0.441
DP-PSSM_KNN	1.561	DP_NB	0.440
AAC_SVM	1.557	DPC_NB	0.440
Pse-PSSM_NB	1.497	DPC_KNN	0.437
Pse-PSSM_XGBoost	1.408	QSOrder_NB	0.397
AAC-PSSM_XGBoost	1.392	AAC-PSSM_NB	0.391
DP-PSSM_XGBoost	1.312	AB-PSSM_NB	0.362
DPC_RF	1.007	EEDP_NB	0.355

Note: [a]Models are denoted as (feature encoding method)_(machine learning algorithm). For example, DP-PSSM_SVM denotes that the model is trained with the DP-PSSM feature based on the SVM algorithm.

Table S21. Performance comparison between BastionX and existing state-of-the-art toolkits for predicting singe types of secreted substrates based on the independent test.

Substrate type	Toolkit	SN	SP	F-value	ACC	MCC
III	BastionX	0.947	**1**	**0.973**	**0.973**	**0.948**
	Bastion3	**0.978**	0.968	**0.973**	**0.973**	0.947
	DeepT3	0.734	1	0.867	0.847	0.761
IV	BastionX	0.973	**1**	**0.986**	0.987	0.974
	Bastion4	0.973	0.973	0.973	0.973	0.947
	PredT4SE-Stack	**1**	0.98	0.96	0.98	0.961
	CNN-T4SE	**1**	0.993	**0.986**	**0.993**	**0.987**
VI	BastionX	**1**	**1**	**1**	**1**	**1**
	Bastion6	**1**	0.931	0.966	0.967	0.933

Note: The best performance value for each metric across different substrate type is highlighted in bold.

Table S22. Detailed information of five secreted substrate proteins used in the case study.

Substrate ID	Substrate type	Substrate/Gene name	Species	Reference
1	I	AprA	*Pseudomonas brassicacearum*	(Chabeaud, et al., 2001)
2	II	*lpp0489*	*Legionella pneumophila*	(Herrmann, et al., 2011)
3	III	*spvC*	*Salmonella typhimurium*	(Yang, et al., 2013)
4	IV	*lpg0160*	*Legionella pneumophila*	(Zou, et al., 2013)
5	VI	Chain A	*Pseudomonas Aeruginousa*	(Dong, et al., 2013)

Table S23. Detailed prediction results of case study in different tools.

Substrate ID	Substrate type	BastionX (type I)	BastionX (type II)	BastionX (type III)	BastionX (type IV)	BastionX (type VI)	BastionX (Final)
1	I	**1**	**1**	0.3202	0.4479	0.5336	I or II
2	II	0	**0.9588**	0.1772	0.6605	0.5641	II
3	III	0	0.4993	**1**	0.9582	0.9881	III
4	IV	0.0288	0.7403	0.9014	**0.9997**	0.5055	IV
5	VI	0.4666	0.851	0.9862	0.7928	**0.9996**	VI

Note: The wrong predict result is highlighted in red. The highest prediction score across different singe type predictor within the BastionX for each case study is highlighted in bold.

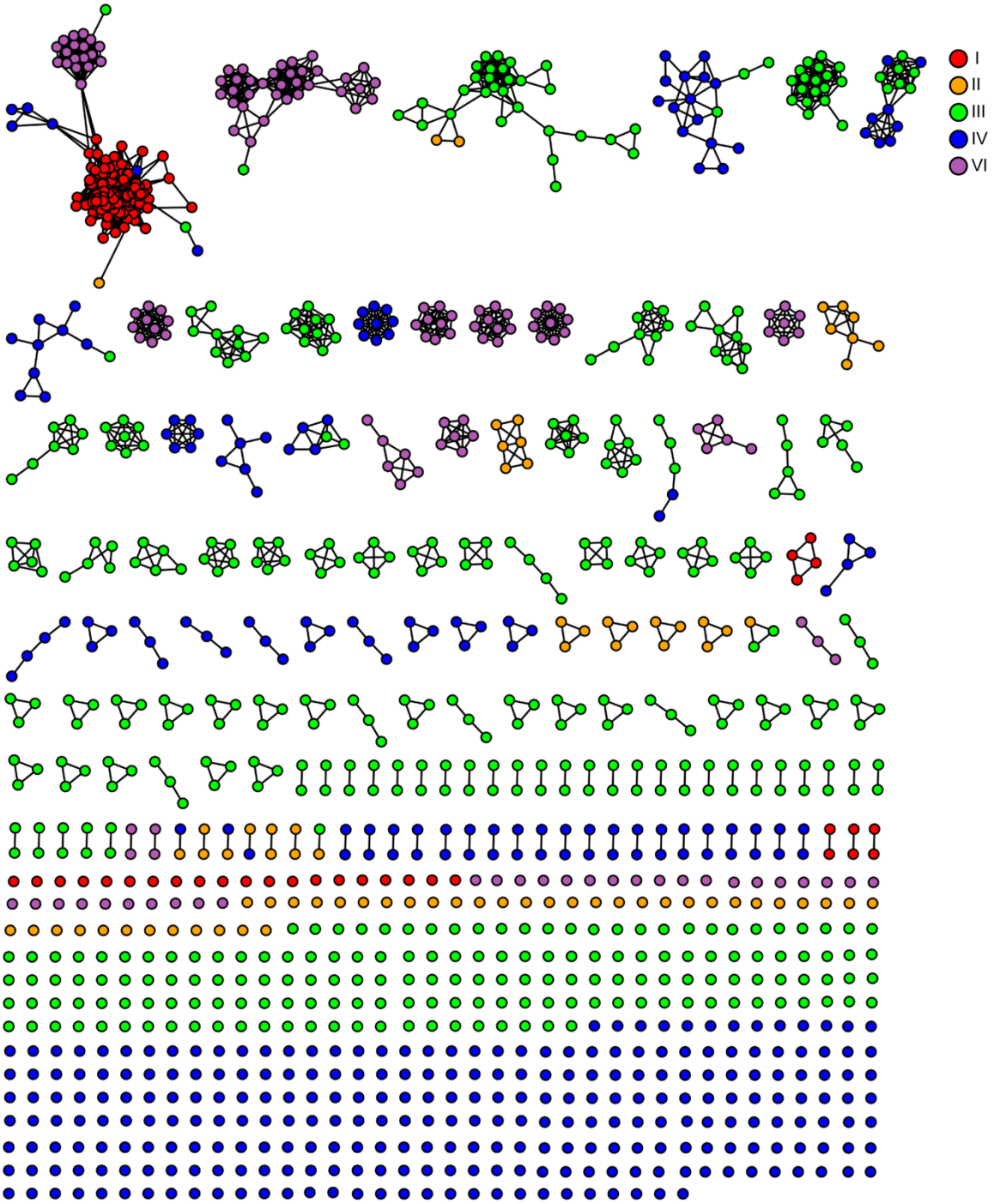


Figure S1. Visualization of sequence similarity network (SSN) across various types of substrates based on the curated datasets after redundancy reduction. An all-by-all BLAST was conducted using the EFI-EST toolkit (Gerlt, et al., 2015) to generate the SSN file, which was further visualized by the Cytoscape software (Shannon, et al., 2003). The dots I, II, III, IV and VI represent type I, II, III, IV and VI substrates, respectively.

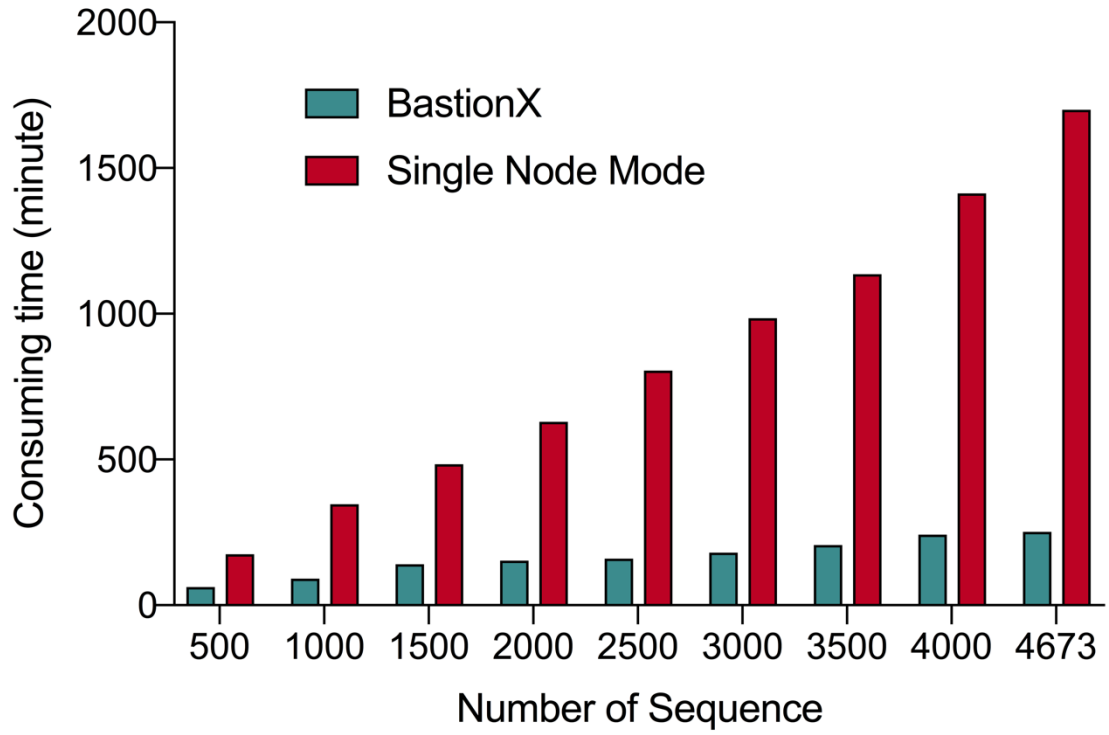


Figure S2. Performance comparison between BastionX (using 10 computing nodes) and its singe node mode in terms of the computing time. Both groups of experiments were conducted based on a genome-scale dataset from *Escherichia coli* IAI39. Sub-nodes in the BastionX predictor and its single node involved in all experiments run 16 threads to concurrently process the sub-tasks.

References

Chabeaud, P.*, et al.* Phase-variable expression of an operon encoding extracellular alkaline protease, a serine protease homolog, and lipase in Pseudomonas brassicacearum. *Journal of bacteriology* 2001;183(6):2117-2120.

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

Chou, K.C. Prediction of protein subcellular locations by incorporating quasi-sequence-order effect. Biochemical and biophysical research communications 2000;278(2):477-483.

Chou, K.C. and Shen, H.B. MemType-2L: a web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. Biochemical and biophysical research communications 2007;360(2):339-345.

Ding, S., et al. A protein structural classes prediction method based on predicted secondary structure and PSI-BLAST profile. Biochimie 2014;97:60-65.

Dong, C., et al. Structural insights into the inhibition of type VI effector Tae3 by its immunity protein Tai3. Biochemical Journal 2013;454(1):59-68.

Gerlt, J.A., et al. Enzyme Function Initiative-Enzyme Similarity Tool (EFI-EST): A web tool for generating protein sequence similarity networks. Biochimica et biophysica acta 2015;1854(8):1019-1037.

Herrmann, V., et al. GamA is a eukaryotic-like glucoamylase responsible for glycogen- and starch-degrading activity of Legionella pneumophila. International journal of medical microbiology : IJMM 2011;301(2):133-139.

Jeong, J.C., Lin, X. and Chen, X.W. On position-specific scoring matrix for protein function prediction. IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM 2011;8(2):308-315.

Juan, E.Y.T., et al. Predicting Protein Subcellular Localizations for Gram-Negative Bacteria using DP-PSSM and Support Vector Machines. Cisis: 2009 International Conference on Complex, Intelligent and Software Intensive Systems, Vols 1 and 2 2009:836-841.

Lin, J.M., et al. Transcription factor binding and modified histones in human bidirectional promoters. Genome research 2007;17(6):818-827.

Liu, B., et al. A discriminative method for protein remote homology detection and fold recognition combining Top-n-grams and latent semantic analysis. BMC bioinformatics 2008;9:510.

Liu, B., et al. iDNA-Prot|dis: identifying DNA-binding proteins by incorporating amino acid distance-pairs and reduced alphabet profile into the general pseudo amino acid composition. PloS one 2014;9(9):e106691.

Liu, T., et al. Accurate prediction of protein structural class using auto covariance transformation of PSI-BLAST profiles. Amino acids 2012;42(6):2243-2249.

Liu, T., Zheng, X. and Wang, J. Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile. Biochimie 2010;92(10):1330-1334.

