# Conotoxin Protein Classification Using Pairwise Comparison and Amino Acid Composition

Nazar Zaki*
Intelligent Systems
Faculty of Info. Technology
UAEU, Al Ain 17551, UAE
nzaki@uaeu.ac.ae.

Fadi Sibai†
Computer System Design
Faculty of Info Technology
UAEU, Al Ain 17551, UAE
fadi.sibai@uaeu.ac.ae.

Piers Campbell‡
Enterprise Systems
Faculty of Info. Technology
UAEU, Al Ain 17551, UAE
p.campbell@uaeu.ac.ae

## ABSTRACT

Conotoxin classification could assist in the study of the structure function relationship of ion-channels and receptors as well as identifying potential therapeutics in the treatment of a wide variety of diseases such as schizophrenia, chronic pain, cardiovascular and bladder dysfunction. In this study, we introduce a novel method (Toxin-AAM) for conotoxin superfamily classification. Toxin-AAM incorporates evolutionary information using a powerful means of pairwise sequence comparison and amino acid composition knowledge. The combination of the sequential model and the discrete model has made the Toxin-AAM method exceptional in classifying conotoxin superfamily, when compared to other state-of-the-art techniques.

## Categories and Subject Descriptors

I.2 [**ARTIFICIAL INTELLIGENCE**]: [bioinformatics, Learning].

## General Terms

Algorithms, Performance, Reliability.

*Dr. Nazar Zaki is a Director of the Bioinformatics Laboratory and a Coordinator of Intelligent Systems with the Faculty of Information Technology, United Arab Emirates University, Al Ain, P. O. Box 17551, UAE,
Email: nzaki@uaeu.ac.ae
†Dr. Fadi Sibai directs the IBM Cell Center of Competency and the Computer Systems Design program, Faculty of Information Technology, United Arab Emirates University, Al Ain, P. O. Box 17551, UAE,
Email: fadi.sibai@uaeu.ac.ae
‡Dr. Piers Campbell is an Assistant Professor of Enterprise Systems in the Faculty of Information Technology at the United Arab Emirates University, Al Ain, P.O. Box 17551, UAE,
Email: p.campbell@uaeu.ac.ae.


*GECCO'11,* July 12–16, 2011, Dublin, Ireland.


## Keywords

Conotoxin classification, amino acid composition, pairwise alignment.

## 1. INTRODUCTION

Conotoxins are small neurotoxic peptides with disulphide connectivity that target ion-channels or G-protein coupled receptors. Conotoxins have a variety of actions, most of which have not yet been explored. Based on the number and pattern of disulphide bonds and the biological activities, conotoxins can be classified into eleven superfamilies: A, D, I1, I2, J, L, M, O, P, S, and T [1–6]. With the growing interest in conotoxins, accurate automated superfamily classification methods are needed to classify the increasing number of discovered sequences and structures. The earlier methods for protein homology detection and classification include FASTA [7], BLAST [8] and PSI-BLAST [9]. Most of the successful methods though typically rely on profile-sequence or profile-profile alignment such as PSI-BLAST [9, 10], COACH [11] and HHsearch [12], profile-profile alignment with SVM [13], profile-based direct kernels [14]. Other methods that utilize structural information are PROSPECT [15], distance-profile [16] and ProfNet [17]. However, these methods depend on multiple sequence alignments and are therefore computationally extensive. Moreover, despite the importance and extensive experimental investigations on conotoxins, the above mentioned methods have not been intensively tested to classify conotoxin families. This is probably due to the nature of the conotoxin sequence. Most of the conotoxin proteins are typically short (10-30 amino acids long) and therefore a profile information generated for any of the conotoxin family or superfamily could be rather limited.

Recently, researchers have turned their attention to classifying conotoxins using alignment-free approaches. Mondal et al. [6] have used several theoretical approaches for classifying conotoxin proteins into their respective superfamilies based on the primary sequence of the mature conotoxin. They incorporated the concept of pseudo-amino acid composition (PseAAC) [18] to represent the conotoxin protein sequence. This representation was further utilized in conjunction with several classifiers such as multi-class support vector machine (SVM), ISort (Intimate Sorting) predictor [19], least distance algorithms [20, 21], a multiple binary approach [22] - known as the one-versus-rest (1-v-r) SVMs.

