Predicting function from sequence in venom peptide families

Arvind Kannan
Email: afk@stanford.edu

G. Seshadri
Email: seshg@stanford.edu

***Abstract*—Toxins from animal venoms are small peptides that recognize specific molecular targets in the brains of prey or predators. Next generation sequencing has uncovered thousands of diverse toxin sequences, but the functions of these peptides are poorly understood. Here we demonstrate that the use of machine learning techniques on sequence-derived features enables high accuracy in the task of predicting a toxin's functionality using only its amino acid sequence. Comparison of the performance of several learning algorithms in this prediction task demonstrates that both physiochemical properties of the amino acid residues in a sequence as well as noncontiguous sequence motifs can be used independently to model the sequence dependence of venom function. We rationalize the observed model performance using unsupervised learning and make broad predictions about the distribution of toxin functions in the venome.**

***Keywords*—Bioinformatics, machine learning, protein function prediction, venomics.**

I. Introduction

Over the past fifty years, proteins and small molecules isolated from animals, plants, and microorganisms have formed the basis of nearly all clinically approved pharmaceuticals. For example, many chemotherapeutics for cancer are derived from plant natural products [1], and mining microbial communities for biological warfare agents continues to be the most common avenue for discovering new antibiotics [2]. As such, the development of both computational and experimental tools that will enable rapid screening of the vast chemical space of molecules present in nature for important biological functions is of profound importance to future drug discovery efforts [3].

In the past decade, much attention has been focused on the therapeutic use of proteins extracted from the venom of snakes, tarantulas, cone snails, and other predatory animals [4]. Venom proteins are particularly attractive therapeutic candidates because they (1) typically exist as small peptides that are easy to synthesize, (2) possess remarkable thermal and chemical stability, (3) are highly tolerant to mutation, and (4) often contain flexible loops that can be reengineered for high affinity and specificity towards relevant biological targets. For example, a neurotoxin first isolated from cone snail venom has been developed into an FDA-approved painkiller [5], and knottin peptides from spider venom have been reengineered as tumor-targeting agents for cancer imaging and treatment [6]. Moreover, the rise of next-generation sequencing technologies over the past five years has given birth to the new field of venomics, in which mass spectrometry and RNA sequencing of venom extracts are coupled with bioinformatics analyses in order to identify and catalogue new venom components [7]. Indeed, the rate at which new sequencing data on such proteins is acquired is far outpacing our conceptual understanding of the structure and function of venom peptides, necessitating the development of new computational tools for effectively mining venomic data (Figure 1).

In this paper, we describe the use of both supervised and unsupervised machine learning techniques in order to identify sequence-derived features that predict the likely biological targets of venom components. We have made use of the wealth of publicly available databases of venom proteins, such as ConoServer [8] and the Knottin Database [9], which contain amino acid sequences as well as annotated structural and functional data for each protein when available. Previous work in this field has led to the development of predictors that use predominantly sequence homology in order to classify sequences into venom families [10]. This approach is fundamentally limiting because homology often provides very little information about the putative function of a protein, especially because the most evolutionarily conserved regions of a toxin sequence typically lie in precursor signal peptide regions, which are subsequently cleaved off in the mature toxin. Moreover, many emerging sources of venom sequence data, such as proteomics studies, provide information only about the mature toxin sequence in the absence of the signal peptide, necessitating the development of predictive tools that do not rely on signal peptide homology.

For this reason, we have developed classifiers of toxin function using both physicochemical properties of the amino acid residues as well as conserved sequence motifs within a peptide sequence. We demonstrate that these derived features segregate venom constituents into functional classes with high performance and low generalization error. We then rationalize the observed performance of our models using unsupervised clustering and feature reduction. Finally, by running our predictors on a larger set of uncharacterized toxin sequences