Shannon, P., et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome research 2003;13(11):2498-2504.

Yang, X., et al. Effective identification of Gram-negative bacterial type III secreted effectors using position-specific residue conservation profiles. PloS one 2013;8(12):e84439.

Zhang, L., Zhao, X. and Kong, L. Predict protein structural class for low-similarity sequences by evolutionary difference information into the general form of Chou's pseudo amino acid composition. Journal of theoretical biology 2014;355:105-110.

Zhang, S., Ye, F. and Yuan, X. Using principal component analysis and support vector machine to predict protein structural class for low-similarity sequences via PSSM. Journal of biomolecular structure & dynamics 2012;29(6):634-642.
Zou, L., Nan, C. and Hu, F. Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles. *Bioinformatics* 2013;29(24):3135-3142.

Appendix 5 - Supplementary information for Chapter 4.1

POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles

Supplementary Material

SUPPLEMENTAL INFORMATION

Table S1 provides a comprehensive list of a wide range of research areas and application topics within the literature for which PSSM profile-based features have proved to be useful.

Table S1. Research topics and areas of PSSM profile-based features in the literature.

Research Area	Feature Descriptors by the Corresponding Research Work	References
Protein structural class prediction	**AAC-PSSM**, **DPC-PSSM**, and **AADP-PSSM**	(Liu, et al., 2010)
	AAC-PSSM, and **PSSM-AC**	(Liu, et al., 2012)
	AAC, and **PSSM**	(Chen, et al., 2008)
	AAC-PSSM, **PSSM-AC**, consensus sequence descriptors, and physicochemical property features	(Dehzangi, et al., 2013)
	RPSSM, and secondary structures	(Ding, et al., 2014)
	tri-gram-PSSM	(Tao, et al., 2015)
	PSSM, physicochemical property features, and GO feature descriptors	(Li, et al., 2014)
	EDP, **EEDP**, and **MEDP**	(Zhang, et al., 2014)
	AAC-PSSM, **TPC**, and **AATP**	(Zhang, et al., 2012)
	PSSM	(Xia, et al., 2012)
Post-translational modification site prediction	**PSSM**, disorder scores, secondary structures, solvent accessibilities, AAIndex, and AAC	(Jiang, et al., 2013)
	AAC, AGG, BLOSUM62, charge-hyd, CKSAAP, binary profiles, disorder scores, KNN, and **PSSM**	(Chen, et al., 2015)

	AAIndex, physicochemical descriptors, **PSSM**, evolutionary conservation scores, CKSAAP; predicted disordered regions, predicted secondary structures, predicted solvent accessibilities; BP, cellular component, molecular function, functional domain from InterPro, pathway information, functional domain from Pfam, protein-protein interaction annotations; functional domain annotations, nucleotide-binding site annotations, disulfide bond annotations, post-translational modified residue annotations, active site annotations, natural variant annotations, metal ion-binding site annotations, and other binding site annotations	(Li, et al., 2015)
	PSSM, AAC, DPC, solvent accessible surface areas, BLOSUM62, PWM, AAIndex	(Bui, et al., 2016)
	binary profiles, AAC, secondary structures, solvent accessible surface areas, and **PSSM**	(Chauhan, et al., 2012)
	PSSM, AAIndex, secondary structures, solvent accessible surface areas, and disorder scores	(Zhang, et al., 2014)
Protein fold recognition	**PSSM**, profile-profile alignments, secondary-structure specific gap-penalties, classic pair and solvation potentials	(Lobley, et al., 2009)
	Sequence and family information; sequence-sequence alignment; sequence-profile alignment; profile-profile alignment (including **PSSM**), and structural information	(Cheng and Baldi, 2006)
	***k*-separated-bigrams-PSSM**	(Sharma, et al., 2013)
	***k*-separated-bigrams-PSSM**	(Saini, et al.)
	PSSM-AC, and **PSSM-CC**	(Dong, et al., 2009)
	tri-gram-PSSM	(Paliwal, et al., 2014)
	PSSM	(Hong, et al., 2011)
Prediction of protein-protein interactions	**D-FPSSM**, and **S-FPSSM**	(Zahiri, et al., 2013)
	physicochemical descriptors, **PSSM-AC**, and **PSSM-CC**	(Guo, et al., 2008)
	physicochemical descriptors, evolutionary conservation scores, information entropy, **PSSM**, ASA, NC_a, and NC_r	(Deng, et al., 2009)
	PSSM, and predicted solvent accessibility	(Murakami and Mizuguchi, 2010)
	PSSM, and **PSSM-AC**	(Gao, et al., 2016)
	PSSM, and ***k*-separated-bigrams-PSSM**	(An, et al., 2016)
	PSSM, and solvent accessible surface areas	(Melo, et al., 2016)

Membrane protein topology prediction	**Pse-PSSM**	(Chou and Shen, 2007)
	PSSM, and IAMPC (Integrated Approach for Membrane Protein Classification)	(Pu, et al., 2007)
	physicochemical descriptors, and **PSSM**	(Hayat and Khan, 2012)
	PSSM, and secondary structures	(Yan, et al., 2015)
	PSSM, AAC, DPC, physicochemical descriptors, and biochemical feature descriptors	(Mishra, et al., 2014)
	PSSM, and biochemical feature descriptors	(Chen, et al., 2011)
Prediction of protein subcellular localization	**PSSM**	(Xie, et al., 2005)
	DP-PSSM	(Juan, et al., 2009)
	Pse-PSSM	(Juan, et al., 2008)
	PSSM, and PSFM	(Guo, et al., 2006)
	PseAAC, and **PSSM-AC**	(Wang and Li, 2013)
Bacterial protein prediction	AAC, secondary structures, solvent accessibilities, physicochemical descriptors, and **PSSM**	(Yang, et al., 2013)
	AAC, DPC, **PSSM-composition**, and **PSSM-AC**	(Zou, et al., 2013)
	AAC, DPC, and **PSSM**	(Garg and Gupta, 2008)
	AAC, DPC, MM, and **PSSM**	(Selvaraj, et al., 2016)
	AAC, DPC, physicochemical property features, and **PSSM**	(Restrepo-Montoya, et al., 2011)
HIV 1 protease cleavage prediction	**PSSM**	(Jensen, et al., 2003)
	PSSM	(Jensen, et al., 2006)
	geno2pheno, and **PSSM**	(Seclen, et al., 2011)
	geno2pheno, and **PSSM**	(Bunnik, et al., 2011)
Protein disorder prediction	**PSSM**, and BLOSUM62	(Jones and Cozzetto, 2015)
	PSSM	(Jones and Ward, 2003)

	PSSM, and physicochemical property features	(Shimizu, et al., 2007)
	PSSM, secondary structures, and solvent accessibilities	(Becker, et al., 2013)
	PSSM, and physicochemical descriptors	(Su, et al., 2006)
Protein secondary structure prediction	**PSSM**	(Bouziane, et al., 2011)
	PSSM, and SPSSM	(Li, et al., 2012)
	PSSM	(Tang, et al., 2011)
	conformation parameters, **PSSM**, net charges, hydrophobic and side chain mass	(Huang and Chen, 2013)
Prediction of DNA-binding sites	**PSSM**	(Ahmad and Sarai, 2005)
	biochemical descriptors and **PSSM**	(Wang, et al., 2010)
	AAC, DPC and **PSSM**	(Kumar, et al., 2007)
	physicochemical descriptors, biochemical descriptors and **PSSM**	(Huang, et al., 2011)
	binary profile, BLOSUM62 and **PSSM**	(Hwang, et al., 2007)
Prediction of RNA-binding sites	**PSSM**, **smoothed-PSSM**	(Cheng, et al., 2008)
	physicochemical descriptors, hydrophobicity, relative accessible surface areas, secondary structures, **PSSM**, and side-chain environment	(Liu, et al., 2010)
	PSSM	(Kumar, et al., 2008)
	PSSM, residue interface propensity, predicted residue accessibility values, and residue hydrophobicity scores	(Murakami, et al., 2010)
	biochemical property features, and **PSSM**	(Wang, et al., 2010)
	PSSM, **smoothed-PSSM**, and sequence-derived descriptors	(Walia, et al., 2012)
Protein function prediction	**AB-PSSM**, **RPM-PSSM**, and physicochemical property features	(Jeong, et al., 2011)
	PSSM, UniProtKB/Swiss-Prot text mining, amino acid trigram mining, FFPRED, orthologous groups, profile-profile comparison, and functional space	(Cozzetto, et al., 2013)
	GO annotations, and **PSSM**	(Wass and Sternberg, 2008)

*PSSM denotes that the original PSSM profile was directly used in the corresponding paper.

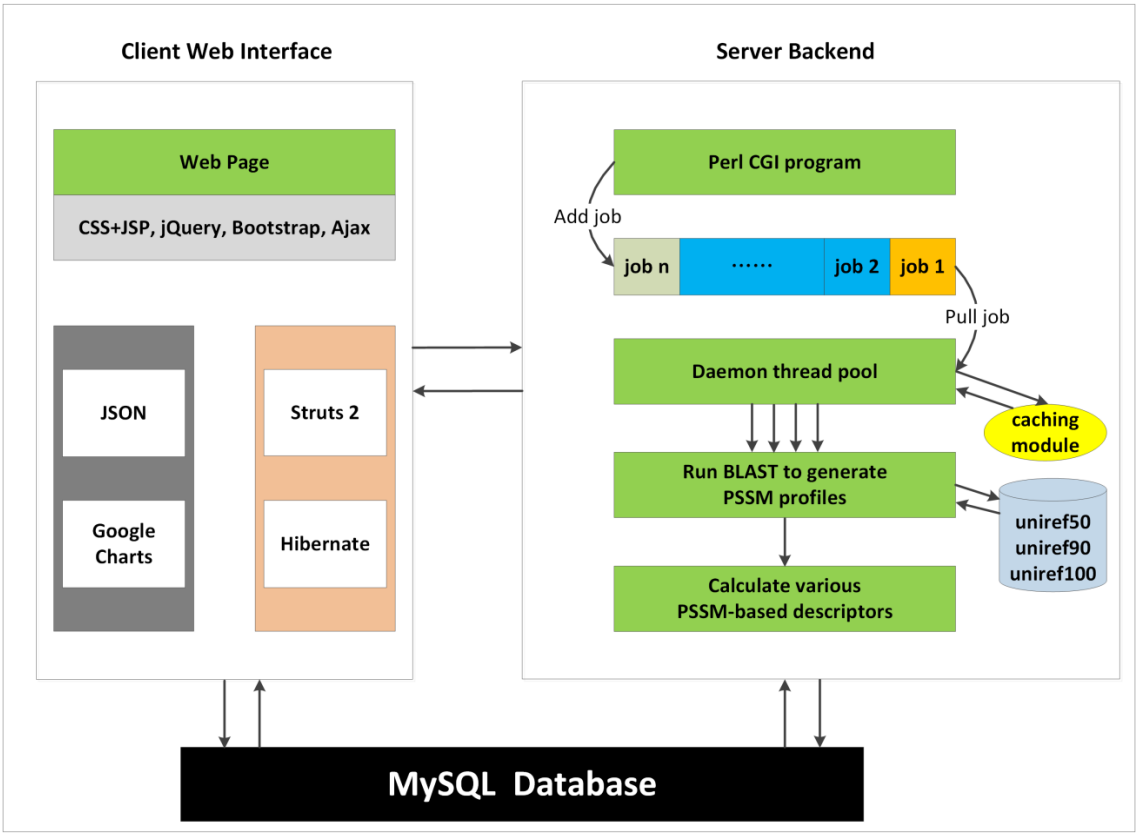


Fig. S1. The architecture of the POSSUM web server.

The architecture of the POSSUM server is illustrated in Fig. S1. There are two main components to this architecture: Client Web Interface and Server Backend. These two components can interactively exchange the data of submitted jobs, and inform each other. Please refer to the main text of the manuscript for a detailed description and discussion.

The POSSUM server is currently configured and hosted on an extensible cloud computing facility provided by the e-Research Centre at Monash University, equipped with 4 cores, 16GB memory and a 1TB hard disk. Importantly, this configuration can be readily expanded and upgraded in accordance with the increasing user demand of the webserver.