Despite the success of the alignment-free methods discussed above, these methods have a major limitation. They

consider only the PseAAC to represent the conotoxin protein sequence and therefore the evolutionarily and structural relationships within the conotoxin superfamily were not incorporated. Since the superfamily is originally defined as a group of evolutionarily related proteins, members of a conotoxin superfamily could result from divergent evolution of homologues with significant similarity in the amino acid sequences. It is well established that homology can be inferred from sequence similarity, and, that homological relationships usually imply the same or at least very similar structural relationships.

One obvious way to gain evolutionarily information is through protein sequence similarity detection. Sequence similarity typically implies homology, which in turn may imply structural and functional similarity. For instance, sequences of several A-conotoxins have been determined with a high degree of homology evident. The discovery of a statistically significant similarity between two proteins is frequently used to justify inferring a common functional role for the two proteins. However, in many cases, proteins within the same conotoxin superfamily may not have significant similarity due to the hyper-variability of mature toxins, therefore a combination of sequential and non-sequential (discrete) models is highly desired.

In this study, we introduce a new feature extraction method for protein representation by incorporating evolution information using a powerful means of pairwise sequence comparison (sequential model)and amino acid composition knowledge (discrete model). The conventional Amino Acid Compositions (AAC) contain 20 components, each of which reflect the normalized occurrence frequency for one of the 20 native amino acids in a sequence. Owing to it's simplicity, the AAC model was widely used in many earlier statistical methods for predicting protein attributes [23, 24]. The extracted features are then used in conjunction with SVMs to discriminate between conotoxin superfamilies members.

# 2. METHODS

## 2.1 Datasets

The dataset used to evaluate our method was developed by Mondal et al. [6]. The protein sequences for conotoxins were collected from the Swiss-Prot release 47.1 [25]. Superfamilies with few sequences such as P-conotoxin and S-conotoxin were not included in the analysis. I-conotoxin superfamily was not included either as it was previously divided into two distinct gene superfamilies (I1-conotoxin and I2-conotoxin). The outcome of this process was a dataset that included 156 mature conotoxin sequences from A, M, O and T superfamilies. Data redundancy was removed using a greedy incremental algorithm as implemented in the CD-HIT program which was developed by Li et al. [26]. The final dataset consists of 116 entries from four superfamilies, i.e. A (25 entries), M (13 entries), O (61 entries) and T (17 entries) was constructed. A negative dataset that included 60 sequences that do not belong to any of the four mentioned superfamilies was formed from different eukaryotes with diverse functions. The CD-HIT program was used once again to screen the negative set which resulted in 60 sequences with identity less than 40%. The dataset is available at http://faculty.uaeu.ac.ae/nzaki/Toxin-AAM.htm. Once the benchmark dataset was constructed, the subsequent problem was how to find a precise prediction engine to represent the protein samples for training the engine and conducting the prediction?

## 2.2 Overview of the Toxin-AAM algorithm

Figure 1 illustrates the overview of the proposed method, which we call it conotoxin Amino Acid Matching (Toxin-AAM). The method consists of two major steps: (a) protein feature extraction/representation and (b) classification. In the subsequent sections we describe both steps.

