Philippine Journal of Science
148 (S1): 237-245, Special Issue on Genomics
ISSN 0031 - 7683
Date Received: 19 Mar 2019

Predicting the Molecular Targets of Conopeptides by using Principal Component Analysis and Multiclass Logistic Regression

Xavier Eugenio Asuncion[1,4], **Abdul-Rashid Sampaco III**[2,4],
Henry Adorna[2,4], **Joselito Magadia**[3,4], **Vena Pearl Boñgolan**[2,4], and **Arturo Lluisma**[1,4]*

[1]The Marine Science Institute, University of the Philippines
Diliman, Quezon City 1101 Philippines
[2]Department of Computer Science, University of the Philippines
Diliman, Quezon City 1101 Philippines
[3]School of Statistics, University of the Philippines
Diliman, Quezon City 1101 Philippines
[4]Philippine Genome Center, University of the Philippines
Diliman, Quezon City 1101 Philippines

Computational tools for inferring molecular targets from the primary structure would be crucial in exploiting the wealth of sequence data. In this work, we have developed a computational method in predicting the molecular targets of conopeptides given only their primary structures. Our proposed method makes use of descriptors calculated from the primary structure, and machine learning to create a model that can identify the most likely target among five target types. Our proposed method is based on principal component analysis (PCA) and multiclass logistic regression algorithms. PCA was used to reduce the dimensionality of the data, which resulted in the improvement of the model's performance. By using nested cross-validation, a multiclass logistic regression with PCA was able to achieve an accuracy of 89% – outperforming other classical machine learning algorithms. We also compared our proposed method to a basic sequence similarity search and found that our method produced better overall results. These results suggest that our proposed method may be used as a complementary method to sequence similarity search in identifying candidate targets of newly sequenced and isolated conopeptides.

Keywords: bioinformatics, conopeptides, machine learning

INTRODUCTION

With around 800 cone snail species, *Conus* has emerged as one of the most promising sources of marine drugs (Himaya and Lewis 2018, Gao *et al.* 2017, Prashanth *et al.* 2014). Like other venomous organisms, cone snails use their extremely potent venom to capture their prey and to protect themselves from potential predators. Over the course of their evolution, each *Conus* species has developed a unique set of bioactive peptides, which are commonly referred to as conotoxins or conopeptides. Due to their small size (typically less than 5 kDa), diversity (around 100 per species with little overlap between species), and their high specificity to an array of biological targets, conopeptides serve as excellent templates for the design of novel drugs (Prashanth *et al.* 2014, Lewis *et al.* 2012). In 2004, the first conopeptide-derived drug – Prialt – was approved by the US Food and Drug Administration (Pope and Deer 2013) and, since

*Corresponding Author: aolluisma@up.edu.ph

then, several other conopeptides reached the advanced stages of clinical trials (Nielsen *et al.* 2005, Lubbers *et al.* 2005, Barton *et al.* 2004, Sandall *et al.* 2003). Thus, with more than 80,000 estimated conopeptides and only 6260 conopeptides known to date (Kaas *et al.* 2007, 2011), cone snail venoms still hold great promise for the discovery of new drug leads.

In order to accelerate the discovery of conopeptides, a combination of transcriptomics and proteomics approaches is often integrated into marine biodiscovery pipelines (Degueldre *et al.* 2017, Jin *et al.* 2015, Lavergne *et al.* 2015, Safavi-Hemami *et al.* 2014, Violette *et al.* 2012). Unlike in the traditional approach to venom peptide identification (*i.e.*, using reverse-phase HPLC followed by the Edman sequence analysis), a multi-omics strategy also known as venomics enables the rapid and high-throughput identification of venom components – including peptides, enzymes, and other biomolecules (Oldrati *et al.* 2016). Despite the increased pace in conopeptide identification, the molecular targets of most of these sequences remain unknown to date (Himaya and Lewis 2018). Only less than 1% of the identified conopeptides have available structural and functional data because of the tedious and expensive nature of laboratory experiments. Hence, computational approaches offer complementary means to help fill these gaps and limitations in experimental methods.

Since the identification of biological activity still remains the bottleneck in developing conopeptide-based drugs, a number of groups have successfully applied machine learning techniques to predict the biological targets of conopeptides given only the mature sequence information (Yuan *et al.* 2013, Ding *et al.* 2014, Wu *et al.* 2016, Zhang *et al.* 2016, Wang *et al.* 2017). The main idea behind machine learning approaches in sequence-based problems is to represent the sequences as vectors of sequence-derived descriptors, which a supervised machine learning algorithm can use as features to develop a function mapping the primary sequence to their corresponding molecular properties (Walsh *et al.* 2015). Yuan *et al.* (2013) first proposed a model based on binomial distribution and radial basis function network to predict the three types of ion-channel targeted conotoxins. By using dipeptide composition as features and jackknife cross-validation, the model of Yuan *et al.* (2013) was able to achieve an overall accuracy of 89.3%. Subsequently, Ding *et al.* (2014) developed a model called iCTX-type to improve the prediction of biological targets of conopeptides using a combination of incremental feature selection strategy and support vector machine (SVM). With the use of dipeptide mode of pseudo-amino composition (PseAAC) as features, the iCTX-type model reached an overall accuracy of 91% via jackknife cross-validation. In 2016, Wu *et al.* incorporated new properties of residues into PseAAC to enhance the previous models by using F-score as the feature selection strategy and SVM as the supervised classification algorithm. With an overall accuracy of 94.6% based on jackknife cross-validation results, the proposed model of Wu *et al.* (2016) was able to outperform the previous models of Yuan *et al.* (2013) and Ding *et al.* (2014). To deal with the imbalanced number of target types in the dataset, as well as to improve the discriminative power of the model, Zhang *et al.* (2016) proposed a random forest-based model called ICTCPred that uses SMOTE technique and hybrid features. The optimal features of the model were obtained by combining the relief strategy with the incremental feature selection (IFS) method. Based on jackknife cross-validation results, the overall accuracy of ICTCPred is 91%. In 2017, Wang *et al.* developed a model combining the analysis of variance and correlation (AVC) with SVM. By using dipeptide composition as features and five-fold cross-validation, the proposed model of Wang *et al.* (2017) was able to reach an overall accuracy of 92.0%.