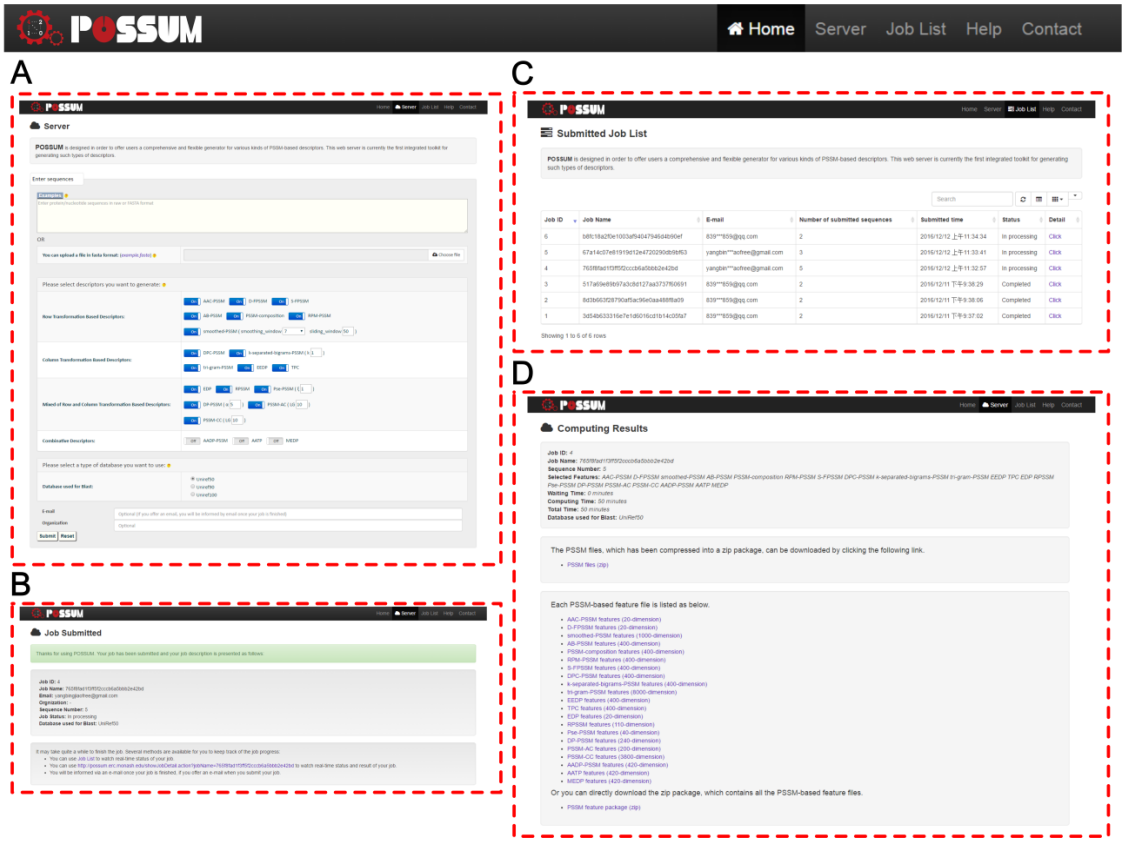


Fig. S2. An example of the user interface of the POSSUM server: (A) Webpage displaying users' submission options; (B) Webpage summarizing the submitted information; (C) Webpage listing status of all submitted jobs, and (D) The result page containing the original PSSM files and calculated descriptors by POSSUM, as well as the links for downloading the corresponding PSSM-based feature files.

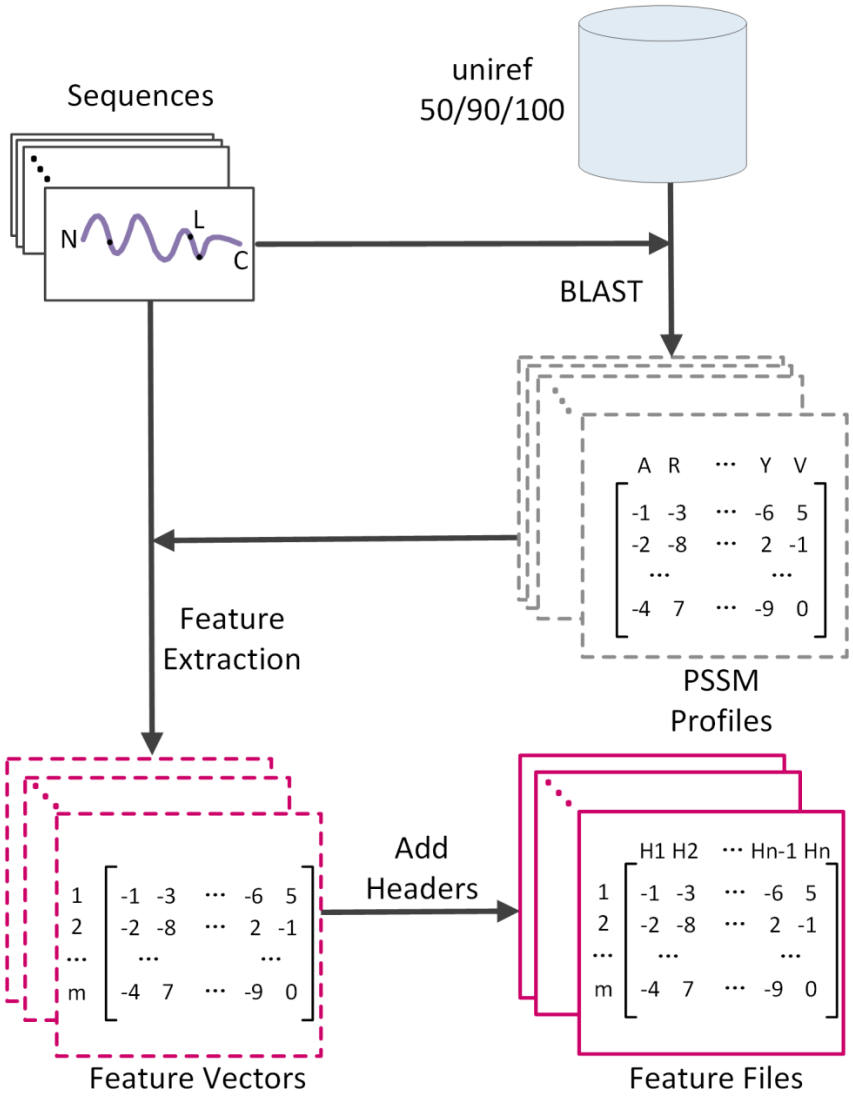


Fig. S3. Workflow of the POSSUM server.

The workflow of the POSSUM server is displayed in Fig. S3.

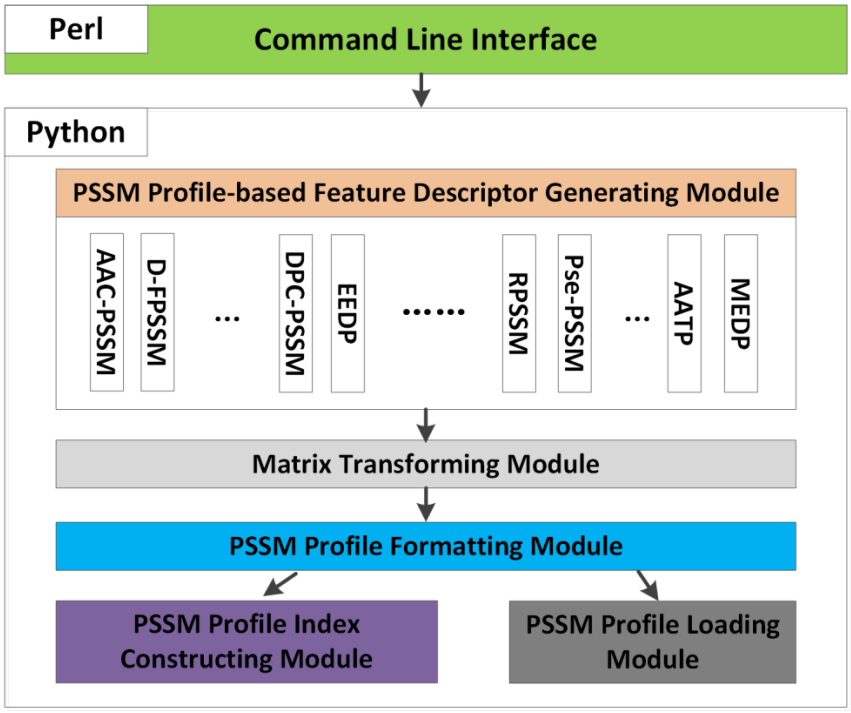


Fig. S4. Architecture of the POSSUM standalone toolkit.

The architecture of the POSSUM standalone toolkit is displayed in Fig. S4. The toolkit was implemented in Python (for core function implementation) and Perl (for universal command line interface). The major components of the toolkit are briefly described as follows:

- Command Line Interface: This module is made available to provide a universal and user-friendly command line interface, via which users can effectively interact with the toolkit. This module allows users to specify and apply different parameters and it invokes the descriptor generating process.

- PSSM Profile-based Feature Descriptor Generating Module: This module can be used to wrap up and output the descriptor files based on the raw descriptor vectors (generated by the Matrix Transforming Module) in accordance with the user-specified parameters.

- Matrix Transforming Module: This module can be used to transform the PSSM matrix (which is abstracted from the original PSSM profile) to generate user-specified raw descriptor vectors. Various applicable matrix transformation functions in groups of row transformations, column transformations, and mixture of row and column transformations are available within this module.

- PSSM Profile Formatting Module: This module can be used to abstract the PSSM matrix from the PSSM profile.
- PSSM Profile Index Constructing Module: This module is a fundamental part of the program that scans the FASTA sequences and the PSSM profile folder to build a hash map for each query sequence and its corresponding PSSM profile.
- PSSM Profile Loading Module: This module looks up the hash table (built by the PSSM Profile Index Constructing Module) to check the availability of the PSSM profile for a sequence and loads the corresponding PSSM profile into the memory.

Comparison of the computational time of PSSM profile-based feature descriptor generation by POSSUM on different uniref databases

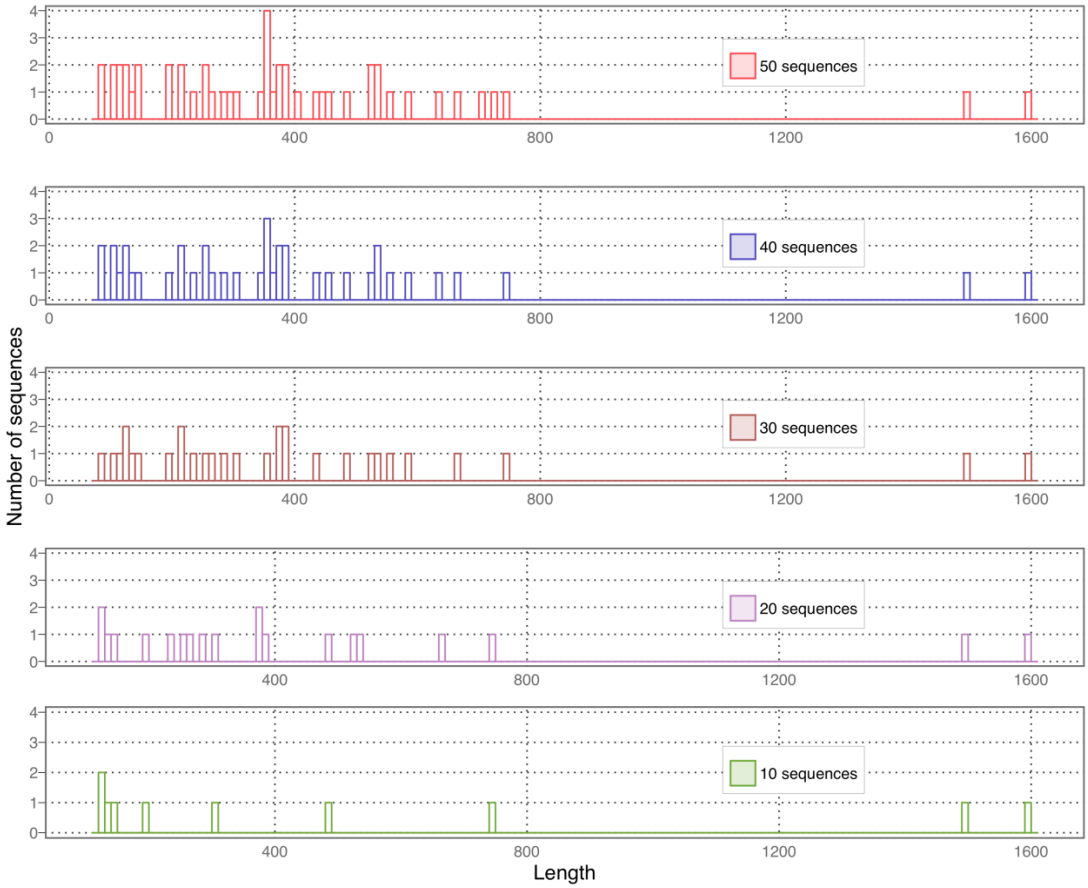


Fig. S5. The distribution of submitted sequence lengths.