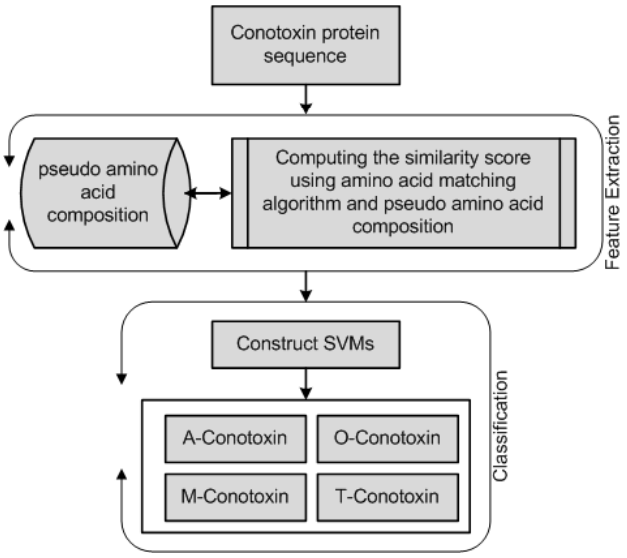


**Figure 1: Overview of the Toxin-AAM algorithm.**

## 2.3 Feature extraction

### *2.3.1 Amino acid matching algorithm*

In this step the protein features are generated by matching all the amino acid pairs available in the two protein sequences of interest. A pair of amino acids will have a non-zero entry if it occurs anywhere (not necessarily contiguous) in the amino acid sequence. The feature extraction algorithm is summarized as follows:

1. Let $\sum$ be a finite amino acid sequence.
2. For protein sequences $s$, $t$ we denote by $|s|$ and $|t|$ the lengths of the sequence $s = s_1...s_{|s|}$ and $t = t_1...t_{|t|}$ respectively. Clear the matching score: $S_{s,t} = 0$
3. The amino acid composition (AAC) of the protein $s$ is $s = [a_1, ..., a_{20}]$
4. For each of the two sequences $s$, $t$, we extract all the amino acid pairs $(s_i, s_j)$ and $(t_m, t_n)$ which could occur anywhere in $s$, $t$ (not necessarily contiguous) along with the corresponding distances $d_{s_{i,j}}$ and $d_{t_{m,n}}$ between each of the two amino acids.
5. For the sequences $s$, $t$, we match all the pairs $(s_i, s_j)$ with $(t_m, t_n)$ as depicted by steps (5) and (6.a.ii). We initially clear all the matching scores $S_{s,t} = 0$, for all $s$ and $t$, and set the weight decaying factor $\lambda$ $(0 < \lambda < 1)$ to 0.8, which has experimentally proven to improve the

accuracy. The weight decaying factor contributes to the matching score as seen in step (6.a.ii).

6. For all pairs $(s_i, s_j)$ and $(t_m, t_n)$ in sequences $s$ and $t$, respectively:

   (a) if the pairs $(s_i, s_j)$ and $(t_m, t_n)$ match, i.e. $s_i = t_m$ and $s_j = t_n$ , then

      i. add their corresponding distances $d_{s_{i,j}}$ and $d_{t_{m,n}}$, $d_{s,t} = d_{s_{i,j}} + d_{t_{m,n}}$, such that the distance $d = 1$ if the amino acids in the pair are contiguous, $d = 2$ if the amino acids in the pair are separated by a gap of 1 amino acid, $d = 3$ if the amino acids in the pair are separated by a gap of 2 amino acids, etc...

      ii. update the score $S_{s,t} = S_{s,t} + \lambda^{d_{i,j}} \times a_i \times a_j$, where $a_i$, $a_j$ are the corresponding composition values of the amino acids pairs $(s_i, s_j)$, respectively.

The above algorithm can be illustrated by the following example:

Consider the two amino acid sequences $s = lqlwa$, $t = lqal$. The corresponding normalized amino acid composition of the amino acids are ($a = 0.11056$), ($l = 0.22112$), ($q = 0.11056$) and ($w = 0.11056$). For each of the two sequences $s$ and $t$ we extract all the amino acid pairs ($lq$, $ll$, $ql$, $la$, $qa$ and $al$). The two sequences are implicitly transformed into feature vectors, where each feature vector is indexed by the one pair of amino acids. The seven dimensional features vectors are given in Table 1:

**Table 1: Matching the two sequences ($lqlwa$) and ($lqal$) to seven dimensional feature vectors.**

| Sequence | $lq$ | $ll$ | $lw$ | $la$ | $ql$ | $qw$ | $qa$ |
|---|---|---|---|---|---|---|---|
| $s$ ($lqlwa$) | $\lambda^1$ | $\lambda^2$ | $\lambda^3$ | $\lambda^4$ | $\lambda^1$ | $\lambda^2$ | $\lambda^3$ |
| $t$ ($lqal$) | $\lambda^1$ | $\lambda^3$ | 0 | $\lambda^2$ | $\lambda^2$ | 0 | $\lambda^1$ |

For $\lambda = 0.8$, the matching score of each pair is calculated as follows:

$lq = \lambda^2 \times a_1 \times a_2 \longrightarrow (0.8)^2 \times (0.22112) \times (0.11056) = 0.015646$.