Despite showing promising results, one major limitation in previous methods is the limited number of target types that they included in their models (Mansbach *et al.* 2019). Previous methods are limited to predicting conopeptide sequences that target three types of voltage-gated ion channels (*e.g.*, sodium, calcium, and potassium channels). Thus, to address this issue we initiated the development of a new dataset that contains an updated set of conopeptide sequences with additional target types. In this study, we also proposed a complementary method to sequence similarity search in predicting the potential targets of conopeptides given only their primary structures. The general idea of our proposed method is to represent the sequences as numerical features and to use PCA and multiclass logistic regression to learn a function that can map these features to their corresponding target types. Currently, the model is trained on making predictions for five target types of conopeptides that include both voltage-gated and ligand-gated ion channels. The proposed model will be useful in characterizing newly sequenced and isolated conopeptides. It can facilitate the development of new protocols for prioritizing the sequences that will be further subjected to chemical synthesis and biological assays.

MATERIALS AND METHODS

Dataset

Mature conopeptide sequences with experimentally validated molecular targets were retrieved from the ConoServer database in Jan 2019 (Kaas *et al.* 2007, 2011). To create a high-quality dataset, synthetic mutants and other conopeptide-like sequences which are primarily enzymatic and hormonal (*e.g.*, conohyal, conolysin, and con-ins) were excluded from the final dataset. In addition, for purposes of cross-validation, only conopeptide sequences which belong to target types with at least eight known members were retained in the final dataset. The final dataset contains 156 mature conopeptide sequences that have affinities to five different targets: nicotinic acetylcholine receptors (nAChR), sodium channels, calcium channels, N-methyl-D-aspartate receptor (NMDAR), and potassium channels (Table 1).

Table 1. Number of sequences per target type in the final dataset.

Target type	Number of sequences
Nicotinic acetylcholine receptors	68
Sodium channels	38
Calcium channels	29
N-methyl-D-aspartate receptors	13
Potassium channels	8

Sequence Representation

Due to limitations of the tool used in calculating descriptors on sequences with non-standard residues, a number of preprocessing steps was first performed prior to descriptor calculation. For sequences that contain non-standard residues, the non-standard residues were replaced by their parent amino acids. In cases where no parent amino acids were available, these residues were either deleted or replaced by a glycine residue. After removing the non-standard residues, each conopeptide sequence was then represented by a vector of numerical features derived from calculating hybrid descriptors (*i.e.*, multiple groups of descriptors). In this study, a total of 9207 descriptors were calculated using the protr package in R-3.5.3 (Xiao *et al.* 2015). These include descriptors such as composition-based descriptors, autocorrelation descriptors, composition/transition/distribution (CTD) descriptors, conjoint triad descriptors, quasi-sequence-order descriptors, and pseudo-amino acid composition (Table 2). The detailed description of each descriptor is presented in the protr documentation.

Feature Scaling

Since the dataset contains features with a varying range of values, the dataset was first subjected to feature scaling. Many machine learning algorithms require the features to be on the same scale because most of them use distance or similarity calculations. Feature scaling will allow the features to have contributions that are proportional to the calculated distances. In this study, the data was standardized using the following equation:

Table 2. List of descriptors calculated for each sequence.

Descriptor groups	Descriptors	Number of features
Amino acid composition	Amino acid composition Dipeptide composition Tripeptide composition	20 400 8000
Autocorrelation	Normalized Moreau-Broto Geary Moran	56 56 56
CTD	Composition Transition Distribution	21 21 105
Conjoint triad	Conjoint triad	343
Quasi-sequence-order	Sequence-order-coupling number Quasi-sequence-order descriptors	14 54
Pseudo amino acid composition	Type I Type II	27 34

$$x' = \frac{x - \mu}{\sigma} \qquad (1)$$

where x is the original feature vector, μ is the mean of the feature vector, σ is its standard deviation, and x' is the scaled feature vector.

Principal Component Analysis

To reduce the dimensionality of the data, PCA was performed before feeding the data to a multiclass logistic regression algorithm. PCA is a widely used dimensionality reduction technique in fields such as data mining, bioinformatics, and machine learning among others (Tharwat 2016). PCA reduces the number of dimensions of the data by projecting the data to a lower-dimensional space wherein the variance is maximum. By maximizing the variance, PCA is still able to preserve the global structure of the data even after the transformation. As a result of the transformation, PCA is also able to extract new features which are linear combinations of the original features. Since this new set of features are orthogonal and uncorrelated to each other, PCA is able to effectively eliminate the redundant features from the original feature space.

In this study, PCA was implemented using the scikit-learn library in Python 3.6 (Pedregosa *et al.* 2011). In particular, singular value decomposition (SVD) was used to project the data into a lower-dimensional space and to rank the most relevant principal components that will comprise the new feature space (Tharwat

2016). SVD is a matrix decomposition method that allows any n x m matrix A to be expressed as a product of three matrices:

$$A_{nxm} = U_{nxn} \times \Sigma_{nxm} \times V^{T}_{mxm} \tag{2}$$

where U is an n x n matrix where the columns are called left-singular vectors, Σ is a diagonal n x m matrix with non-negative diagonals called singular values that are sorted in descending order, and V is an m x m matrix where the columns are called right-singular vectors. When SVD is used in PCA, the right singular vectors of A also correspond to the eigenvectors of the covariance matrix representing the principal axes of the data. The projection of the data on these axes are called principal components, wherein a dimension is basically a linear combination of the input features. On the other hand, singular values are related to the eigenvalues i of the covariance matrix via the equation:

$$\lambda_i = \frac{\sigma_i^2}{N-1} \tag{3}$$

where i is the singular value in the i-th row of the diagonal matrix, and N is the number of samples. Hence, eigenvectors can be sorted based on the eigenvalues i to provide the ranking of the principal components as they represent the variances of the respective principal components. The number of dimensions of the new feature subspace depends on the chosen K principal components. For this study, the top principal components that explained 99% of the variance were selected as input to the multiclass logistic regression algorithm.