Next, in order to illustrate the computational power of POSSUM, we randomly selected 50 sequences from the UniProt database (http://www.uniprot.org/). We subsequently evaluated POSSUM server's CPU computing time for generating PSSM profile-based feature descriptors on the three different uniref databases (i.e. uniref50, uniref90 and uniref100). Specifically, we submitted 10, 20, 30, 40 and 50 sequences to the POSSUM server to generate all 21 types of PSSM profile-based feature descriptors. The distributions of sequence lengths for these tasks, their computational time against different uniref databases, and the distributions of the computational time over a certain task (generating PSSM profile-based feature descriptors for 50 sequences on uniref50) are shown in Fig. S5, Fig. S6 and Fig. S7, respectively.

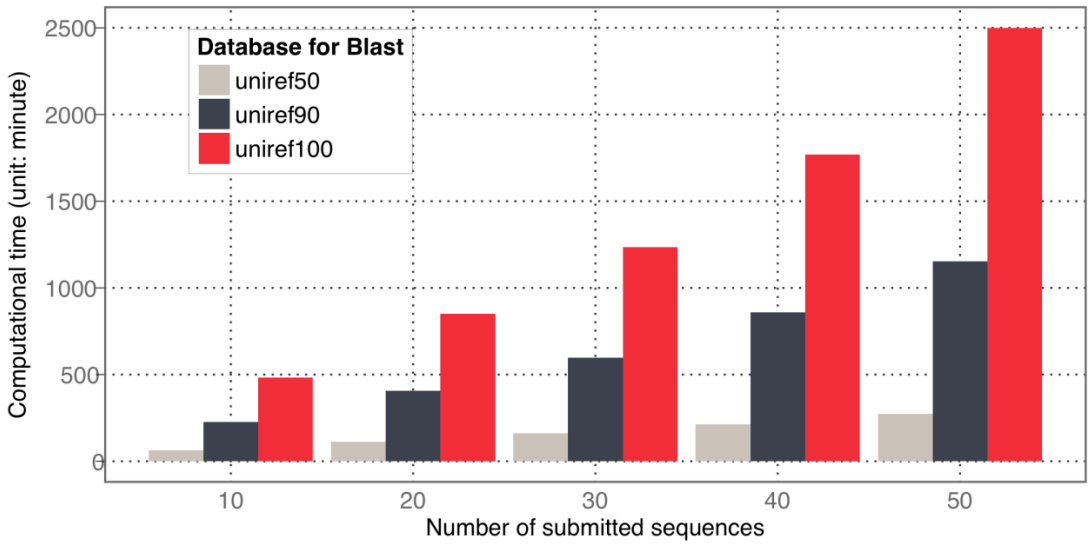


Fig. S6. Comparisons of the computational time for the POSSUM server to process and generate the PSSM profile-based feature descriptors of varying numbers of sequences using three different uniref databases (i.e. uniref50, uniref90 and uniref100). The three databases were generated based on different sequence identity thresholds. The computational time on the y-axis indicates the total computational time for submitted sequences (unit: minute).

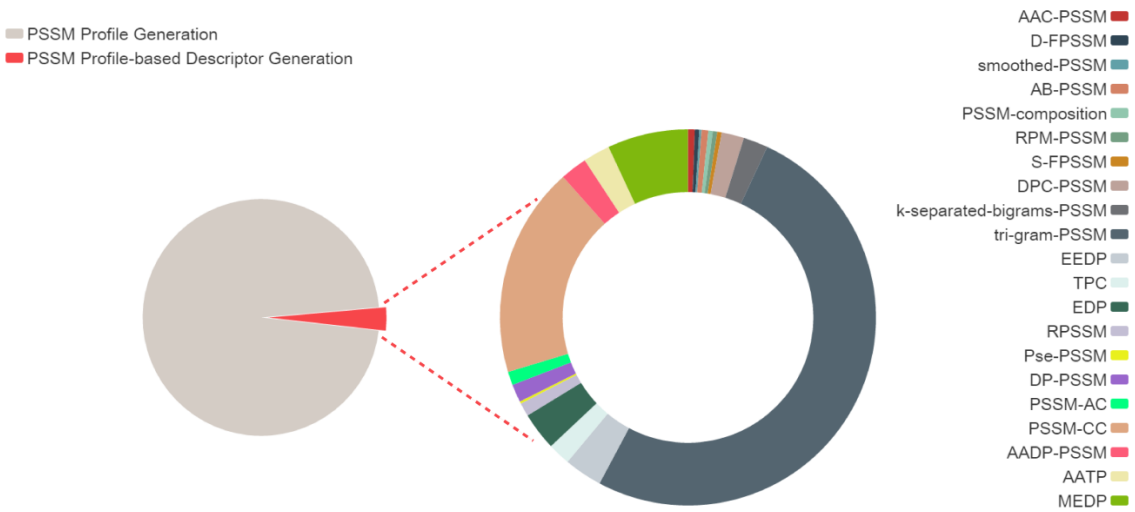


Fig. S7. Distribution of the computational time involved in the task of generating all types of PSSM profile-based feature descriptors as a whole. The results were obtained over the 50 randomly selected sequences based on the uniref50 database.

Fig. S6 suggests a near linear relationship between the CPU computational time and the number of submitted sequences, provided the same uniref database was used. Nevertheless, the computational time considerably varied depending on which uniref database was used for the same task. Users should keep in mind there is a trade-off between the quality of the PSSM profiles generated and computational efficiency, and select which options would best suit their practical needs.

Furthermore, generating a PSSM profile is the most time-consuming step during the entire feature descriptor generation process (Fig. S7, left panel), accounting for 96.8% of the computing time. In this regard, parallelization of the PSSM profile generation is expected to significantly boost the throughput of the POSSUM server. In addition, we also notice that during the calculation of PSSM profile-based feature descriptors (Fig. S7, right panel), the tri-gram-PSSM is the most time-consuming step due to a very large number of features (described as a vector in a 8000-dimensional space) required to be generated.

Application of POSSUM-calculated features to the prediction of type IV secretion effectors and performance evaluation based on the 10 times of 5-fold cross-validation tests

To demonstrate the usefulness of PSSM-based features generated by POSSUM, we further applied POSSUM features to the prediction of type IV secretion effector proteins and examined

the performance of machine learning models trained using these features. We employed the dataset prepared in (Zou, et al., 2013) as the benchmark dataset for the performance comparison, which included 340 type IV effectors and 1132 non-effectors. After removing the sequence redundancy, 338 positive and 338 negative samples were finally selected. Based on this dataset, all 21 types of feature descriptors were generated using POSSUM. In addition, some well-known sequence-based descriptors were used as a reference, such as composition of k-spaced amino acid pairs (CKSAAP) (Chen, et al., 2011), amphiphilic pseudo-amino acid composition (APAAC), pseudo-amino acid composition (PAAC), and quasi-sequence-order (QSO), which are originally proposed in (Chou, 2000; Chou, 2001) and implemented using the protr package (Xiao, et al., 2015).

Table S2. The list of performances of various descriptors.

Descriptors groups	Descriptor	SN	SP	ACC	F-value	MCC
Row transformation	AAC-PSSM	0.883±0.007	0.919±0.009	0.901±0.005	0.899±0.005	0.803±0.011
	D-FPSSM	0.829±0.010	0.895±0.008	0.862±0.007	0.856±0.008	0.725±0.014
	smoothed-PSSM	0.835±0.005	0.919±0.005	0.877±0.003	0.871±0.003	0.757±0.007
	AB-PSSM	0.868±0.004	0.925±0.007	0.896±0.005	0.893±0.004	0.795±0.009
	PSSM-composition	0.879±0.008	0.908±0.003	0.894±0.004	0.891±0.004	0.789±0.007
	RPM-PSSM	0.866±0.007	**0.935±0.008**	0.900±0.003	0.896±0.003	0.803±0.007
	S-FPSSM	0.843±0.008	0.923±0.006	0.883±0.005	0.877±0.005	0.769±0.010
Column transformation	DPC-PSSM	0.873±0.006	0.915±0.006	0.894±0.004	0.891±0.005	0.789±0.009
	k-separated-bigrams-PSSM	0.859±0.007	0.916±0.011	0.888±0.006	0.884±0.006	0.777±0.013
	tri-gram-PSSM	0.869±0.007	0.890±0.009	0.880±0.007	0.878±0.007	0.760±0.014
	EEDP	0.878±0.005	0.931±0.007	**0.904±0.005**	0.901±0.005	**0.810±0.010**
	TPC	0.904±0.005	0.897±0.007	0.901±0.004	0.901±0.004	0.802±0.007
Mixed of row and column transformation	EDP	0.854±0.005	0.915±0.004	0.884±0.003	0.880±0.004	0.771±0.006
	RPSSM	0.871±0.006	0.922±0.004	0.897±0.003	0.893±0.003	0.794±0.006
	Pse-PSSM	0.874±0.007	0.926±0.006	0.900±0.005	0.897±0.006	0.801±0.011
	DP-PSSM	0.873±0.007	0.933±0.005	0.903±0.004	0.900±0.005	0.808±0.007

	PSSM-AC	0.770±0.008	0.914±0.010	0.842±0.006	0.829±0.006	0.691±0.013
	PSSM-CC	0.815±0.007	0.912±0.007	0.863±0.006	0.855±0.005	0.730±0.011
Combination of above descriptors	AADP-PSSM	0.876±0.005	0.912±0.004	0.894±0.004	0.891±0.004	0.789±0.007
	AATP	**0.905±0.007**	0.902±0.005	0.903±0.005	**0.903±0.005**	0.807±0.010
	MEDP	0.875±0.006	0.929±0.002	0.902±0.003	0.899±0.004	0.806±0.005
Sequence-based descriptors	AAC	0.778±0.008	0.826±0.005	0.802±0.006	0.797±0.006	0.605±0.012
	DPC	0.788±0.010	0.824±0.013	0.806±0.009	0.801±0.009	0.613±0.020
	CKSAAP	0.797±0.011	0.830±0.007	0.814±0.007	0.810±0.008	0.629±0.014
	APAAC	0.766±0.011	0.806±0.017	0.786±0.011	0.781±0.010	0.573±0.022
	PAAC	0.769±0.013	0.805±0.015	0.787±0.008	0.782±0.008	0.575±0.017
	QSO	0.762±0.006	0.842±0.009	0.802±0.005	0.794±0.005	0.606±0.010

The rows highlighted by grey are the descriptors achieving MCC values of 0.800 or larger.

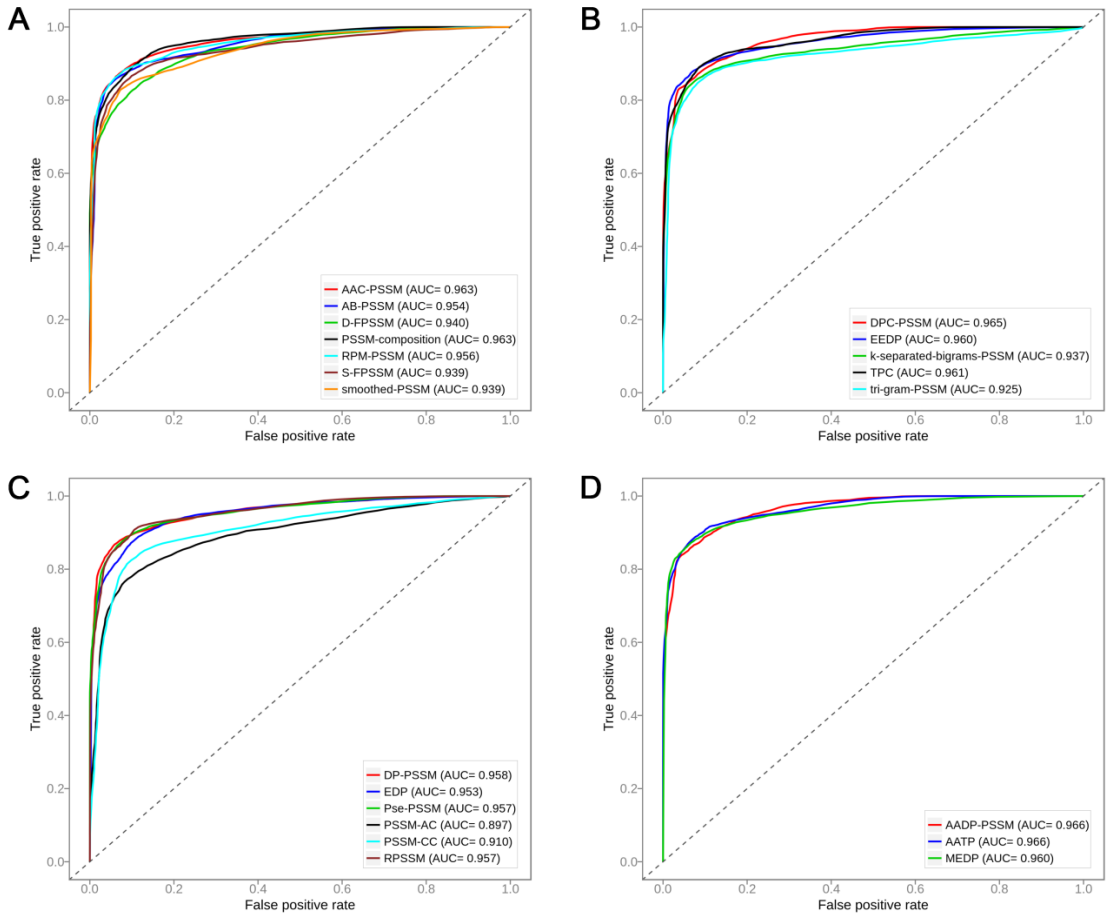


Fig. S8. Prediction performance of type IV secretion effectors using random forest classifiers, trained using multiple different feature descriptors generated by POSSUM as input features. The performance results were evaluated based on the 10 times randomization tests of 5-fold cross-validation. (A) ROC curves of random forest classifiers trained with feature descriptors within the row-transformation group; (B) ROC curves of random forest classifiers trained with feature descriptors within the column-transformation group; (C) ROC curves of random forest classifiers trained with feature descriptors within the mixture of row-transformation and column-transformation group, and (D) ROC curves of random forest classifiers trained with feature descriptors by combinations of rest groups.