:

$qa = \lambda^4 \times a_1 \times a_2 \longrightarrow (0.8)^4 \times (0.11056) \times (0.11056) = 0.005007$.

The total score $S_{s,t}$ in this case is, $0.015646 + \cdots + 0.005007$.

It is obvious that the pair $lq$ yields a higher score than $qa$ since the match is exact (contiguous). This is clearly demonstrates that the proposed method is able to capture the potential similarity between the two sequences.

### 2.3.2 Representation of the protein sequence

In the feature extraction step, we represent each conotoxin protein sequence by a fixed-length of feature vectors. Each coordinate of this feature vector is typically the matching score $S_{s,t}$ as calculated in section 2.3.1. In this case each of the 176 conotoxin protein sequence in the dataset was compared and matched against the rest of the protein sequences. For instance, if we have a protein sequence $s$ then the corresponding score will be $F_s = f_{s_0}, f_{s_1}, \ldots, f_{s_{l-1}}$, where $l$ is the total number of proteins and $f_{s_i}$ is the $S_{s,i}$ score between sequence $s$ and the $i^{th}$ sequence. This process is illustrated as follow:

| | $s_1$ | $s_2$ | ... | $s_{176}$ | Class |
|---|---|---|---|---|---|
| $s_1$ | $S_{1,1}$ | $S_{1,2}$ | ... | $S_{1,176}$ | A |
| $s_2$ | $S_{2,1}$ | $S_{2,2}$ | ... | $S_{2,176}$ | A |
| : | : | : | ... | : | A |
| $s_{25}$ | $S_{25,1}$ | $S_{25,2}$ | ... | $S_{25,176}$ | A |
| $s_{26}$ | $S_{26,1}$ | $S_{26,2}$ | ... | $S_{26,176}$ | M |
| : | : | : | ... | : | : |
| $s_{176}$ | $S_{175,1}$ | $S_{176,2}$ | ... | $S_{176,176}$ | N |

Following the feature extraction step, each conotoxin protein sequence is now represented by a fixed dimensional feature vector of a length equivalent to the number of the conotoxin protein sequences in the training set. This representation satisfies the important requirement of the SVM input, in that SVM requires that each data instance is represented as a vector of real numbers. The SVM adds to this model the ability to learn from negative examples as well, by discriminating between the conotoxin superfamily members.

## 2.4 Classification

The aim of support vector classification is to devise a computationally efficient way of learning 'good' separating hyper-planes in a high-dimensional feature space. The input vectors are mapped into high-dimensional feature space using kernel functions and a hyperplane is constructed which can separate the different classes [27], [28]. To illustrate the idea of using SVM, let us assume that we would like to recognize conotoxin protein sequences belong to 'A-conotoxin' superfamily from a dataset of proteins contains sequences from various conotoxin superfamilies 'non A-conotoxin'. Let $s = (s_1, s_2, ...s_{|s|})$ denotes the conotoxin protein sequence of length $|s|$, where $s_i \in \{A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V\}$ and $r = (r_1, r_2, ..., r_n)$ denotes the input feature vector, where $r_i \in \Re^n$. The classification of the sequence $s$ into 'A-conotoxin' or 'non A-conotoxin' class finds an optimal mapping from $\Re^n$ space into $\{+1, -1\}$ where $+1$ and $-1$ correspond to 'A-conotoxin' and 'non A-conotoxin' classes, respectively. Let $\{(r_j, q_j),\ j = 1, 2, ..., N\}$ denotes the set of training exemplars, where $q_j$ denotes the desired class ('A-conotoxin' or 'non A-conotoxin') for the input feature vector $r_j$ of sequence $s_j$; $N$ denotes the number of training sequences.

SVM first transforms the input to a higher dimensional space with a kernel function and then linearly combines them with a weight vector $w$ to obtain the output [29].