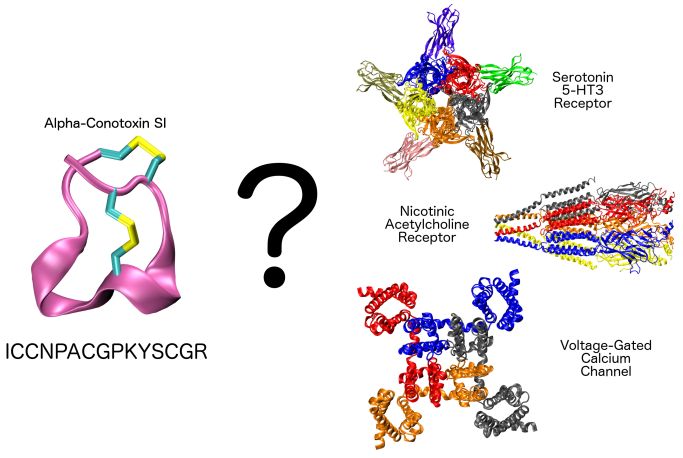

Fig. 1. The protein function prediction problem for venom peptides. On the left is shown a representative toxin from the fish-hunting cone snail *Conus striatus*. On the right are shown three canonical examples of transmembrane proteins in the brain whose function can be modulated via the binding of such toxins. The challenge of functional venomics is to link the amino acid sequence of a venom peptide with the identity of its molecular target.

and comparing the functional clusters that emerge to the known gene superfamilies based on signal peptide homology, we provide insights into the evolution and diversification of toxin function.

II. Methods

A. Training Data

Labeled training data for toxin classification were obtained from two publicly accessible databases, ConoServer and ArachnoServer, which contain functionally annotated peptide sequences from cone snail and spider venom, respectively. In both servers, sequences are binned into functional categories based on published experimental data, although the classes differ in the two cases. Spider toxins can target highly heterogenous protein targets, so ArachnoServer classifies each sequence according to the broad functional class of its target (i.e. membrane proteins, ion channels, enzymes, etc.). In contrast, conotoxins function almost exclusively as neurotoxins, so ConoServer groups these peptides according to their neuromodulatory function on specific subtypes of ion channels and neurotransmitter receptors. In both cases, the distribution of class members is highly skewed, with some functional categories containing significantly more annotated examples than others (Table 1). Since greater than 90% of the annotated examples were encapsulated in the top 3 or top 4 classes for the spider and cone snail datasets, respectively, we consolidated all remaining classes into an "other" category to facilitate learning and prediction given the low number of available examples.

The available functionally annotated sequences used for training the models described in this work constitute less than 10% of the total database size for both organisms, highlighting a growing disparity between the throughput of next generation sequencing and that of functional screens, and emphasizing the need for accurate bioinformatic tools to bridge this gap. Moreover, the low to moderate pairwise sequence similarity within each functional class (Table 1) precludes accurate classification using naive sequence alignments and justifies the use of physically motivated supervised models.

Training examples were pre-processed by identifying and excising any signal or pro-peptide regions from the toxin sequence using the SpiderP and ConoPrec utilities available within ArachnoServer and ConoServer, respectively. These algorithms utilize SVM-based models in conjunction with sequence heuristics to identify likely protease cleavage sites that demarcate the pro-sequence from the mature toxin. This pre-processing step increases the chance that our models learn real biophysical determinants of venom function as opposed to artifacts associated with signal peptide evolution.

B. Feature Representations

Two complementary feature representations of the toxin sequences were designed and compared with respect to their performance in predicting venom function. In the first strategy, we represented each toxin as a six-dimensional vector that summarizes the physicochemical properties of the peptide sequence (Figure 2). In particular, we incorporated the overall sequence length as well as the number of residues in each electrostatic class (hydrophobic, polar, and neutral) as relevant features that constrain the types of binding pockets and receptor motifs which can be accomodated by the peptide. Additionally, we paid special attention to the number and position of cysteine amino acids within each sequence, as these residues form disulfide bonds that significantly constrain the fold and stability of each peptide.

TABLE I. Summary of labeled data used to train functional classifiers for spider and cone snail toxins.

Spider toxins (ArachnoServer)			Cone snail toxins (ConoServer)		
Functional class	**Number of sequences**	**Pairwise sequence similarity**	**Functional class**	**Number of sequences**	**Pairwise sequence similarity**
Membrane	47	25 ± 22 %	Alpha	143	44 ± 20 %
Channel	203	21 ± 13 %	Delta	19	43 ± 21 %
Enzyme	53	49 ± 27 %	Mu	28	35 ± 20 %
Other	18	26 ± 22 %	Omega	51	45 ± 26 %
			Other	26	35 ± 23 %

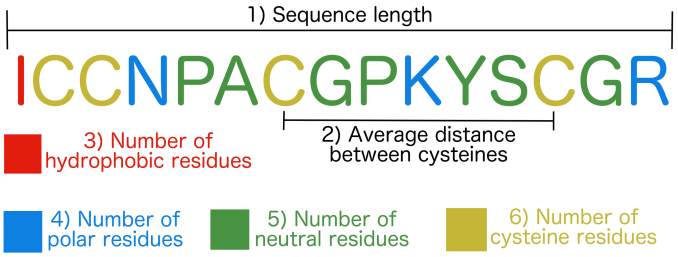

Fig. 2. Schematic of feature representation used for venom classification using physicochemical characteristics of the peptide sequence.