Multiclass Logistic Regression

Multiclass logistic regression is a classification algorithm that generalizes the logistic regression to multiclass classification problems. To construct a multiclass classification with logistic regression, a one-*vs.*-rest scheme (OvR) implementation of scikit-learn was used in this study (Pedregosa *et al.* 2011). In OvR, a binary model is trained for each class. It uses a logistic function to map the range of output values of – to + to a smaller range of 0 to 1. The logistic function is given by the following equation:

$$\hat{p} = \frac{1}{1+e^{-z}} \tag{4}$$

where z is a linear combination of the input features. By fitting a logistic function to the dataset, logistic regression as a supervised machine learning algorithm is able to naturally assign the class probabilities for each data point. Thus, in order to make a prediction in multiclass logistic regression, the class with the highest probability is determined. This class probability can also be interpreted as the distance of a point from the hyperplane that separates the classes. In other words, the farther the points from the separating hyperplane, the higher the probability that it belongs to that certain class. To determine the optimal parameters of this hyperplane, a gradient descent algorithm was used in this study.

Nested Cross-validation

An important element in training any machine learning model is generalizability. Generalizable models have the ability to make accurate predictions for unseen data points. In general, to avoid overfitting, the model should be low on both bias and variance. Thus, to evaluate the ability of the models to generalize given only the available data, a nested cross-validation procedure was applied in this study. As demonstrated by Varma and Simon (2006), a nested cross-validation procedure provides an unbiased estimate of the true error since it gives an estimate of the error that is highly comparable to the error obtained from the independent testing set. In nested cross-validation, an inner cross-validation loop is used to tune the hyperparameters and select the best model, while an outer cross-validation loop is used to evaluate the generalization performance of the model.

For this study, a grid-search strategy with stratified five-fold cross-validation was applied in the inner cross-validation loop, while a leave-one-out cross-validation (LOOCV) was used in the outer cross-validation loop. A stratified five-fold cross-validation was specifically used in the inner cross-validation loop since the dataset is highly imbalanced. In stratified five-fold cross-validation, the percentage of each target type in the whole dataset is preserved in each training fold. On the other hand, a LOOCV strategy was used since the dataset has a small sample size. LOOCV is an extreme case of K-fold cross-validation wherein K is equal to the number of data points in the dataset. In LOOCV, each data point is iteratively left out from the training data and used for testing purposes.

The performance of the model was evaluated using accuracy, precision, recall, and F1-score. The equations of these performance metrics are formulated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 - score = 2 \times \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

where TP represents the true positives, TN represents the true negatives, FP represents false positives, and FN represents false negatives.

RESULTS AND DISCUSSION

Before fitting a logistic regression to the data, PCA was applied to visualize and reduce the dimensionality of the dataset. For purposes of visualization, the top two principal components from PCA that contain the most variance was used (Figure 1). Despite contributing to only 6% of the total variation, these two principal components already showed clustering of sequences according to their molecular targets. A distinct cluster was also formed for conopeptide sequences with high affinities to NMDAR. Conantokins, the only known conopeptides that target NMDAR, are known to have distinct structural features (*i.e.*, presence of Gla residues). On the other hand, potassium channel-targeting conopeptides did not show any form of clustering and appeared to be scattered on the PCA plot. The lack of clustering of this target type can be attributed to the small sample size in the dataset.

To reduce the dimensionality of the data, the top principal components from PCA that explained 99% of the total variance was used. This cutoff value was chosen to preserve the most relevant information from the original dataset. After performing PCA, the number of features was reduced from 9207 to 137 variables (Figure 2). This result indicates that the original dataset may contain many redundant and correlated features, which is expected because of the inherent similarities of the features used (*e.g.*, dipeptide composition and tripeptide composition).

After performing PCA, the top principal components were used as features to represent each sequence. A multiclass logistic regression was implemented to construct a model that maps these features to their corresponding molecular targets. By using a nested cross-validation strategy, the performance of our proposed method in predicting the target types of conopeptides was also compared with the performance of other classical machine learning algorithms. The utility of PCA as a preprocessing step for those algorithms was also evaluated. In terms of accuracy (Table 3), our proposed method – which is based on PCA and logistic regression – was able to produce the best results. It achieved an accuracy of 89%, outperforming the more advanced SVM algorithm. In addition to logistic regression, a significant improvement in accuracy was