In the classification step, SVM constructs a discriminant function by solving the following optimization problem:

Minimize

$$\frac{1}{2} w^T w + C \sum_{j=1}^{N} \xi_j \tag{1}$$

subject to the constrains

$$q_i(w^T \phi(r_j) + b) \geq 1 - \xi_j,\ \xi_j \geq 0 \tag{2}$$

where slack variables $\xi_j$ represent the magnitude of the classification error, $\phi$ represents the mapping function to a higher dimension $n$, $b$ is the bias used to classify the protein

samples and $C(>0)$ is the regularization parameter that decides the trade-off between the training error and the margin of separation [27].

The minimization of the above optimization problem is equivalent to maximizing the following quadratic function:

$$max_\alpha \sum_{j=1}^{N} \alpha_j - \frac{1}{2} \sum_{j=1}^{N} \sum_{i=1}^{N} \alpha_j \alpha_i q_j q_i K(r_j, r_i) \quad (3)$$

subject to $0 \leq \alpha_j \leq C$ and $\sum_{j=1}^{N} \alpha_j q_j = 0$.

The function $K(r_j, r_i)$ in this case is called the kernel function.

Once the parameters $\alpha_j$ are obtained from the optimization, the resulting discriminant function $f$ is given by

$$f(r_i) = \sum_{j=1}^{N} q_j \alpha_j K(r_j, r_i) + b = w^T \phi(r_i) + b \quad (4)$$

where bias $b$ is chosen so that $q_j f(r_j) = 1$ for all $j$ with $0 < \alpha_j < C$. The class corresponding to the input pattern $r_i$ is 'A-conotoxin' if $f(r_i) > 0$ or 'non A-conotoxin' if $f(r_i) < 0$.

In this study, the Radial Basis Function (RBF) kernel was employed which is formulated as follows:

$$K(r_j, r_i) = exp(-\gamma ||r_j - r_i||^2) \quad (5)$$

where $\gamma(>0)$ is the scaling parameter. The RBF kernel non-linearly maps samples into a higher dimensional space, therefore, unlike the linear kernel, it can handle the case when the relation between class labels and attributes is non-linear [30].

# 3. EXPERIMENTAL WORK AND RESULTS

We investigated the ability of Toxin-AAM method to classify conotoxin superfamilies. In our experimental work, we tested the performance of Toxin-AAM on the dataset described in Section 2.1. A jackknife cross validation test was used as it is deemed the most rigorous among others and hence it has been widely adopted by researchers [6, 31, 32]. The performance of Toxin-AAM was measured by how well the system can recognize members of any of the conotoxin superfamilies. Recall (RE) , Precision (PR) and accuracy (AC) are used as evaluation measures of the performance and they are calculated as follow:

$$RE = \frac{true\,positives(TP)}{true\,positives(TP) + false\,negatives(FN)} \quad (6)$$

$$PR = \frac{true\,positives(TP)}{true\,positives(TP) + false\,positives(FP)} \quad (7)$$

$$AC = \frac{true\,positives(TP) + true\,negatives(TN)}{total\,number\,of\,examples(n)} \quad (8)$$

In the feature extraction step the weight decaying factor $\lambda$ was set to 0.8. LIBSVM (Library for Support Vector Machines) [1] [33] as implemented in WEKA [34] was employed to discriminate between proteins from different conotoxin superfamilies. In all of the experimental works, the scaling parameter of the RBF kernel $\gamma$ was set to 0.1, the loss function was 0.1 and the penalty parameter $C$ was set to 10. The training and testing attributes were linearly scaled to the range between $-1$ and $+1$ prior to applying the SVM. The main advantage of this scaling is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Another advantage of scaling the data is to avoid numerical difficulties during the calculation [29]. As kernel values usually depend on the inner products of feature vectors, e.g. the linear kernel and the polynomial kernel generate large attribute values that may cause numerical problems.

In Table 2 we recorded the performance results of the A, M, O and T conotoxin suberfamilies classification.