In parallel, we also evaluated an alternative text-based representation of the toxin sequences using noncontiguous substrings of length n. Letting Γ be the set of all amino acid sequences of length n, we defined the feature representation of a string s as

$$\phi(s) \in \mathbb{R}^{|\Gamma|}, \tag{1}$$

with

$$\phi_u(s) = \sum_{\substack{\mathbf{i}=(i_i,\ldots,i_n) \\ 1\le i_1\le\ldots\le i_n\le|s| \\ \Gamma^{(u)}=s[\mathbf{i}]}} \lambda^{i_n-i_1+1} \tag{2}$$

Here the summation is taken over all noncontiguous substrings of s that match with the uth word in the dictionary Γ, and the discount factor $0 < \lambda \le 1$ penalizes each match based on the extent of noncontiguity of the substring. This representation has the effect of capturing motifs of length n present within the amino acid sequence of each peptide, and is equivalent to the n-gram model in the limit as $\lambda \to 0$. For nonzero λ, the representation has the capacity to model correlations between amino acids that are distal along the primary sequence, and can thus be used to identify 3D structural motifs such as secondary and tertiary elements even in the absence of an explicit structural model.

C. Learning Algorithms

Learning using the six-dimensional physicochemical feature representation described above was carried out using both multinomial logistic regression,

$$\max_{\substack{\{\theta_1,\ldots,\theta_K\} \\ \theta_K=0}} \sum_{i=1}^{m}\left[\sum_{k=1}^{K}\mathbb{1}\{y^{(i)}=k\}\theta_k^T x^{(i)} - \log\sum_{k=1}^{K} e^{\theta_k^T x^{(i)}}\right] \tag{3}$$

as well as multiclass SVM using the one-vs-one strategy,

$$\max_{\substack{\alpha\in\mathbb{R}^m \\ 0\le\alpha_i\le C \\ \alpha^T\vec{y}=0}} \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y^{(i)}y^{(j)}\,K(x^{(i)},x^{(j)}), \tag{4}$$

where the optimization in Equation 4 is carried out for each pair of classes. The SVM algorithm was implemented using both a linear kernel,

$$K(x, y) = x^T y, \tag{5}$$

as well as a Gaussian kernel (RBF),

$$K(x, y) = \exp\left(-\gamma \left|\left|x - y\right|\right|^2\right) \tag{6}$$

The scikit-learn machine learning library was chosen for efficient implementation of the above algorithms, with LIBSVM used as the optimization engine.

Learning using the substring text representation was carried out using a multiclass SVM with the String Subsequence Kernel (SSK) [11],

$$K(s, t) = \sum_{u \in \Gamma} \sum_{\substack{\mathbf{i}=(i_i,\ldots,i_n) \\ 1 \le i_1 \le \ldots \le i_n \le |s| \\ u = s[\mathbf{i}]}} \sum_{\substack{\mathbf{j}=(j_i,\ldots,j_n) \\ 1 \le j_1 \le \ldots \le j_n \le |t| \\ u = t[\mathbf{j}]}} \lambda^{l(\mathbf{i})+l(\mathbf{j})}, \tag{7}$$

with Γ, n, and λ defined as above. Note that while direct enumeration of this sum is exponentially expensive as a function of n, the SSK can be efficiently computed in $\mathcal{O}(n \times |s| \times |t|)$ time using dynamic programming, thereby enabling its use in learning problems of moderate size. An implementation of this kernel in the SHOGUN machine learning library was used to build our functional classifiers.

Free parameters in each learning algorithm (such as the SVM regularization constant C, the bandwidth parameter γ in the RBF, and the parameters n and λ in the SSK) were chosen so as to maximize the average test set accuracy over 10 replicates of 2-fold cross-validation, and were optimized via grid search.