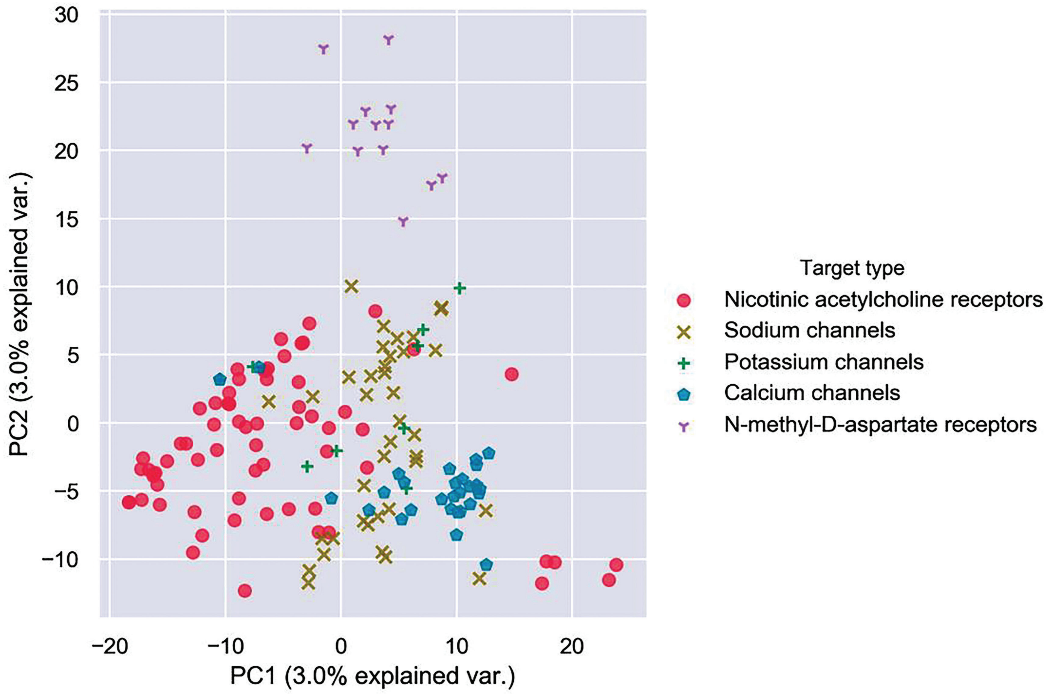

Figure 1. Visualization of the dataset using PCA. The axes correspond to the top two principal components with the highest explained variance. Each sequence is represented by a symbol and colored and marked according to its known target.

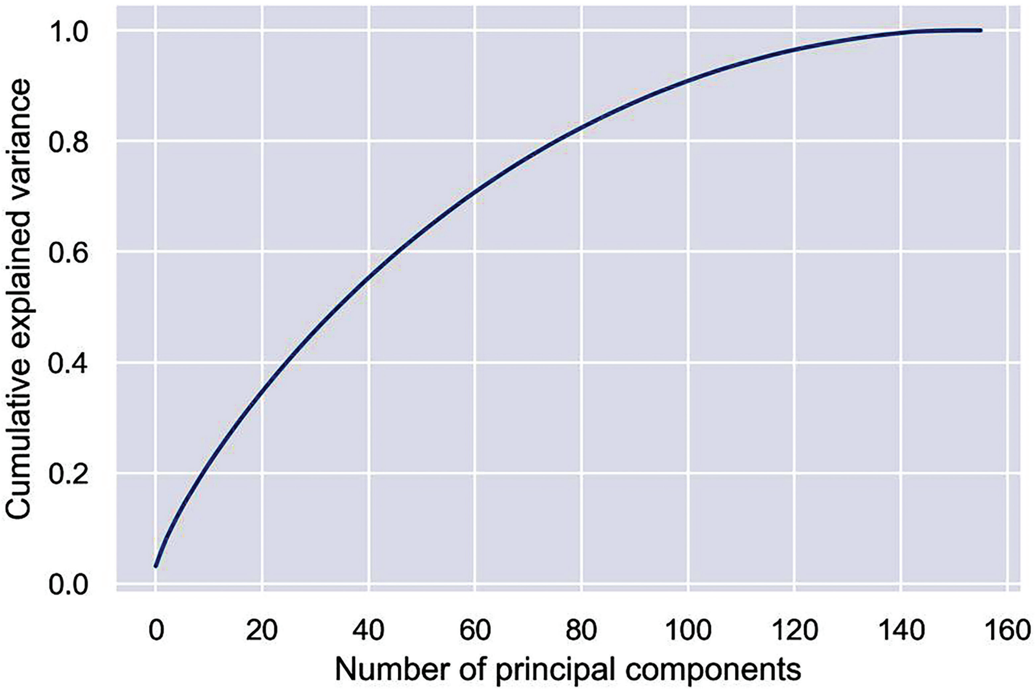

Figure 2. Cumulative variance explained by the principal components.

observed in the linear discriminant analysis (LDA) algorithm when PCA was applied. On the other hand, the accuracies of K-nearest neighbors (KNN) and naïve Bayes decreased when principal components were used as input features for these algorithms. SVM showed no difference in performance, but a significant reduction in training time was observed when PCA was applied prior to model training.

To further evaluate the performance of our proposed model, precision and F1-scores were also calculated (Table 4). In terms of precision, results showed that the combination of PCA and logistic regression still produced the best results. It was able to achieve a precision of 0.90. Together with logistic regression, LDA and KNN also increased in precision when PCA was performed before model training. Only naïve Bayes resulted in a decreased precision after applying PCA. In terms of F1-score, our proposed method still yielded optimal results. It was able to achieve an F1-score of 0.88. The F1-score of LDA also improved when PCA was used as a preprocessing step. Running PCA with KNN or naïve Bayes decreased the F1-scores of these algorithms. Results showed that PCA did not affect both the precision and F1-score of the SVM algorithm.

Table 3. Comparison of model performance based on accuracy.

Machine learning algorithms	Without PCA	With PCA
Linear discriminant analysis	0.47	0.85
Logistic regression	0.87	0.89
K-nearest neighbors	0.76	0.74
Naïve Bayes	0.75	0.67
Support vector machine	0.87	0.87

Table 4. Comparison of model performance based on precision and F1-score.

Machine learning algorithms	Precision		F1-score	
	Without PCA	With PCA	Without PCA	With PCA
Linear discriminant analysis	0.40	0.83	0.43	0.84
Logistic regression	0.87	0.90	0.86	0.88
K-nearest neighbors	0.76	0.78	0.74	0.73
Naïve Bayes	0.75	0.70	0.75	0.65
Support vector machine	0.82	0.82	0.84	0.84

Based on nested cross-validation and various performance metrics, the combination of PCA and multiclass logistic regression was able to outperform other classical machine learning algorithms in predicting the molecular targets of conopeptides. Thus, to fully evaluate the performance of our proposed model, we compared our model to a basic sequence similarity method. Sequence similarity calculations were performed using Clustal Omega (Sievers *et al.* 2011, Sievers and Higgins 2018). Predictions were made in the sequence similarity method by assigning the target type of the closest match (*i.e.*, target type of sequence with the highest similarity) to the query sequence. Results showed that our proposed method still have higher accuracy, precision, and F1-score than a basic sequence similarity search (Table 5).