For each type of PSSM-based features, the random forest classifier was trained and validated based on the 10-time randomization tests of 5-fold cross-validation. Respective results are shown in Table S2 and Fig. S8.

As can be observed from Table S2, PSSM-based descriptors performed much better when compared with sequence-based descriptors in terms of ACC, F-value and MCC scores. These

results indicate that PSSM descriptors are much more informative, significantly contributing to the model performance. On the other hand, the RF classifiers trained using different types of PSSM-derived features achieved a varying performance, in terms of ACC (ranging from 0.842 to 0.904), F-value (ranging from 0.829 to 0.903) and MCC (ranging from 0.691 to 0.810), depending on the particular PSSM feature type used for training the RF models. The performance discrepancy implies that selection of optimal PSSM features that best suit the specific classification task should be exercised with caution. POSSUM is a tool that offers the opportunity to do the latter, by allowing interested users to address this technically challenging yet important question and meet their specific needs and facilitate their efforts to optimize the model performance within a homogenous framework. Statistically quantifying the contribution of various PSSM-based features to the prediction performance of the machine learning models is a relevant question of interest, as well as combining different feature selection techniques to identify a condensed subset of the most important PSSM features that collectively determine the model performance.

Furthermore, and rather surprisingly, certain uncommon (not well known) descriptors such as DP-PSSM and EEDP achieved reasonable performances. In contrast, some popular descriptors such as PSSM-AC and PSSM-CC performed poorly in this assessment (Fig. S8C). Taken together, we recommend that PSSM matrix transformations be a requisite for the application of POSSUM-calculated PSSM features to protein class classification and prediction tasks. In addition, various PSSM-based descriptors should be comprehensively assessed based on a well-prepared benchmark dataset for the purpose of identifying the best-performing descriptors. As can be seen from Fig. S8D, feature groups based on the combinations of other individual types of descriptors achieved a high and stable prediction performance, suggesting that the combinations of descriptors are likely to further improve the performance. This can be further validated and examined by assessing the performance of different approaches in a real application, e.g. protein classification (Nanni, et al., 2014). Nanni *et al*. reported that models trained based on the fusion of PSSM-based features and sequence-derived features could outperform those trained using only PSSM features. In summary, the application of PSSM-based features to the prediction of bacterial secreted effectors serves as a demonstration of the usefulness of POSSUM, and validates the need to develop and make available such tool to the wider research community.

Finally, it is worth mentioning that bioinformatics applications of the variety of PSSM-based feature descriptors that can be calculated by POSSUM need not be restricted to prediction of bacterial secretion effector proteins; in fact, these versatile and informative PSSM features can be applied to address a wide range of sequence-based classification tasks related to e.g. protein sequence analysis, remote homology detection, protein family prediction, protein structure and function prediction, in combination with other complementary features. We hope the new bioinformatics tool presented in this work, POSSUM, can be adopted as a useful starting point to develop more accurate predictors for bioinformatics' open questions.

References:

Ahmad, S. and Sarai, A. (2005) PSSM-based prediction of DNA binding sites in proteins, *BMC bioinformatics*, **6**, 33.

An, J.Y., *et al*. (2016) Improving protein-protein interactions prediction accuracy using protein evolutionary information and relevance vector machine model, *Protein science : a publication of the Protein Society*.

Becker, J., Maes, F. and Wehenkel, L. (2013) On the encoding of proteins for disordered regions prediction, *PloS one*, **8**, e82252.

Bouziane, H., Messabih, B. and Chouarfia, A. (2011) Profiles and majority voting-based ensemble method for protein secondary structure prediction, *Evolutionary bioinformatics online*, **7**, 171-189.

Bui, V.M., *et al*. (2016) SOHSite: incorporating evolutionary information and physicochemical properties to identify protein S-sulfenylation sites, *BMC genomics*, **17 Suppl 1**, 9.

Bunnik, E.M., *et al*. (2011) Detection of inferred CCR5- and CXCR4-using HIV-1 variants and evolutionary intermediates using ultra-deep pyrosequencing, *PLoS pathogens*, **7**, e1002106.

Chauhan, J.S., *et al*. (2012) GlycoPP: a webserver for prediction of N- and O-glycosites in prokaryotic protein sequences, *PloS one*, **7**, e40155.

Chen, K., Kurgan, L.A. and Ruan, J. (2008) Prediction of protein structural class using novel evolutionary collocation‐based sequence representation, *Journal of computational chemistry*, **29**, 1596-1604.

Chen, S.A., *et al*. (2011) Prediction of transporter targets using efficient RBF networks with PSSM profiles and biochemical properties, *Bioinformatics*, **27**, 2062-2067.

Chen, Z., *et al*. (2011) Prediction of ubiquitination sites by using the composition of k-spaced amino acid pairs, *PloS one*, **6**, e22930.

Chen, Z., *et al*. (2015) Towards more accurate prediction of ubiquitination sites: a comprehensive review of current methods, tools and features, *Briefings in bioinformatics*, **16**, 640-657.

Cheng, C.W., *et al*. (2008) Predicting RNA-binding sites of proteins using support vector machines and evolutionary information, *BMC bioinformatics*, **9 Suppl 12**, S6.

Cheng, J. and Baldi, P. (2006) A machine learning information retrieval approach to protein fold recognition, *Bioinformatics*, **22**, 1456-1463.

Chou, K.C. (2000) Prediction of protein subcellular locations by incorporating quasi-sequence-order effect, *Biochemical and biophysical research communications*, **278**, 477-483.

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

Chou, K.C. and Shen, H.B. (2007) MemType-2L: a web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM, *Biochemical and biophysical research communications*, **360**, 339-345.

Cozzetto, D., *et al*. (2013) Protein function prediction by massive integration of evolutionary analyses and multiple data sources, *BMC bioinformatics*, **14**, S1.

Dehzangi, A., *et al*. (2013) A combination of feature extraction methods with an ensemble of different classifiers for protein structural class prediction problem, *IEEE/ACM transactions on computational biology and bioinformatics*, **10**, 564-575.

Deng, L., *et al*. (2009) Prediction of protein-protein interaction sites using an ensemble method, *BMC bioinformatics*, **10**, 426.

Ding, S., *et al*. (2014) A protein structural classes prediction method based on predicted secondary structure and PSI-BLAST profile, *Biochimie*, **97**, 60-65.

Dong, Q., Zhou, S. and Guan, J. (2009) A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation, *Bioinformatics*, **25**, 2655-2662.

Gao, Z.G., *et al*. (2016) Ens-PPI: A Novel Ensemble Classifier for Predicting the Interactions of Proteins Using Autocovariance Transformation from PSSM, *BioMed research international*, **2016**, 4563524.

Garg, A. and Gupta, D. (2008) VirulentPred: a SVM based prediction method for virulent proteins in bacterial pathogens, *BMC bioinformatics*, **9**, 62.

Guo, J., Lin, Y. and Liu, X. (2006) GNBSL: a new integrative system to predict the subcellular location for Gram-negative bacteria proteins, *Proteomics*, **6**, 5099-5105.

Guo, Y., *et al*. (2008) Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences, *Nucleic acids research*, **36**, 3025-3030.

Hayat, M. and Khan, A. (2012) MemHyb: predicting membrane protein types by hybridizing SAAC and PSSM, *Journal of theoretical biology*, **292**, 93-102.

Hong, Y., *et al*. (2011) Predicting protein folds with fold-specific PSSM libraries, *PloS one*, **6**, e20557.

Huang, H.L., *et al*. (2011) Predicting and analyzing DNA-binding domains using a systematic approach to identifying a set of informative physicochemical and biochemical properties, *BMC bioinformatics*, **12 Suppl 1**, S47.

Huang, Y.F. and Chen, S.Y. (2013) Extracting physicochemical features to predict protein secondary structure, *TheScientificWorldJournal*, **2013**, 347106.

Hwang, S., Gou, Z. and Kuznetsov, I.B. (2007) DP-Bind: a web server for sequence-based prediction of DNA-binding residues in DNA-binding proteins, *Bioinformatics*, **23**, 634-636.

Jensen, M.A., *et al*. (2006) A reliable phenotype predictor for human immunodeficiency virus type 1 subtype C based on envelope V3 sequences, *Journal of virology*, **80**, 4698-4704.

Jensen, M.A., *et al*. (2003) Improved coreceptor usage prediction and genotypic monitoring of R5-to-X4 transition by motif analysis of human immunodeficiency virus type 1 env V3 loop sequences, *Journal of virology*, **77**, 13376-13388.

Jeong, J.C., Lin, X. and Chen, X.W. (2011) On position-specific scoring matrix for protein function prediction, *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, **8**, 308-315.

Jiang, Y., *et al*. (2013) Prediction and Analysis of Post-Translational Pyruvoyl Residue Modification Sites from Internal Serines in Proteins, *PloS one*, **8**, e66678.

Jones, D.T. and Cozzetto, D. (2015) DISOPRED3: precise disordered region predictions with annotated protein-binding activity, *Bioinformatics*, **31**, 857-863.

Jones, D.T. and Ward, J.J. (2003) Prediction of disordered regions in proteins from position specific score matrices, *Proteins*, **53 Suppl 6**, 573-578.

Juan, E.Y., Jhang, J. and Li, W. (2008) Predicting protein subcellular localization using PsePSSM and suport vector machines. *Proceedings of the 11th Join Conference on Information Sciences*. pp. 1-6.

Juan, E.Y., *et al*. (2009) Predicting Protein Subcellular Localizations for Gram-Negative Bacteria using DP-PSSM and Support Vector Machines. *Complex, Intelligent and Software Intensive Systems, 2009. CISIS'09. International Conference on*. IEEE, pp. 836-841.

Kumar, M., Gromiha, M.M. and Raghava, G.P. (2007) Identification of DNA-binding proteins using support vector machines and evolutionary profiles, *BMC bioinformatics*, **8**, 463.

Kumar, M., Gromiha, M.M. and Raghava, G.P. (2008) Prediction of RNA binding sites in a protein using SVM and PSSM profile, *Proteins*, **71**, 189-194.

Li, D., *et al*. (2012) A novel structural position-specific scoring matrix for the prediction of protein secondary structures, *Bioinformatics*, **28**, 32-39.

Li, F., *et al*. (2015) GlycoMine: a machine learning-based approach for predicting N-, C- and O-linked glycosylation in the human proteome, *Bioinformatics*, **31**, 1411-1419.

Li, L., *et al*. (2014) PSSP-RFE: accurate prediction of protein structural class by recursive feature extraction from PSI-BLAST profile, physical-chemical property and functional annotations, *PloS one*, **9**, e92863.

Liu, T., *et al*. (2012) Accurate prediction of protein structural class using auto covariance transformation of PSI-BLAST profiles, *Amino acids*, **42**, 2243-2249.