III. Results and Discussion

A. Performance of Learning Algorithms

We first compared the performance of the learning algorithms described above with respect to cross-validation accuracy on both the spider and cone snail datasets (Figure 3). All four algorithms performed significantly better than chance on both datasets (baseline accuracies of 63% and 54% for ArachnoServer and ConoServer, respectively), suggesting that both feature representations capture at least some of the correlations between toxin sequence and function. Moreover, the top performing models on each dataset exhibit impressive multi-class accuracies considering the small training set sizes and relatively simple models used for learning. Quite surprisingly, all four algorithms exhibited uniformly higher performance on the ArachnoServer data than on the ConoServer data, despite the latter featuring substantially higher sequence similarity within each class on average (Table 1), suggesting that spider toxins manifest a more direct link between sequence and function than conotoxins. This conclusion is consistent with the fact that test set accuracy increases dramatically as a function of model complexity (RBF and SSK models versus linear models) for the ConoServer data but remains relatively constant across all models for the ArachnoServer data.

Since overall test set accuracy can sometimes be a poor proxy for algorithm performance in multi-class learning problems, especially when class membership is skewed, we computed confusion matrices for the SSK model, which exhibited the highest accuracy on both datasets (Figure 4). Classification accuracy varied significantly from class to class, with the most populous class always exhibiting significantly lower error rates than the others. Interestingly, classification

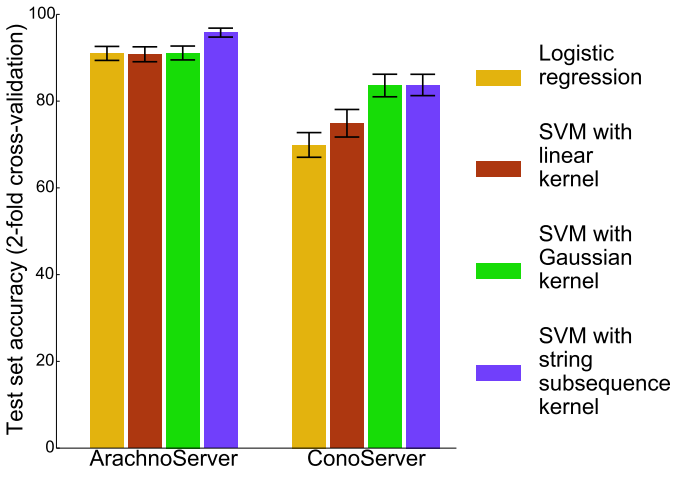

Fig. 3. Comparison of the performance of four learning algorithms with respect to multi-class prediction accuracy on the two toxin datasets considered in this work. Error bars reflect the standard deviation over 10 iterations of 2-fold cross-validation on each dataset. All algorithms except the string subsequence SVM were trained on the physicochemical feature representation shown in Figure 2.

accuracy did not correlate with class size outside of the largest class, and certain functional categories such as the δ-conotoxins consistently performed more poorly than others. This phenomenon was observed for all learning algorithms, and could reflect either bias in the training data (i.e. non-uniform sampling of the sequences in each class by experimentalists) or intrinsic heterogeneity of certain venom functions. In all datasets and models, the "other" class formed via concatenation of the functions with the fewest available examples was consistently misclassified into one of the more common classes. This result is unsurprising given that the sequences in the "other" class cover a diverse range of structures and functions, so any two sequences in this category likely share very little overlap in either physicochemical properties or primary sequence motifs. As such, we suspect that the performance of our algorithms could be further improved by either (a) excluding the "other" class entirely and training the model only on the most populous classes, or (b) explicitly breaking up the "other" class into its constituent functions in spite of low sub-class representation within the training set. Quite strikingly, all three classes except the "other" meta-class in the ArachnoServer dataset were predicted with near-perfect accuracy by all four learning

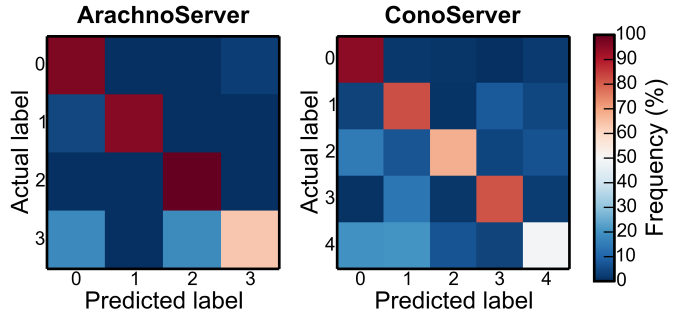

Fig. 4. Confusion matrices for the SSK model evaluated on both the ArachnoServer and ConoServer datasets. Reported frequencies are averages over 10 iterations of 2-fold cross-validation on each dataset. Classes are ranked according to the number of available training examples, and the last class corresponds to the "other" category in both cases.