The differences between the two methods were also examined based on their performance at the individual target types (Table 6). For purposes of comparing the performance of the two methods using a single criterion, the F1-score metric was highlighted in this study. Our proposed method showed better F1-score performance in predicting the conopeptide sequences that target calcium and sodium channels. On the other hand, the sequence similarity-based method outperformed our proposed method in predicting the target types of conopeptides that have high affinities to nAChR and potassium channels. Both methods were able to perfectly predict the conopeptides targeting the NMDAR. On the other hand, both methods showed poor performance in predicting potassium channel-targeting conopeptides, which is probably due to the lack of experimentally validated sequences of this target type in the database. Despite this limitation, in general, the results suggest that our proposed method is highly comparable to sequence similarity-based methods, and may be used as a complementary method for functional annotation of conopeptide sequences.

Table 5. Comparison of performance of the proposed method and sequence similarity-based method.

Method	Accuracy	Precision	F1-score
PCA and logistic regression	0.89	0.90	0.88
Sequence similarity	0.87	0.87	0.87

Aside from these general findings, our proposed method also provides practical advantages over the existing sequence similarity-based method. One practical advantage of our method is that it requires a smaller storage space than sequence similarity-based methods. Our method does not need all the training data to be stored in a database since a model has already been trained beforehand. To make predictions, our method only needs the learned parameters of the model and the calculated descriptors of the query sequence. Another practical advantage of our method is the ability of the logistic regression algorithm to naturally assign probabilities for each target type when making predictions. These probabilities are useful in designing experiments for newly sequenced conopeptides because these values can be used as a guide in prioritizing which receptor or ion-channels to test first.

CONCLUSION

In this study, we initiated the development of a high-quality dataset of conopeptide sequences with experimentally known molecular targets. By using the molecular descriptors calculated from the primary structures as features, we were able to develop a model that predicts five target types of conopeptides using PCA and multiclass logistic regression algorithms. Results showed that the combination of PCA and multiclass logistic regression was able to outperform other classical machine learning algorithms. Results also showed that our proposed method is capable of creating high-quality predictions for the different target types. In terms of overall performance, our proposed model can be considered as an improvement to a basic sequence similarity search. By examining the performance at the individual target types, our proposed method produced better results in predicting sequences that target calcium and potassium channels, while a basic sequence similarity method outperformed our method in predicting sequences targeting the nAChR and potassium channels. Our method also offers practical advantages in terms of requirement in storage space, and its ability to assign probabilities to its predictions. To the best of our knowledge, this is the first machine learning model that picks a conopeptide target type from an array of five potential target types. Overall, we were able to show that our proposed model can be used as a complementary

Table 6. Comparison of the performance per target type between the proposed method (Prop.) and sequence similarity-based method (Seq.).

Target type	Precision		Recall		F1-score	
	Prop.	Seq.	Prop.	Seq.	Prop.	Seq.
Nicotinic acetylcholine receptor	0.89	0.97	0.97	0.93	0.93	0.95
Sodium channel	0.87	0.76	0.87	0.82	0.87	0.78
Calcium channel	0.86	0.80	0.86	0.83	0.86	0.81
N-methyl-D-aspartate receptor	1.00	1.00	1.00	1.00	1.00	1.00
Potassium channel	1.00	0.57	0.25	0.50	0.40	0.53

method to sequence similarity search. The proposed model will be useful for high-throughput annotation of candidate receptors of newly sequenced and isolated conopeptides.

For future work, in addition to incorporating more data to improve the overall performance of the model, we would also like to explore semi-supervised learning techniques and ensemble-based methods to create a more robust model for predicting the potential targets of conopeptides. Also, a deeper analysis of the features will also be investigated to gain further insights and understanding of the data and the model. Finally, we also plan to deploy the model as a web application that can be easily and freely accessed by other researchers in the future.

ACKNOWLEDGMENTS

This study is part of Project 2 of Conus Structural Exogenomics Program, which is funded by the University of the Philippines (UP) Emerging Interdisciplinary Research Program and implemented by the Philippine Genome Center and UP Diliman Marine Science Institute, in collaboration with the School of Statistics and Department of Computer Science.

STATEMENT ON CONFLICT OF INTEREST

The authors declare no conflict of interest.

NOTES ON APPENDICES

The complete appendices section of the study is accessible at http://philjournsci.dost.gov.ph

REFERENCES

BARTON M, WHITE H, WILCOX K. 2004. The effect of cgx-1007 and ci-1041, novel NMDA receptor antagonists, on NMDA receptor-mediated EPSCS. Epilepsy Research 59(1): 13–24.

DEGUELDRE M, VERDENAUD M, SHEILA Z, GILLES N, DUCANCEL F, DE PAUW E, QUINTON L. 2017. Diversity in sequences, folds and pharmacological activities of toxins from four *Conus* species revealed by the combination of cutting-edge technologies of proteomics, transcriptomics, and bioinformatics. Toxicon 130: 116–125.

DING H, DENG E-Z, YUAN L-F, LIU L, LIN H, CHEN W, CHOU K-C. 2014. iCTX-Type: A sequence-based predictor for identifying the types of conotoxins in targeting ion channels. BioMed Research International, Vol. 2014, Article ID 286419, 10 pages.

GAO B, PENG C, YANG J, YI Y, ZHANG J, SHI Q. 2017. Cone snails: A big store of conotoxins for novel drug discovery. Toxins 9(12): 397.

HIMAYA S, LEWIS RJ. 2018. Venomics-accelerated cone snail venom peptide discovery. International Journal of Molecular Sciences 19(3): 788.

JIN A, VETTER I, HIMAYA S, ALEWOOD P, LEWIS R, DUTERTRE S. 2015. Transcriptome and proteome of *Conus planorbis* identify the nicotinic receptors as primary target for the defensive venom. Proteomics 15(23–24): 4030–4040.

KAAS Q, WESTERMANN J, HALAI R, WANG C, CRAIK D. 2007. Conoserver, a databse for conopeptide sequences and structures. Bioinformatics 24(3): 445–446.

KAAS Q, YU R, JIN A, DUTERTRE S, CRAIK D. 2011.Conoserver: Updated content, knowledge, and discovery tools in the conopeptide database. Nucleic Acids Research 40(D1): D325–D330.