**Table 2: Toxin-AAM results.**

| Superfamily | RE | PR | AC |
|---|---|---|---|
| A | 0.957 | 0.955 | 0.9545 |
| M | 0.966 | 0.966 | 0.9659 |
| O | 0.891 | 0.892 | 0.892 |
| T | 0.966 | 0.966 | 0.9659 |
| N | 0.95 | 0.93 | 0.94 |

## 3.1 Comparison to the existing methods

The BLAST algorithm was tested by Mondal et al. [6] to scan against the non-redundant Swiss-Prot database containing 202,310 sequences. The accuracy values for identifying the members of A, M, O and T superfamilies were 88.0%, 69.2%, 85.2% and 11.8% respectively. Thus, it can be interpreted from the performance that the BLASTP tool for searching homologues is not suitable for the hyper variable conotoxins. Therefore, it was imperative to use a superior classification system.

In Table 3, we further compare the performance of the Toxin-AAM to other several methods such as multi-class SVMs, One-versus-rest SVMs, Least Hamming distance and ISort predictor to classify A, M, O, T and N subsets of peptides. Table 3 shows that Toxin-AAM was able to add considerable classification accuracy.

To evaluate the feature extraction step, we replaced the matching algorithm discussed in section 2.3.1 with the Smith-Waterman (SW) algorithm as implemented in Fasta [7]. The SW [35] has undergone two decades of empirical optimization in the field of bioinformatics and thus, considerable prior knowledge is implicitly incorporated into the pairwise sequence similarity scores [36]. To study the effect of incorporating AAC knowledge in the matching algorithm, we removed the values of $a_i$ and $a_j$ in step 6 of the matching algorithm 2.3.1. The comparison results shown in Table 4 indicate that Toxin-AAM method is able to outperform the traditional pairwise method. By incorporating AAC, the Toxin-AAM method was able to add reasonable classification accuracy.

We assess the statistical significance of differences among methods using a two-tailed signed $t$-test. As shown in Table 5, the performance difference of Toxin-AAM in terms of $RE$ is statistically significant at a threshold of 0.05 (in bold). The resulting induced performance ranking of methods is Toxin-AAM, Multi-class SVMs, One-versus-rest SVMs, Least

[1] http://www.csie.ntu.edu.tw/ cjlin/libsvm

**Table 3: A performance comparison of Toxin-AAM to other existing methods.**

| Method | A RE (PR) | M RE (PR) | O RE (PR) | T RE (PR) | Average RE (PR) |
|---|---|---|---|---|---|
| Toxin-AAM | 0.957 (0.955) | 0.966 (0.966) | 0.891 (0.892) | 0.966 (0.966) | 0.945 (0.945) |
| Multi-class SVMs | 0.840 (0.955) | 0.920 (0.800) | 0.870 (0.869) | 0.940 (0.940) | 0.893 (0.891) |
| One-versus-rest SVMs | 0.840 (0.955) | 0.846 (1.000) | 0.820 (0.962) | 0.765 (0.929) | 0.818 (0.962) |
| Least Hamming distance | 0.800 (0.667) | 0.539 (0.539) | 0.771 (0.723) | 0.824 (0.824) | 0.734 (0.688) |
| ISort | 0.760 (0.792) | 0.692 (0.600) | 0.705 (0.683) | 0.882 (0.790) | 0.76 (0.716) |

**Table 4: A performance comparison of Toxin-AAM to Toxin-AAM without incorporating AAC and with the Smith-Waterman (SW) algorithm as implemented in Fasta.**

| Method | A RE (PR) | M RE (PR) | O RE (PR) | T RE (PR) | Average RE (PR) |
|---|---|---|---|---|---|
| Toxin-AAM with AAC knowledge | 0.957 (0.955) | 0.966 (0.966) | 0.891 (0.892) | 0.966 (0.9659) | 0.945 (0.9447) |
| Toxin-AAM without AAC knowledge | 0.736 (0.858) | 0.926 (0.858) | 0.427 (0.653) | 0.816 (0.903) | 0.726 (0.818) |
| Fasta | 0.840 (0.955) | 0.846 (1.000) | 0.820 (0.962) | 0.765 (0.929) | 0.8685 (0.908) |

Hamming distance and ISort. Multi-class SVMs is also performed significantly better then ISort. In Table 6, we show the statistical significance in terms of $PR$. In this case Toxin-AAM, Multi-class SVMs and One-versus-rest SVMs performed significantly better than Least Hamming distance and ISort.