Liu, T., Zheng, X. and Wang, J. (2010) Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile, *Biochimie*, **92**, 1330-1334.

Liu, Z.P., *et al*. (2010) Prediction of protein-RNA binding sites by a random forest method with combined features, *Bioinformatics*, **26**, 1616-1622.

Lobley, A., Sadowski, M.I. and Jones, D.T. (2009) pGenTHREADER and pDomTHREADER: new methods for improved protein fold recognition and superfamily discrimination, *Bioinformatics*, **25**, 1761-1767.

Melo, R., *et al*. (2016) A Machine Learning Approach for Hot-Spot Detection at Protein-Protein Interfaces, *International journal of molecular sciences*, **17**.

Mishra, N.K., Chang, J. and Zhao, P.X. (2014) Prediction of membrane transport proteins and their substrate specificities using primary sequence information, *PloS one*, **9**, e100278.

Murakami, Y. and Mizuguchi, K. (2010) Applying the Naive Bayes classifier with kernel density estimation to the prediction of protein-protein interaction sites, *Bioinformatics*, **26**, 1841-1848.

Murakami, Y., *et al*. (2010) PiRaNhA: a server for the computational prediction of RNA-binding residues in protein sequences, *Nucleic acids research*, **38**, W412-W416.

Nanni, L., Lumini, A. and Brahnam, S. (2014) An empirical study of different approaches for protein classification, *TheScientificWorldJournal*, **2014**, 236717.

Paliwal, K.K., *et al*. (2014) A tri-gram based feature extraction technique using linear probabilities of position specific scoring matrix for protein fold recognition, *IEEE transactions on nanobioscience*, **13**, 44-50.

Pu, X., *et al*. (2007) Prediction of membrane protein types from sequences and position-specific scoring matrices, *Journal of theoretical biology*, **247**, 259-265.

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

Saini, H., *et al*. Protein Fold Recognition Using Genetic Algorithm Optimized Voting Scheme and Profile Bigram.

Seclen, E., *et al*. (2011) High concordance between the position-specific scoring matrix and geno2pheno algorithms for genotypic interpretation of HIV-1 tropism: V3 length as the major cause of disagreement, *Journal of clinical microbiology*, **49**, 3380-3382.

Selvaraj, M., *et al*. (2016) BacHbpred: Support Vector Machine Methods for the Prediction of Bacterial Hemoglobin-Like Proteins, *Advances in bioinformatics*, **2016**, 8150784.

Sharma, A., *et al*. (2013) A feature extraction technique using bi-gram probabilities of position specific scoring matrix for protein fold recognition, *Journal of theoretical biology*, **320**, 41-46.

Shimizu, K., Hirose, S. and Noguchi, T. (2007) POODLE-S: web application for predicting protein disorder by using physicochemical features and reduced amino acid set of a position-specific scoring matrix, *Bioinformatics*, **23**, 2337-2338.

Su, C.T., Chen, C.Y. and Ou, Y.Y. (2006) Protein disorder prediction by condensed PSSM considering propensity for order or disorder, *BMC bioinformatics*, **7**, 319.

Tang, Z., *et al*. (2011) Improving the performance of beta-turn prediction using predicted shape strings and a two-layer support vector machine model, *BMC bioinformatics*, **12**, 283.

Tao, P., *et al*. (2015) Prediction of protein structural class using tri-gram probabilities of position-specific scoring matrix and recursive feature elimination, *Amino acids*, **47**, 461-468.

Walia, R.R., *et al*. (2012) Protein-RNA interface residue prediction using machine learning: an assessment of the state of the art, *BMC bioinformatics*, **13**, 1.

Wang, L., *et al*. (2010) BindN+ for accurate prediction of DNA and RNA-binding residues from protein sequence features, *BMC systems biology*, **4 Suppl 1**, S3.

Wang, X. and Li, G.-Z. (2013) Multilabel learning via random label selection for protein subcellular multilocations prediction, *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, **10**, 436-446.

Wass, M.N. and Sternberg, M.J. (2008) ConFunc--functional annotation in the twilight zone, *Bioinformatics*, **24**, 798-806.

Xia, X.Y., *et al*. (2012) Accurate prediction of protein structural class, *PloS one*, **7**, e37653.

Xiao, N., *et al*. (2015) protr/ProtrWeb: R package and web server for generating various numerical representation schemes of protein sequences, *Bioinformatics*, btv042.

Xie, D., *et al*. (2005) LOCSVMPSI: a web server for subcellular localization of eukaryotic proteins using SVM and profile of PSI-BLAST, *Nucleic acids research*, **33**, W105-W110.

Yan, R., *et al*. (2015) Prediction of structural features and application to outer membrane protein identification, *Scientific reports*, **5**, 11586.

Yang, X., *et al*. (2013) Effective identification of Gram-negative bacterial type III secreted effectors using position-specific residue conservation profiles, *PloS one*, **8**, e84439.

Zahiri, J., *et al*. (2013) PPIevo: protein-protein interaction prediction from PSSM based evolutionary information, *Genomics*, **102**, 237-242.

Zhang, L., Zhao, X. and Kong, L. (2014) Predict protein structural class for low-similarity sequences by evolutionary difference information into the general form of Chou's pseudo amino acid composition, *Journal of theoretical biology*, **355**, 105-110.

Zhang, N., *et al*. (2014) Discriminating between lysine sumoylation and lysine acetylation using mRMR feature selection and analysis, *PloS one*, **9**, e107464.

Zhang, S., Ye, F. and Yuan, X. (2012) Using principal component analysis and support vector machine to predict protein structural class for low-similarity sequences via PSSM, *Journal of biomolecular structure & dynamics*, **29**, 634-642.

Zou, L., Nan, C. and Hu, F. (2013) Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles, *Bioinformatics*, **29**, 3135-3142.

Appendix 6 - Supplementary information for Chapter 4.2

DIFFUSER: A distributed framework for high-throughput generation of machine-learning features from DNA, RNA and protein sequences

Supplementary file

Table S1. The details of various features that could be generated by DIFFUSER based on the protein sequence.

Category	Method	Description	Dimension	Reference
Simple sequence derived features	Kmer[a]	Sub-sequences of length k contained within a protein sequence	20^k (k=2)	(1), (2)
	DR	Distance-based residues	20+20*20*d (d=3)	(3), (2)
	Distance Pair (DP)	The PseAAC using the distance-pairs and reduced alphabet	$n+dn^2$ (n=14, d=3)	(4), (2)
	EAAC	Enhanced amino acid composition	(peptide-w+1) *20 (w = 5)	(5)
	CKSAAP	Composition of k-spaced amino acid pairs	400*(g+1) (g=5)	(6), (5)
	DDE	The dipeptide deviation from expected mean	400	(7), (5)
	GAAC	Grouped amino acid composition	5	(8), (5)
	EGAAC	Enhanced grouped amino acid composition	5*(peptide-w+1) (w=5)	(5)
	CKSAAGP	Composition of k-spaced amino acid group pairs	25*(g+1) (g=5)	(5)
	GDPC	Grouped dipeptide composition	25	(5)
	GTPC	Grouped tripeptide composition	125	(5)
	BINARY	The binary encoding of amino acids	20*peptide	(9), (10)
	NUM	Numerical values by mapping amino acids in an alphabetical order	peptide	(10)
Physicochemical features	AC	Autocovariance	N*lag (lag=2, N=3)	(11), (2)
	CC	Cross-covariance	N*(N-1) * lag (lag=2, N=3)	(11), (2)
	ACC	Auto-cross-covariance	N*N*lag (lag=2, N=3)	(11), (2)
	PDT	Physicochemical distance based transformation	531*lamada (lamada=1)	(12), (2)
	Moran	Moran	lag*N (lag=30, N=8)	(13), (5)
	Geary	Geary	lag*N (lag=30, N=8)	(14), (5)
	NMBroto	Normalized Moreau-Broto	lag*N (lag=30, N=8)	(15), (5)
	PAAC	Pseudo-amino acid composition	20+lamada (lamada=30)	(16), (5)
	APAAC	Amphiphilic PAAC	20+N*lamada (N=3, lamada=30)	(16), (5)
	CTDC	The composition among CTD (composition, transition and distribution)	N*3 (N=13)	(17), (5)
	CTDT	The transition among CTD	N*3 (N=13)	(17), (5)
	CTDD	The distribution among CTD	N*15 (N=13)	(17), (5)
	CTriad	Conjoint Triad	343	(18), (5)
	KSCTriad	Conjoint k-spaced Triad	(k+1) * 343 (k=0)	(18), (5)
	SOCNumber	Sequence-order-coupling number	lag*2 (lag=30)	(19), (5)
	QSOrder	The quasi-sequence-order feature	20+20+lag*2 (lag=30)	(19), (5)
	KNNprotein	K-nearest neighbor for proteins	60	(5)
	KNNpeptide	K-nearest neighbor for peptides	60	(19), (5)
	AAINDEX	Feature based on the amino acid index database (AAindex)	peptide*531	(20), (5)
	BLOSUM62	Feature based on the BLOSUM62 matrix	peptide*20	(21), (5)
	ZSCALE	Feature based on the transformation of each amino acid into five physicochemical variables	peptide*5	(22), (5)
	PseKRAAC (type1 to type16)	Pseudo k-tuple reduced amino acid composition	$raactype^k$ (k =2)	(23), (5)
	EBGW	Feature based on grouped weight	lamada*3 (lamada=11)	(10)
PSSM based features	AAC-PSSM	Extension of the concept of traditional AAC feature encoding method from the primary sequence to the PSSM profile	20	(24), (25)
	D-FPSSM	Calculation of the amino acid distribution by summing up the entries within each column of the PSSM profile	20	(26), (25)
	smoothed-PSSM	Feature based on the PSSM profile segmented by soothed windows	sliding_window * 20 (sliding _window=50)	(27), (28)
	AB-PSSM	Feature based on averaged blocks of the PSSM profile	400	(29), (25)

	PSSM-composition	Feature by converting the original PSSM profile into a 20*20 matrix through summing up all rows of the same amino acid	400	(30), (28)
	RPM-PSSM	Feature based a 'filtered' PSSM that is generated by a residue probing method	400	(29), (25)
	S-FPSSM	Feature based on the matrix transformation of a 'filtered' PSSM (called FPSSM)	400	(26), (31)
	DPC-PSSM	Extension of the concept of traditional DPC from the primary sequence into the PSSM profile	400	(24), (31)
	k-separated-bigrams-PSSM	Extension of the concept of traditional Kmer from the primary sequence into the PSSM profile	400	(32), (25)
	tri-gram-PSSM	Feature based on a tri-gram probability matrix composed of the probabilities of individual tri-grams, which is computed from the PSSM linear probabilities	8000	(33), (25)
	EEDP	Feature based on the direct transformation of an ED-PSSM that is generated by an evolutionary formula (EDF)	400	(34), (25)
	TPC-PSSM	Feature based on the transition probability matrix (TPM), which is extended from the PSSM to avoid complete loss of the sequence-order information	400	(35), (25)
	EDP	Feature based on the averaging of an ED-PSSM that could be generated by an evolutionary formula (EDF)	20	(34), (25)
	RPSSM	Feature by calculating the correlation between two adjacent residues via importing the transition probability matrix into the PSSM profile	110	(36), (25)
	Pse-PSSM	Feature based on a set of PSSM transformations and dimension normalization	40	(37), (31)
	DP-PSSM	Extension of the Pse-PSSM feature encoding method to describe the relationship of an amino acid and its α-th succeeding amino acid	(α+1)* 40 (α=5)	(38), (25)
	PSSM-AC	Extension of the concept of autocross-covariance transformation (AC) from the primary sequence into the PSSM profile	lg*20 (lg=10)	(11), (28)
	PSSM-CC	Extension of the concept of autocross-covariance transformation (CC) from the primary sequence into the PSSM profile	lg*20 (lg=10)	(11), (25)
	AADP-PSSM	Feature in combination of the AAC-PSSM and DPC-PSSM	420	(24), (25)
	AATP	Feature in combination of the AAC-PSSM and TPC (PSSM)	420	(35), (25)
	MEDP	Feature in combination of the EEDP and EDP	420	(34), (25)
Predicted Structure based features	SSEC	Feature based on the secondary structure elements (content)	3	(5)
	SSEB	Feature based on the secondary structure elements (binary)	peptide*3	(5)
	Disorder	Feature based on the disorder profile	peptide	(39), (5)
	DisorderB	Feature based on the disorder profile (binary)	2	(5)
	DisorderC	Feature based on the disorder profile (content)	peptide*2	(5)
	ASA	Feature based on the accessible surface area	peptide	(5)
	TA	Feature based on the torsional angles	peptide*2	(5)
Other profile based features	LOGO	Feature based on the occurrence frequencies of amino acids calculated by the Two Sample Logo program	peptide	(10)
	LOGO-P-Value	Extension of the LOGO feature by integrating the P-value	peptide	In this work
	LOGO-BLOSUM62	Extension of the LOGO feature by the BLOSUM62 matrix	peptide	In this work

Note: [a]Kmer indicates the frequency of *k* amino acids. Specially, Kmer is same as amino acid composition (AAC) when k is set to 1, while as dipeptide composition (DPC) when k is set to 2.