LAVERGNE V, HARLIWONG I, JONES A, MILLER D, TAFT R, ALEWOOD P. 2015. Optimized deep-targeted proteotranscriptomic profiling reveals unexplored *Conus* toxin diversity and novel cysteine frameworks. Proceedings of the National Academy of Sciences 112(29): E3782–E3791.

LEWIS R, DUTERTRE S, VETTER I, CHRISTIE M. 2012. *Conus* venom peptide pharmacology. Pharmacological Reviews 64: 259–298.

LUBBERS N, CAMPBELL T, POLAKOWSKI J, BULAJ G, LAYER R, MOORE J, GROSS G, COX B. 2005. Postischemic administration of cgx-1051, a peptide from cone snail venom, reduces infarct size in both rat and dog models of myocardial ischemia and reperfusion. Journal of Cardiovascular Pharmacology 46(2): 141–146.

MANSBACH RA, TRAVERS T, McMAHON BH, FAIR JM, GNANAKARAN S. 2019. Snails *in silico*: A review of computational studies on the conopeptides. Marine Drugs 17(3): 145.

NIELSEN C, LEWIS R, ALEWOOD D, DRINKWATER R, PALANT E, PATTERSON M, YAKSH T, McCUMBER D, SMITH M. 2005. Anti-allodynic efficacy of the conopeptide, xen2174, in rates with neurpathic pain. Pain 118(102): 112–124.

OLDRATI V, ARRELL M, VIOLETTE A, PERRET F, SPRÜNGLI X, WOLFENDER JL, STÖCKLIN R. 2016. Advances in venomics. Molecular Biosystems 12(12): 3530–3543.

PEDREGOSA F, VAROQUAUX G, GRAMFOR A, MICHEL V, THIRION B, GRISEL O, BLONDEL M, PRETTENHOFER P, WEISS R, DUBOURG V, VANDERPLAS J, PASSOS A, CORNAPEAU D, BRUCHER M, PERROT M, DUCHESNAY E. 2011. Scikit-learn: Machine learning in python. Journal of Machine Learning Research 12: 2825–2830.

POPE J, DEER T. 2013. Ziconotide: A clinical update and pharmacologic review. Expert Opinion on Pharmacotherapy 14(7): 957–966.

PRASHANTH J, BRUST A, JIN A, ALEWOOD P, DUTERTRE S, LEWIS R. 2014. Cone snail venomics: From novel biology to novel therapeutics. Future Medicinal Chemistry 6(15): 1659–1675.

SAFAVI-HEMAMI H, HU H, GORASIA D, BANDYOPADHYAY P, VEITH P, YOUNG N, REYNOLDS E, YANDELL M, OLIVERA B, PURCELL A. 2014. Combined proteomic and transcriptomic interrogation of the venom gland of *Conus geographus* uncovers novel components and functional compartmentalization. Mol Cell Proteomics 13(4): 938–953.

SANDALL D, SATKUNANATHAN N, KEAYS D, POLIDANO M, LIPING X, PHAM V, DOWN J, KHALIL Z, LIVETT B, GAYLER K. 2003. A novel conotoxin identified by gene sequencing is active in suppressing the vascular response to selective stimulation of sensory nerves *in vivo*. Biochemistry 42(22): 6904–6911.

SIEVERS F, HIGGINS D. 2018. Clustal omega for making accurate alignments of many protein sciences. Protein Sci. 27: 135–145.

SIEVERS F, WILM A, DINEEN D, GIBSON T, KARPLUS K, LI W, LOPEZ R, McWILLIAM H, REMMERT M, SÖDING J, THOMPSON J, HIGGINS D. 2011. Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. Molecular Systems Biology 7(1): 539.

THARWAT A. 2016. Principal component analysis—A tutorial. International Journal of Applied Pattern Recognition 3(3): 197–240.

VARMA S, SIMON R. 2006. Bias in error estimation when using cross-validation for model selection. BMC Bioinformatics 7(1): 91.

VIOLETTE A, BIASS D, DUTERTRE S, KOUA D, PIQUEMAL D, PIERRAT F, STÖCKLIN R, FAVREAU P. 2012. Large-scale discovery of conopeptdes and conoproteins in the injectable venom of a fish-hunting cone snail using a combined proteomic and transcriptomic approach. Journal of Proteomics 75(17): 5215–5225.

WALSH I, POLLASTRI G, TOSATTO S. 2017. Correct machine learning on protein sequences: A peer-reviewing prespective. Briefings in Bioinformatics 5: 831–840.

WANG XF, WANG JM, WANG XL, ZHANG Y. 2017. Predicting the types of ion channel-targeted conotoxins based on avc-svm model. BioMed Research International, Vol. 2017, Article ID 2929807, 8 pages.

WU Y, ZHENG Y, TANG H. 2016. Identifying the types of ion channel-targeted conotoxins by incorporating new properties of residues into pseudo amino acid composition. BioMed Research International, Vol. 2016, Article ID 3981478, 5 pages.

XIAO N, CAO DS, ZHU MF, XU QS. 2015. Protr/protrweb: R package and web server for generating various numerical representation schemes of protein sequences. Bioinformatics 31(11): 1857–1859.

YUAN LF, DING C, GUO SH, DING H, CHEN W, LIN H. 2013. Prediction of the types of ion channel-targeted conotoxins based on radial basis function network. Toxicology In Vitro 27(2): 852–856.

ZHANG L, ZHANG C, GAO R, YANG R, SONG Q. 2016. Using the smote technique and hybrid features to predict the types of ion channel-targeted conotoxins. Journal of Theoretical Biology 403: 75–84.

APPENDIX

Table I. List of conopeptide sequences examined and their corresponding target.