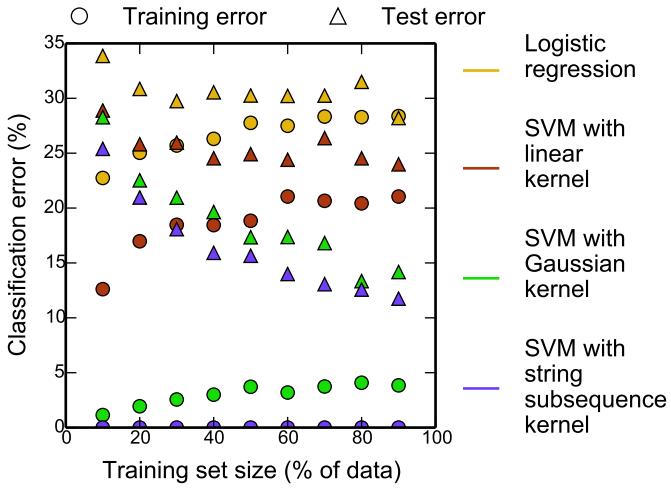

Fig. 5. Comparison of training and test error as a function of training set size for the 4 learning algorithms used in conotoxin classification.

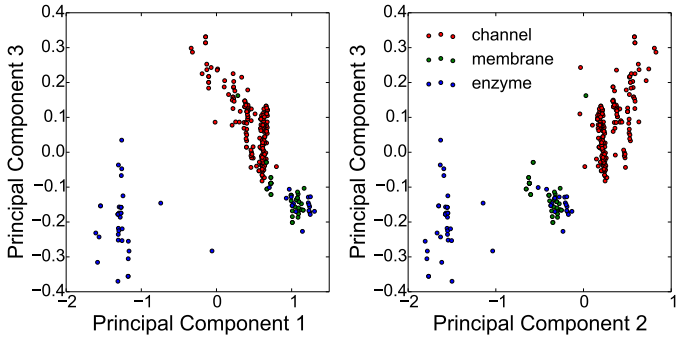

Fig. 6. Principal component analysis of the spider toxin dataset with respect to the 6 physicochemical features used for functional classification. The three most populous functional classes form differentiated clusters on low-dimensional subspaces of the feature space, and reasonable separation of the classes can be observed with as few as 3 principal components.

algorithms, providing further evidence for a robust mapping between sequence and function in this dataset.

B. Error Analysis

Next, in order to analyze the contributions of model bias and variance to the observed prediction errors, we plotted training and test set accuracy as a function of training set size for all 4 algorithms when evaluated on the ConoServer dataset (Figure 5). This analysis revealed that the primary source of generalization error in our learning algorithms shifts from bias to variance as the model complexity increases from the linear models (logistic regression and linear SVM) to the highly nonlinear models (RBF and SSK). For example, in the logistic classifier, the gap between training and test error shrinks to 0 as the training set size is increased, while the overall performance remains poor. This suggests a high bias regime, where the learned model is not sufficiently rich to benefit from additional data. In contrast, in the SSK and RBF classifiers, a significant gap between training and test error remains even when 90% of the data is used for training, although the overall performance of these methods is significantly higher than that of the linear models. This suggests a high variance regime, where the available training data is small relative to the model complexity and the likelihood of over-fitting is high. The string subsequence model demonstrates an extreme case of this phenomenon, where the training data is linearly separable (0% training error) in the high-dimensional space of noncontiguous sequence motifs, but the resultant classifier is nonetheless imperfect with respect to the test set.