[b]The values in parentheses represent the default values for the parameters. This applies to Tables 2 and 3.

Table S2. The details of various features that could be generated by DIFFUSER based on the DNA sequence.

Category	Method[a]	Description	Dimension[b]	Ref
Simple sequence derived features	Kmer	Sub-sequences of length k contained within a DNA sequence	4^k(k=2)	(40), (41)
	RevKmer	The reverse complementary k-mer	2^{2k-1} (k = 1, 3, …) $2^{2k-1}+2^{k-1}$ (k = 2, 4, …) default k =2	(40), (42)
	IDKmer	Extension of the Kmer with the increment of diversity	2k (k=6)	(43), (44)
	Mismatch	Feature based on the occurrences of k-mers, allowing at most m mismatches	4^k (k=3, m=1)	(45), (46)
	Sub-sequence	Feature based on the occurrences of k-mers, allowing non-contiguous matches	4^k (k=3, d=1)	(46), (47)
Physicoch emical features	DAC	Dinucleotide-based autocovariance	N*lag (lag=2, N=6)	(11), (2)
	DCC	Dinucleotide-based cross-covariance	N*(N-1)*lag (lag=2, N=6)	(11), (2)
	DACC	Dinucleotide-based auto-cross-covariance	N*N*lag (lag=2, N=6)	(11), (2)
	TAC	Trinucleotide-based autocovariance	N*lag (lag=2, N=2)	(11), (2)
	TCC	Trinucleotide-based cross-covariance	N*(N-1)*lag (lag=2, N=2)	(11), (2)
	TACC	Trinucleotide-based auto-cross-covariance	N*N*lag (lag=2, N=2)	(11), (2)
	MAC	Feature based on the Moran autocorrelation	N (N=6)	(15), (2)
	GAC	Feature based on the Geary autocorrelation	N (N=6)	(14), (2)
	NMBAC	Feature based on the normalized Moreau–Broto autocorrelation	N (N=6)	(13), (2)
	PseDNC	Feature based on the pseudo dinucleotide composition	16+lamada (lamada=2)	(48), (2)
	PseKNC	Feature based on the pseudo k-tuple nucleotide composition	4^k+lamada (k=3, lamada=2)	(49), (2)
	PC-PseDNC-General	General parallel correlation of the pseudo dinucleotide composition	16+lamada (lamada=2)	(50), (2)
	PC-PseTNC-General	General parallel correlation of the pseudo trinucleotide composition	64+lamada (lamada=2)	(50), (2)
	SC-PseDNC-General	General series correlation of the pseudo dinucleotide composition	16+lamada*N (lamada=2, N=6)	(50), (2)
	SC-PseTNC-General	General series correlation of the pseudo trinucleotide composition	64+lamada*N (lamada=2, N=2)	(50), (2)

Table S3. The details of various features that could be generated by DIFFUSER based on the RNA sequence.

Category	Method[a]	Description	Dimension[b]	Reference
Sequence based features	Kmer	Sub-sequences of length k contained within a RNA sequence	4^k (k=2)	(51), (2)
	Mismatch	Feature based on the occurrences of k-mers, allowing at most m mismatches	4^k (k=3, m=1)	(45), (46)
	Sub-sequence	Feature based on the occurrences of k-mers, allowing non-contiguous matches	4^k (k=3, d=1)	(46), (47)
Physicochemical features	DAC	Dinucleotide-based autocovariance	N*lag (lag=2, N=6)	(11), (2)
	DCC	Dinucleotide-based cross-covariance	N*(N-1)*lag (lag=2, N=6)	(11), (2)
	DACC	Dinucleotide-based auto-cross-covariance	N*N*lag (lag=2, N=6)	(11), (2)
	MAC	Feature based on the Moran autocorrelation	N (N=6)	(15), (2)
	GAC	Feature based on the Geary autocorrelation	N (N=6)	(14), (2)
	NMBAC	Feature based on the normalized Moreau–Broto autocorrelation	N (N=6)	(13), (2)
	PC-PseDNC-General	General parallel correlation of the pseudo dinucleotide composition	16+lamada (lamada=2)	(52), (2)
	SC-PseDNC-General	General series correlation of the pseudo dinucleotide composition	16+lamada*N (lamada=2, N=6)	(52), (2)
Predicted Structure based features	Triplet	Feature based on the local structure–sequence triplet element	32	(53), (2)
	PseSSC	Feature based on the pseudo-structure status composition	10^k+l (k=2, l=2)	(54), (2)
	PseDPC	Feature based on the pseudo-distance structure status pair composition	10+100k+l (k=0, l=2)	(55), (2)

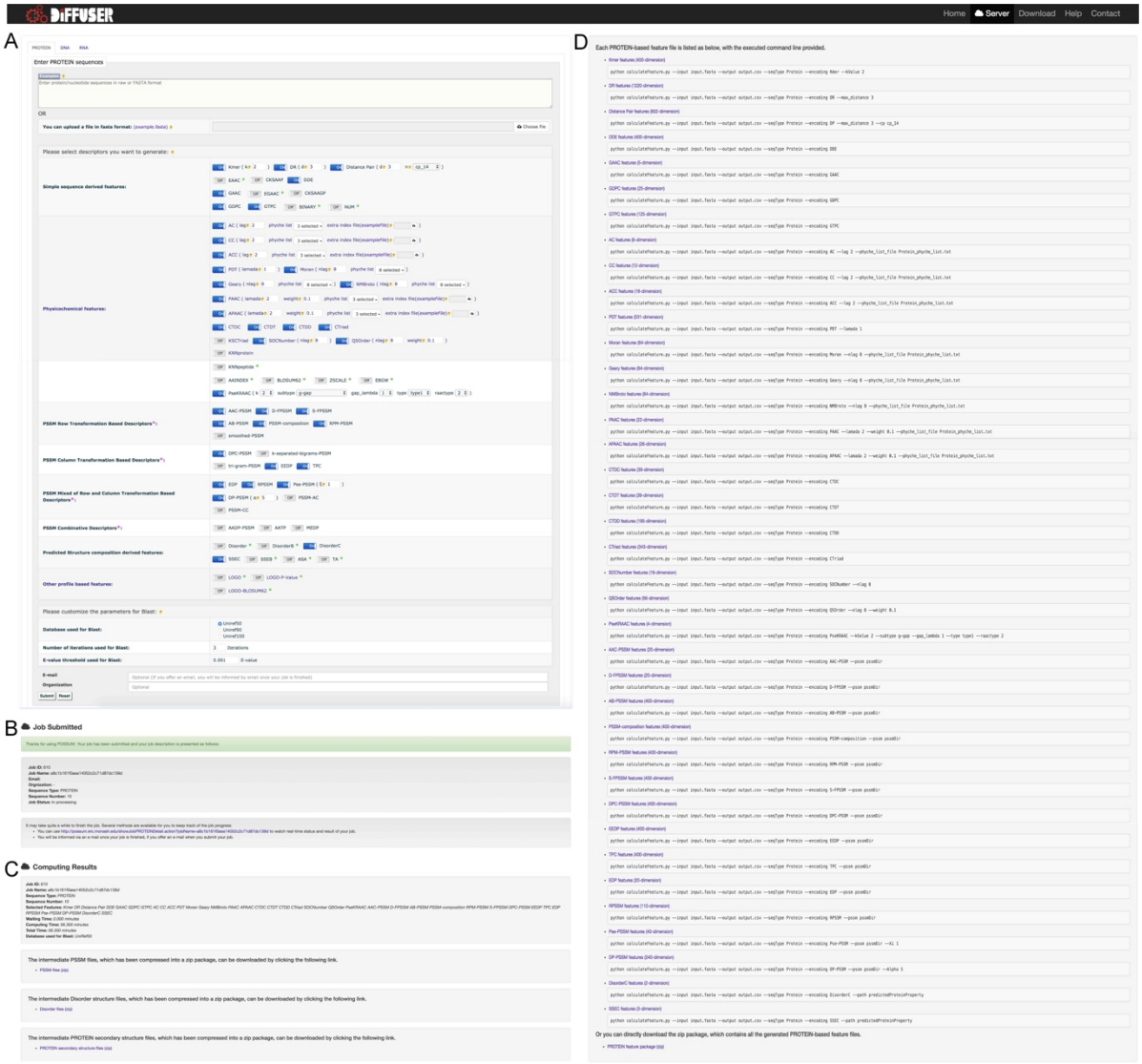


Fig. S1. An example of the user interface of the DIFFUSER server: (A) Webpage displaying three separate panels for users' submission options; (B) Webpage summarizing the submitted job information and indicating the way to check its status; and (C) The result page containing the generated features and the intermediate files and (D) additionally listing the corresponding command line with all parameters needed to generate the same features locally.

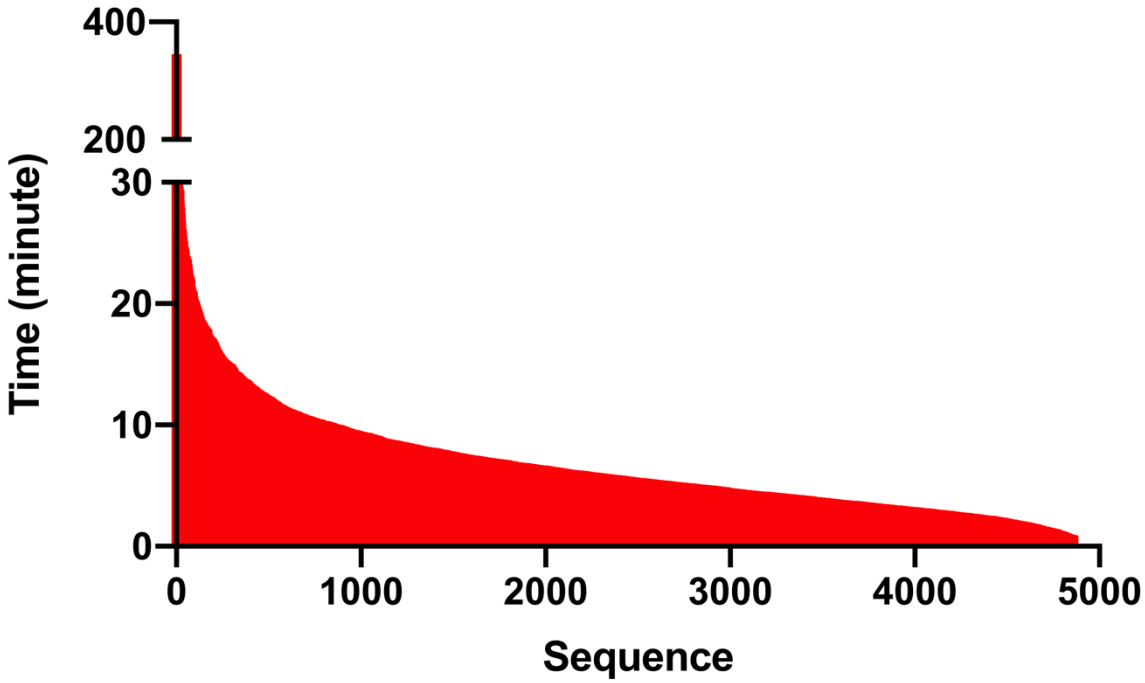


Fig. S2. Distribution of consuming time of 4859 sequences in the *Klebsiella pneumoniae* MGH78578. Each bar represents the consuming time of a sequence processed by the diffuser in one node mode. The median consuming time among all the 4859 sequences is 5.783 minutes. Removing the sequences with unexpectedly long consuming time (more than 10 minutes), the median consuming time among the rest 3985 sequences is 3.8 minutes.