ID	Sequence	Target
P00001	ICCNPACGPKYSCX	nAChRs
P00006	GCCSHPACNVNNPHICGX	nAChRs
P00010	GCCSDPRCAWRCX	nAChRs
P00015	GGCCSHPACAANNQDXCX	nAChRs
P00023	GRCCHPACGKNYSCX	nAChRs
P00025	YCCHPACGKNFDCX	nAChRs
P00026	GCCSTPPCAVLYCX	nAChRs
P00030	GCCSDORCRYRCX	nAChRs
P00032	GCCSHPACNVNNPHICX	nAChRs
P00038	ZSOGCCWNPACVKNRCX	nAChRs
P00039	GCCSNPVCHLEHSNLCX	nAChRs
P00050	RDOCCYHPTCNMSNPQICX	nAChRs
P00051	GCCSLPPCAANNPDXCX	nAChRs
P00074	ECCNPACGRHYSCX	nAChRs
P00090	GCCCNPACGPNYGCGTSCS	nAChRs
P00095	GCCSYPPCFATNPDCX	nAChRs
P00098	IRDXCCSNPACRVNNOHVC	nAChRs
P00099	GCCSLPPCALSNPDXCX	nAChRs
P00132	GCCSHPACAGNNQHICX	nAChRs
P00405	GCCSDPRCNMNNPDXCX	nAChRs
P00595	RDPCCSNPVCTVHNPQICX	nAChRs
P00840	GOSFCKADEKOCEYHADCCNCCLSGICAOSTNWILPGCSTSSFfKI	sodium
P01268	ACSGRGSRCOOQCCMGLRCGRGNPQKCIGAHXDV	sodium
P01269	GEXXYQKMLXNLRXAEVKKNAX	NMDARs
P01270	GCOwDPWCX	calcium
P01272	GCPwDPWCX	calcium
P01309	GDCPwKPWCX	potassium
P01338	GEXXLQXNQXLIRXKSNX	NMDARs
P01356	CRIONQKCFQHLDDCCSRKCNRFNKCVX	potassium
P01386	CKGKGAKCSRLMYDCCTGSCRSGKCX	calcium
P01397	CCGVONAACHOCVCKNTCX	nAChRs
P01449	GCCGSYONAACHOCSCKDROSYCGQX	nAChRs
P01484	CKGKGAPCRKTMYDCCSGSCGRRGKCX	calcium
P01540	CKAAGKPCSRIAYNCCTGSCRSGKCX	calcium
P01546	CKSOGSSCSOTSYNCCRSCNOYTKRCYX	calcium
P01552	WCKQSGEMCNLLDQNCCDGYCIVLVCT	sodium
P01553	WCKQSGEMCNVLDQNCCDGYCIVFVCT	sodium
P01554	ACRKKWEYCIVPIIGFIYCCPGLICGPFVCV	sodium
P01556	ACSKKWEYCIVPILGFVYCCPGLICGPFVCV	sodium
P01558	GRCCHPACGKYYSCX	nAChRs
P01559	CCKYGWTCWLGCSPCGC	sodium

P01560	CCKYGWTCLLGCSPCGC	sodium
P01561	DDCIKOYGFCSLPILKNGLCCSGACVGVCADLX	sodium
P01566	RDCCTOORKCKDRRCKOMKCCAX	sodium
P01571	RDCCTOOKKCKDRQCKOQRCCAX	sodium
P01574	KCNFDKCKGTGVYNCGXSCSCXGLHSCRCTYNIGSMKSGCACICTYY	nAChRs
P01594	GOOCCLYGSCROFOGCYNALCCRKX	nAChRs
P01598	VKPCRKEGQLCDPIFQNCCRGWNCVLFCV	sodium
P01603	CKQADEPCDVFSLDCCTGICLGVCMW	calcium
P01611	HOOCCLYGKCRRYOGCSSASCCQRX	nAChRs
P01615	ZNCCNGGCSSKWCKGHARCCX	sodium
P01616	RHGCCKGOKGCSSRECROQHCCX	sodium
P01630	CKQAGESCDIFSQNCCVGTCAFICIEX	sodium
P01632	CKGKGAOCTRLXYDCCHGSCSSSKGRCX	calcium
P01635	GCCGPYONAACHOCGCKVGROOYCDROSGGX	nAChRs
P01638	CKGKGASCHRTSYDCCTGSCNRGKCX	calcium
P01662	ZRCCNGRRGCSSRWCRDHSRCC	sodium
P01684	DLRQCTRNAPGSTWGRCCLNPMCGNFCCPRSGCTCAYNWRRGIYCSC	nAChRs
P01685	DDXSXCIINTRDSPWGRCCRTRMCGSMCCPRNGCTCVYHWRRGHGCSCPG	nAChRs
P01686	DVQDCQVSTOGSKWGRCCLNRVCGPMCCPASHCYCVYHRGRGHGCSC	nAChRs
P01688	CCGVONAACPOCVCNKTCGX	nAChRs
P01689	CCGVONAACHOCVCTGKC	nAChRs
P01690	ZGCCGEPNLCFTRWCRNNARCCRQQ	sodium
P01691	ZGCCNVPNGCSGRWCRDHAQCCX	sodium
P01692	GRCCEGPNGCSSRWCKDHARCCX	sodium
P01693	GRCCDVPNACSGRWCRDHAQCCX	sodium
P01725	CKGKGQSCSKLMYDCCTGSCSRRGKCX	calcium
P01726	CKGKGASCRKTMYDCCRGSCRSGRCX	calcium
P01727	CLSOGSSCSOTSYNCCRSCNOYSRKC	calcium
P01728	CKPOGSOCRVSSYNCCSSCKSYNKKCG	calcium
P01740	GCCGKYONAACHOCGCTVGROOYCDROSGGX	nAChRs
P01741	DDDCEPPGNFCGMIKIGPPCCSGWCFFACA	calcium
P01742	GCLEVDYFCGIPFANNGLCCSGNCVFVCTPQ	calcium
P01743	CKSOGTOCSRGMRDCCTSCLLYSNKCRRY	calcium
P01744	RDCCTOOKKCKDRRCKOLKCCAX	sodium
P01793	CKLKGQSCRKTSYDCCSGSCGRSGKCX	calcium
P01794	CRSSGSOCGVTSICCGRCYRGKCTX	calcium
P02228	ZRLCCGFOKSCRSRQCKOHRCCX	sodium
P02485	NGRCCHPACGKHFSCX	nAChRs
P02491	GCCSHPACSVNNPDICX	nAChRs
P02496	GCCARAACAGIHQELCX	nAChRs
P02497	GCCSHPACSGNHQELCDX	nAChRs
P02501	CCSHPACAANNQDXCX	nAChRs
P02511	QCCANPPCKHVNCX	nAChRs
P02526	CRAXGTYCXNDSQCCLNXCCWGGCGHOCRHPX	potassium