The linear SVM presents a compromise between these two error regimes, providing reasonable performance (high enough accuracy to make statistical predictions on toxin libraries but too low for accurate single-sequence classification) while avoiding model variance. As such, we suspect that the linear SVM will be more robust to systematic sources of error in the training set, such as experimenter bias in the choice of sequences to characterize from each class, relative to the higher variance nonlinear models. For this reason, we chose to use the linear SVM when making predictions on the distribution of uncharacterized toxin functions in order to minimize false discovery at test time. An alternative strategy to reduce variance would be to increase the degree of regularization in the richer nonlinear algorithms (decrease C), which has the effect of sacrificing optimal accuracy in favor of model robustness.

C. Unsupervised Analyses

The extremely high performance and model insensitivity of the physicochemical feature representation on the ArachnoServer dataset was surprising given the low dimensionality (6) of the feature space relative to that of the sequences themselves ($20^{l(s)}$). In order to better understand the origins of the observed accuracy, we performed unsupervised clustering of the entire training set using principle component analysis (Figure 6). Briefly, feature vectors for each training sequence were constructed as in Figure 2, mean-substracted and normalized in order to weight all features equally, and then processed using the PCA implementation in the scikit-learn package. When functional labels were added to the transformed data, we found that the three most populous classes form differentiated clusters on low-dimensional subspaces of the feature space. Strikingly, reasonable separation between the classes was observed with as few as 3 principal components. These results corroborate the high performance of the supervised algorithms which use this feature set, and lead to the unexpected conclusion that a small number of independent physicochemical properties can explain a significant fraction of the functional variation among spider toxin sequences.

D. Feature Reduction

The PCA results motivated a forward search procedure to identify the minimal number of physicochemical features required for accurate prediction of spider toxin function. For each of the three learning algorithms based on this feature set, we started from a random model and sequentially added features to the predictor one-by-one in random order. We monitored the algorithm performance as a function of the number of features used at each iteration of this procedure (Figure 7), and found that the accuracy quickly converged to its maximal value for all three models using only a few features at a time. The test set accuracy when only four features were used was indistinguishable within error from the fully optimized models, in accordance with the PCA results.

E. Predictions on Uncharacterized Toxins

After analyzing and optimizing our prediction algorithms as described above, we classified all unannotated protein sequences within

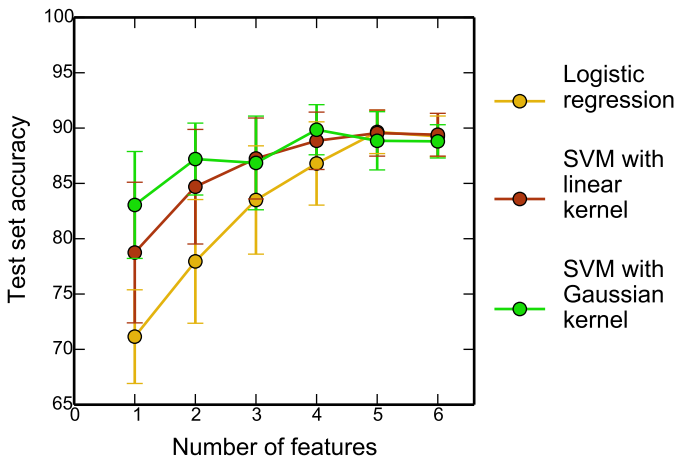

Fig. 7. Forward search procedure for feature reduction on the spider toxin dataset. Test set accuracy was evaluated using 2-fold cross-validation. Error bars reflect the standard deviation over 20 replicates of the forward search procedure, each with a different order of feature addition.

the ConoServer and ArachnoServer databases using the linear SVM (as motivated above). Table 2 summarizes the predicted fraction of the spider and cone snail venomes present within different functional classes. We observed that the predicted distribution of toxin sequences within each functional class differs significantly from the empirical distribution of the training set. For example, only 35% of toxins within the cone snail venome are predicted to be in the alpha functional class, while 54% of experimentally characterized sequences belong to this class. The discrepancy between the two distributions points to substantial experimental bias in the functional characterization of venoms, and may guide future high throughput functional screening efforts to improve sampling of the functional space.