References

1. Liu, B., Wang, X., Lin, L., Dong, Q. and Wang, X. (2008) A discriminative method for protein remote homology detection and fold recognition combining Top-n-grams and latent semantic analysis. *BMC bioinformatics*, **9**, 510.
2. Liu, B. (2017) BioSeq-Analysis: a platform for DNA, RNA and protein sequence analysis based on machine learning approaches. *Briefings in bioinformatics*.
3. Liu, B., Xu, J., Zou, Q., Xu, R., Wang, X. and Chen, Q. (2014) Using distances between Top-n-gram and residue pairs for protein remote homology detection. *BMC bioinformatics*, **15 Suppl 2**, S3.
4. Liu, B., Xu, J., Lan, X., Xu, R., Zhou, J., Wang, X. and Chou, K.C. (2014) iDNA-Prot|dis: identifying DNA-binding proteins by incorporating amino acid distance-pairs and reduced alphabet profile into the general pseudo amino acid composition. *PloS one*, **9**, e106691.
5. Chen, Z., Zhao, P., Li, F., Leier, A., Marquez-Lago, T.T., Wang, Y., Webb, G.I., Smith, A.I., Daly, R.J., Chou, K.C. *et al.* (2018) iFeature: a Python package and web server for features extraction and selection from protein and peptide sequences. *Bioinformatics*, **34**, 2499-2502.
6. Chen, K., Kurgan, L. and Rahbari, M. (2007) Prediction of protein crystallization using collocation of amino acid pairs. *Biochemical and biophysical research communications*, **355**, 764-769.
7. Saravanan, V. and Gautham, N. (2015) Harnessing Computational Biology for Exact Linear B-Cell Epitope Prediction: A Novel Amino Acid Composition-Based Feature Descriptor. *OMICS*, **19**, 648-658.
8. Lee, T.Y., Lin, Z.Q., Hsieh, S.J., Bretana, N.A. and Lu, C.T. (2011) Exploiting maximal dependence decomposition to identify conserved motifs from a group of aligned signal sequences. *Bioinformatics*, **27**, 1780-1787.
9. Chen, Z., Chen, Y.Z., Wang, X.F., Wang, C., Yan, R.X. and Zhang, Z. (2011) Prediction of ubiquitination sites by using the composition of k-spaced amino acid pairs. *PloS one*, **6**, e22930.
10. Zhang, Y., Xie, R., Wang, J., Leier, A., Marquez-Lago, T.T., Akutsu, T., Webb, G.I., Chou, K.C. and Song, J. (2018) Computational analysis and prediction of lysine malonylation sites by exploiting informative features in an integrative machine-learning framework. *Briefings in bioinformatics*.
11. Dong, Q., Zhou, S. and Guan, J. (2009) A new taxonomy-based protein fold recognition approach based on autocross-covariance transformation. *Bioinformatics*, **25**, 2655-2662.
12. Liu, B., Wang, X., Chen, Q., Dong, Q. and Lan, X. (2012) Using amino acid physicochemical distance transformation for fast protein remote homology detection. *PloS one*, **7**, e46633.
13. Feng, Z.P. and Zhang, C.T. (2000) Prediction of membrane protein types based on the hydrophobic index of amino acids. *J Protein Chem*, **19**, 269-275.
14. Sokal, R.R. and Thomson, B.A. (2006) Population structure inferred by local spatial autocorrelation: an example from an Amerindian tribal population. *Am J Phys Anthropol*, **129**, 121-131.
15. Horne, D.S. (1988) Prediction of protein helix content from an autocorrelation analysis of sequence hydrophobicities. *Biopolymers*, **27**, 451-477.
16. Cao, D.S., Xu, Q.S. and Liang, Y.Z. (2013) propy: a tool to generate various modes of Chou's PseAAC. *Bioinformatics*, **29**, 960-962.

17. Lin, J.M., Collins, P.J., Trinklein, N.D., Fu, Y., Xi, H., Myers, R.M. and Weng, Z. (2007) Transcription factor binding and modified histones in human bidirectional promoters. *Genome research*, **17**, 818-827.
18. Shen, J., Zhang, J., Luo, X., Zhu, W., Yu, K., Chen, K., Li, Y. and Jiang, H. (2007) Predicting protein-protein interactions based only on sequences information. *Proceedings of the National Academy of Sciences of the United States of America*, **104**, 4337-4341.
19. Chou, K.C. (2000) Prediction of protein subcellular locations by incorporating quasi-sequence-order effect. *Biochemical and biophysical research communications*, **278**, 477-483.
20. Tung, C.W. and Ho, S.Y. (2008) Computational identification of ubiquitylation sites from protein sequences. *BMC bioinformatics*, **9**, 310.
21. Lee, T.Y., Chen, S.A., Hung, H.Y. and Ou, Y.Y. (2011) Incorporating distant sequence features and radial basis function networks to identify ubiquitin conjugation sites. *PloS one*, **6**, e17331.
22. Chen, Y.Z., Chen, Z., Gong, Y.A. and Ying, G. (2012) SUMOhydro: a novel method for the prediction of sumoylation sites based on hydrophobic properties. *PloS one*, **7**, e39195.
23. Zuo, Y., Li, Y., Chen, Y., Li, G., Yan, Z. and Yang, L. (2017) PseKRAAC: a flexible web server for generating pseudo K-tuple reduced amino acids composition. *Bioinformatics*, **33**, 122-124.
24. Liu, T., Zheng, X. and Wang, J. (2010) Prediction of protein structural class for low-similarity sequences using support vector machine and PSI-BLAST profile. *Biochimie*, **92**, 1330-1334.
25. Wang, J., Yang, B., Revote, J., Leier, A., Marquez-Lago, T.T., Webb, G., Song, J., Chou, K.C. and Lithgow, T. (2017) POSSUM: a bioinformatics toolkit for generating numerical sequence feature descriptors based on PSSM profiles. *Bioinformatics*, **33**, 2756-2758.
26. Zahiri, J., Yaghoubi, O., Mohammad-Noori, M., Ebrahimpour, R. and Masoudi-Nejad, A. (2013) PPIevo: protein-protein interaction prediction from PSSM based evolutionary information. *Genomics*, **102**, 237-242.
27. Cheng, C.W., Su, E.C., Hwang, J.K., Sung, T.Y. and Hsu, W.L. (2008) Predicting RNA-binding sites of proteins using support vector machines and evolutionary information. *BMC bioinformatics*, **9 Suppl 12**, S6.
28. Wang, J., Yang, B., An, Y., Marquez-Lago, T., Leier, A., Wilksch, J., Hong, Q., Zhang, Y., Hayashida, M., Akutsu, T. *et al.* (2017) Systematic analysis and prediction of type IV secreted effector proteins by machine learning approaches. *Briefings in bioinformatics*.
29. Jeong, J.C., Lin, X. and Chen, X.W. (2011) On position-specific scoring matrix for protein function prediction. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, **8**, 308-315.
30. Zou, L., Nan, C. and Hu, F. (2013) Accurate prediction of bacterial type IV secreted effectors using amino acid composition and PSSM profiles. *Bioinformatics*, **29**, 3135-3142.
31. Wang, J., Yang, B., Leier, A., Marquez-Lago, T.T., Hayashida, M., Rocker, A., Zhang, Y., Akutsu, T., Chou, K.C., Strugnell, R.A. *et al.* (2018) Bastion6: a bioinformatics approach for accurate prediction of type VI secreted effectors. *Bioinformatics*, **34**, 2546-2555.
32. Saini, H., Raicar, G., Lal, S., Dehzangi, A., Imoto, S. and Sharma, A. Protein Fold Recognition Using Genetic Algorithm Optimized Voting Scheme and Profile Bigram.

33. Paliwal, K.K., Sharma, A., Lyons, J. and Dehzangi, A. (2014) A tri-gram based feature extraction technique using linear probabilities of position specific scoring matrix for protein fold recognition. *IEEE transactions on nanobioscience*, **13**, 44-50.
34. Zhang, L., Zhao, X. and Kong, L. (2014) Predict protein structural class for low-similarity sequences by evolutionary difference information into the general form of Chou's pseudo amino acid composition. *Journal of theoretical biology*, **355**, 105-110.
35. Zhang, S., Ye, F. and Yuan, X. (2012) Using principal component analysis and support vector machine to predict protein structural class for low-similarity sequences via PSSM. *Journal of biomolecular structure & dynamics*, **29**, 634-642.
36. Ding, S., Li, Y., Shi, Z. and Yan, S. (2014) A protein structural classes prediction method based on predicted secondary structure and PSI-BLAST profile. *Biochimie*, **97**, 60-65.
37. Chou, K.C. and Shen, H.B. (2007) MemType-2L: a web server for predicting membrane proteins and their types by incorporating evolution information through Pse-PSSM. *Biochemical and biophysical research communications*, **360**, 339-345.
38. Juan, E.Y.T., Li, W.J., Jhang, J.H. and Chiu, C.H. (2009) Predicting Protein Subcellular Localizations for Gram-Negative Bacteria using DP-PSSM and Support Vector Machines. *Cisis: 2009 International Conference on Complex, Intelligent and Software Intensive Systems, Vols 1 and 2*, 836-841.
39. Cai, Y., Huang, T., Hu, L., Shi, X., Xie, L. and Li, Y. (2012) Prediction of lysine ubiquitination with mRMR feature selection and analysis. *Amino acids*, **42**, 1387-1395.
40. Noble, W.S., Kuehn, S., Thurman, R., Yu, M. and Stamatoyannopoulos, J. (2005) Predicting the in vivo signature of human gene regulatory sequences. *Bioinformatics*, **21 Suppl 1**, i338-343.
41. Liu, B., Fang, L., Wang, S., Wang, X., Li, H. and Chou, K.C. (2015) Identification of microRNA precursor with the degenerate K-tuple or Kmer strategy. *Journal of theoretical biology*, **385**, 153-159.
42. Gupta, S., Dennis, J., Thurman, R.E., Kingston, R., Stamatoyannopoulos, J.A. and Noble, W.S. (2008) Predicting human nucleosome occupancy from primary sequence. *PLoS computational biology*, **4**, e1000134.
43. Zhang, L. and Luo, L. (2003) Splice site prediction with quadratic discriminant analysis using diversity measure. *Nucleic acids research*, **31**, 6214-6220.
44. Chen, W., Luo, L. and Zhang, L. (2010) The organization of nucleosomes around splice sites. *Nucleic acids research*, **38**, 2788-2798.
45. Leslie, C.S., Eskin, E., Cohen, A., Weston, J. and Noble, W.S. (2004) Mismatch string kernels for discriminative protein classification. *Bioinformatics*, **20**, 467-476.
46. El-Manzalawy, Y., Dobbs, D. and Honavar, V. (2008) Predicting flexible length linear B-cell epitopes. *Computational systems bioinformatics / Life Sciences Society. Computational Systems Bioinformatics Conference*, **7**, 121-132.
47. Luo, L., Li, D., Zhang, W., Tu, S., Zhu, X. and Tian, G. (2016) Accurate Prediction of Transposon-Derived piRNAs by Integrating Various Sequential and Physicochemical Features. *PloS one*, **11**, e0153268.
48. Chen, W., Feng, P.M., Lin, H. and Chou, K.C. (2013) iRSpot-PseDNC: identify recombination spots with pseudo dinucleotide composition. *Nucleic acids research*, **41**, e68.
49. Guo, S.H., Deng, E.Z., Xu, L.Q., Ding, H., Lin, H., Chen, W. and Chou, K.C. (2014) iNuc-PseKNC: a sequence-based predictor for predicting nucleosome positioning in genomes with pseudo k-tuple nucleotide composition. *Bioinformatics*, **30**, 1522-1529.

50. Liu, B., Zhang, D., Xu, R., Xu, J., Wang, X., Chen, Q., Dong, Q. and Chou, K.C. (2014) Combining evolutionary information extracted from frequency profiles with sequence-based kernels for protein remote homology detection. *Bioinformatics*, **30**, 472-479.
51. Wei, L., Liao, M., Gao, Y., Ji, R., He, Z. and Zou, Q. (2014) Improved and Promising Identification of Human MicroRNAs by Incorporating a High-Quality Negative Set. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, **11**, 192-201.
52. Chen, W., Zhang, X., Brooker, J., Lin, H., Zhang, L. and Chou, K.C. (2015) PseKNC-General: a cross-platform package for generating various modes of pseudo nucleotide compositions. *Bioinformatics*, **31**, 119-120.
53. Xue, C., Li, F., He, T., Liu, G.P., Li, Y. and Zhang, X. (2005) Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. *BMC bioinformatics*, **6**, 310.
54. Liu, B., Fang, L., Liu, F., Wang, X., Chen, J. and Chou, K.C. (2015) Identification of real microRNA precursors with a pseudo structure status composition approach. *PloS one*, **10**, e0121501.
55. Liu, B., Fang, L., Liu, F., Wang, X. and Chou, K.C. (2016) iMiRNA-PseDPC: microRNA precursor identification with a pseudo distance-pair composition approach. *Journal of biomolecular structure & dynamics*, **34**, 223-235.