P02539	GCCSDORCNYDHPXICX	nAChRs
P02548	CKSTGASCRRTSYDCCTGSCRSGRCX	calcium
P02549	CKSKGAKCSKLMYDCCSGSCSGTVGRCX	calcium
P02570	NXSXCPwHPWCX	calcium
P02581	CKSOGTOCSRGMRDCCTSCLSYSNKCRRY	calcium
P02585	GCCSDPRCRYRCR	nAChRs
P02586	ACCSDRRCRWRCX	nAChRs
P02587	CCNCSSKWCRDHSRCCX	sodium
P02588	MCPPLCKPSCTNCX	nAChRs
P02591	LOSCCSLNLRLCOVOACKRNOCCTX	potassium
P02594	GEXXVAKMAAXLARXNIAKGCKVNCYP	NMDARs
P02595	EACYAOGTFCGIKOGLCCSEFCLPGVCFGX	sodium
P02608	GCCSHPACSVNHPELCX	nAChRs
P02625	DXCCXOQWCDGACDCCS	sodium
P02635	GCCSHPACSVNNPDICX	nAChRs
P02637	GEXXVAKMAAXLARXDAVNX	NMDARs
P02665	GCCARAACAGIHQELCX	nAChRs
P02675	CQGRGASCRKTMYNCCSGSCNRGRCX	calcium
P02707	ZNCCNGGCSSKWCRDHARCCX	sodium
P02710	DGCSSGGTFCGIHOGLCCSEFCFLWCITFID	sodium
P02748	IRDECCSNPACRVNNPHVCRRR	nAChRs
P02752	DECCSNPACRVNNPHVCRRR	nAChRs
P02832	AARCCTYHGSCLKEKCRRKYCCX	nAChRs
P02871	WPCKVAGSPCGLVSECCGTCNVLRNRCV	sodium
P02881	GCCSROOCIANNPDLCX	nAChRs
P03272	SRCFPPGIYCTPYLPCCWGICCGTCRNVCHLRI	potassium
P03322	DCCPAKLLCCNP	sodium
P03340	CKGKGASCRRTSYDCCTGSCRLGRCX	calcium
P03532	GEXXHSKYQXCLRXIRVNKVQQXC	NMDARs
P03535	GEPXVAKWAXGLRXKAASNX	NMDARs
P03536	DEPXYAXAIRXYQLKYGKI	NMDARs
P03537	GEDXYAXGIRXYQLIHGKI	NMDARs
P03584	RCCTGKKGSCSGRACKNLKCCAX	sodium
P03585	ZKCCTGKKGSCSGRACKNLRCCAX	sodium
P03623	VTDRCCKGKRECGRWCRDHSRCCX	sodium
P03625	VGERCCKNGKRGCGRWCRDHSRCCX	sodium
P03641	GDXXYSKFIXRERXAGRLDLSKFP	NMDARs
P03787	VLEKDCPPHPVPGMHKCVCLKTC	nAChRs
P03833	NGRCCHPACGKHFNCX	nAChRs
P03901	RTCCSROTCRMEYPXLCGX	nAChRs
P03906	LOOCCTOOKKHCOAOACKYKOCCKS	potassium
P04069	ZTOGCCWNPACVKNRCX	nAChRs
P04200	GCCSNPACMVNNPQIC	nAChRs
P04228	GGCCSHPACQNNPDXCX	nAChRs

P04241	CKGTGKSCSRIAYNCCTGSCRSGKCX	calcium
P04243	CKGKGASCRRTSYDCCTGSCRSGRCX	calcium
P05237	GTYLYPFSYYRLWRYFTRFLHKQPYYYVHI	potassium
P05239	ARFLHPFQYYTLYRYLTRFLHRYPIYYIRY	potassium
P05345	GCCSDPPCRNKHPDLCX	nAChRs
P05355	CKGQSCSSCSTKEFCLSKGSRLMYDCCTGSCCGVKTAGVT	calcium
P05526	VRCLEKSGAQPNKLFRPPCCQKGPSFARHSRCVYYTQSRE	nAChRs
P05840	RDCQEKWEYCIVPILGFVYCCPGLICGPFVCV	sodium
P05841	GEXXLAXKAOXFARXLANX	NMDARs
P05842	ADXXYLKFIXEQRKQGKLDPTKFP	NMDARs
P05851	GEXXLSXNAVXFARXLANX	NMDARs
P05865	CCHPACGKNYSCX	nAChRs
P05941	TWEECCKNPGCRNNHVDRCRGQV	nAChRs
P05949	RGCCSHPACNVDHPEICX	nAChRs
P06003	PECCTHPACHVSNPELCX	nAChRs
P06679	SCCARNPACRHNHPCV	nAChRs
P06732	PCQSVRPGRVWGKCCLTRLCSTMCCARADCTCVYHTWRGHGCSCVM	nAChRs
P06773	CAGIGSFCGLPGLVDCCSGRCFIVCLP	sodium
P06776	SGCCSNPACRVNNPNICX	nAChRs
P06805	CAGIGSFCGLPGLVDCCSDRCFIVCLP	sodium
P06814	KPCCSIHDNSCCGL	nAChRs
P06822	GXCGDOGATCGKLRLYCCSGFCDXYTKTCKDKSSA	sodium
P06831	GCCSHPVCSAMSPICX	nAChRs
P06849	CAAFGSFCGLPGLVDCCSGRCFIVCLL	sodium
P07401	GEXXYSXAIX	NMDARs
P07646	CKPOGSKCSOSMRDCCTTCISYTKRCRKYY	calcium
P07647	CKPOGSKCSOSMRDCCTTCISYTKRCRKYYN	calcium