F. Extension to Gene Superfamily Prediction

Finally, using our predictions on the entire conotoxin venome, we addressed the question of whether signal peptide homology is predictive of venom function, which motivated our initial development of a sequence-based functional classifier using only the mature toxin sequence. Predictions of putative venom function were carried out on all conotoxin sequences in the database that possessed a signal peptide, and the functional clusters that emerged were compared to gene superfamily assignments of each toxin based on signal peptide homology (Figure 8). Our models predicted a strong correlation between the functional role of a mature toxin and signal peptide that controls its processing and export through the ER after protein synthesis. This result was wholly unexpected and counterinuitive, given that the signal peptide is cleaved prior to venom secretion and thus plays no structural or functional role in the mature toxin. As such, the observed correlation implies a functionally decoupled co-evolution of these two regions of the peptide. Future research using similar machine learning techniques as those described here may provide valuable insights into the nature and mechanism of this co-evolution.

IV. Conclusions and Future Work

In sum, we have described the use of both supervised and unsupervised machine learning techniques to help elucidate the biological function of toxin peptides from two model organisms. Using rigorous cross-validation and error analysis, we showed that both

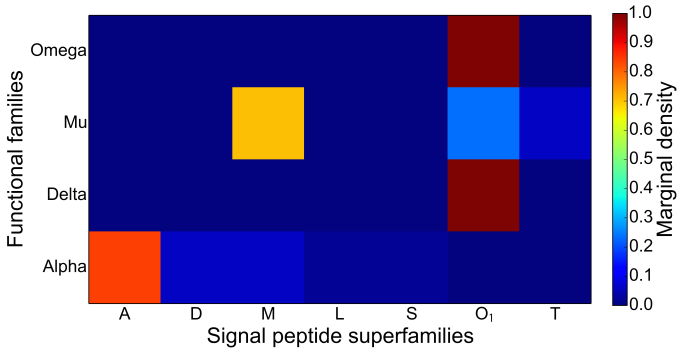

Fig. 8. Correlation between the functional role of a cone snail peptide, as predicted by the mature toxin sequence, and its gene superfamily, as assigned by the signal peptide sequence.

physicochemical summary statistics derived from the peptide sequence, as well as purely text-based models, can segregate functional toxin families with reasonable performance and low generalization error. Moreover, we suggest that application of the predictors to toxin sequences of unknown function may shed light on both the distribution and evolution of novel function within the venome.

References

[1] Da Rocha, Adriana B., Rafael M. Lopes, and Gilberto Schwartsmann. "Natural products in anticancer therapy." Current Opinion in Pharmacology 1.4 (2001): 364-369.

[2] Riesenfeld, Christian S., Patrick D. Schloss, and Jo Handelsman. "Metagenomics: genomic analysis of microbial communities." Annu. Rev. Genet. 38 (2004): 525-552.

[3] Drews, Jrgen. "Drug discovery: a historical perspective." Science 287.5460 (2000): 1960-1964.

[4] Saez, Natalie J., et al. "Spider-venom peptides as therapeutics." Toxins 2.12 (2010): 2851-2871.

[5] Stix, Gary. "A toxin against pain." Scientific American 292.4 (2005): 88-93.

[6] Kimura, Richard H., et al. "Engineered knottin peptides: a new class of agents for imaging integrin expression in living subjects." Cancer research 69.6 (2009): 2435-2442.

[7] Escoubas, Pierre, and Glenn F. King. "Venomics as a drug discovery platform." (2009): 221-224.

[8] Kaas, Quentin, et al. "ConoServer, a database for conopeptide sequences and structures." Bioinformatics 24.3 (2008): 445-446.

[9] Gelly, JeanChristophe, et al. "The KNOTTIN website and database: a new information system dedicated to the knottin scaffold." Nucleic acids research 32.suppl 1 (2004): D156-D159.

[10] Kaas, Quentin, Jan-Christoph Westermann, and David J. Craik. "Conopeptide characterization and classifications: an analysis using ConoServer." Toxicon 55.8 (2010): 1491-1509.

[11] Lodhi, Huma, et al. "Text classification using string kernels." The Journal of Machine Learning Research 2 (2002): 419-444.

TABLE II. Predicted distribution of toxin functions in uncharacterized test set.

Spider toxins (ArachnoServer)			**Cone snail toxins (ConoServer)**		
Functional class	**Training set fraction (observed)**	**Test set fraction (predicted)**	**Functional class**	**Training set fraction (observed)**	**Test set fraction (predicted)**
Membrane	0.64	0.975	Alpha	0.535	0.351
Channel	0.13	0.013	Delta	0.071	0.227
Enzyme	0.16	0.009	Mu	0.104	0.066
Other	0.06	0.002	Omega	0.191	0.21
			Other	0.097	0.143