# 4. DISCUSSION AND CONCLUSION

In this study, we introduced a novel method (Toxin-AAM) for conotoxin superfamily classification. The method is based on combining a sequential model and a discrete model. The combination of the sequential model and the discrete model has made the Toxin-AAM method superior in classifying conotoxin superfamily when compared with other state-of-the-art techniques. The method has also shown significantly improved results when compared to traditional sequence similarity search techniques such as Fasta. The reason behind these improvements is the fact that Toxin-AAM compares each pair of amino acids (not necessarily contiguous) that occurs in the two sequences of interest. For instance, if we compare two sequences $s = lqlwa$ and $t = lqal$ using the SW algorithm then the results are as follows:

$$\begin{array}{lll} \boldsymbol{s} & lg\text{-}l & 3 \\ & :: : & \\ \boldsymbol{t} & lg\text{-}l & 4 \end{array}$$

**Figure 2: Comparison results of the sequences $s = lqlwa$ and $t = lqal$ using SW algorithm.**

It is clear that some of the sequence information has not been taken into consideration. Additional sequence information such as comparing $q - -a$ in sequence $s$ to $qa$ in sequence $t$ and $l - - - a$ in sequence $s$ to $l - a$ in sequence $t$ are not included (please refer to Table 1 to see the matching of these strings using Toxin-AAM).

Besides accuracy improvement, the Toxin-AAM method also shows improved efficiency. The amino acid matching algorithm was implemented on an x86 quadcore PC using Microsoft Visual Studio.NET 2003 with fast code optimization. The implementation of the Toxin-AAM launches 16 parallel threads which are allocated an equal number of amino acid pairs to match and which generate the scores of pairs of sequences assigned to them. One significant source of efficiency is that our implementation restricts the generation of substrings (pairs of amino acids) to only those with an inter-distance of 8 or less. This restriction reduces tremendously the execution time, without compromising the score accuracy. Our implementation also benefited from fast code and SIMD vectorization compilation. The total computational cost of the algorithm is $O(|s| \times |t|)$ for computing the similarity score of any pair of sequences. In the parallel load-balanced implementation, this cost is roughly divided by the number of cores (up to the number of threads limit, 16) on which the work is distributed upon. When matching multiple pairs of sequences, this cost is multiplied by the numbers of pairs of protein sequences to match. Moreover, parallelism (via throughput computing) can be employed at the outer level, whereby multiple protein sequence matches are simultaneously performed on different computers, further reducing the cost by the number of pairs of sequences to match. The CPU time for computing the matching scores between all the 176 protein sequences is approximately one second.

Finally, the results reported here may provide a pointer for potentially newer applications of in-silico methods that address problems in the burgeoning, but computationally less explored field of conotoxins. While this discriminative framework is specially developed for identifying conotoxin superfamily members, it naturally extends to other problems in bio-sequence analysis, such as the identification and classification of promoters, protein remote homology, splice sites, and other features in genomic DNA.

Currently, the proposed method extracts all the possible amino acid pairs which could occur anywhere in the two sequences of interest, in the future, we will explore the possibility of extracting substrings of various amino acids lengths.

# 5. ACKNOWLEDGMENTS

The authors would like to acknowledge the support by the Faculty of Information Technology (FIT) and the Office of Research Support and Sponsored Projects (RSSP), United Arab Emirates University (UAEU), UAE. Special thanks to Sukanta Mondal for making the conotoxin superfamily datasets available for research.

**Table 5: Statistical significance of the differences of predictive performance in terms of the Recall ($RE$) between pairs of classification methods.**

| Method | Multi-class SVMs | One-versus-rest SVMs | Least Hamming distance | ISort |
|---|---|---|---|---|
| Toxin-AAM | **0.038** | **0.017** | **0.019** | **0.004** |
| Multi-class SVMs | | 0.059 | 0.054 | **0.012** |
| One-versus-rest SVMs | | | 0.237 | 0.282 |
| Least Hamming distance | | | | 0.536 |
| ISort | | | | |

**Table 6: Statistical significance of the differences of predictive performance in terms of the Precision ($PR$) between pairs of classification methods.**

| Method | Multi-class SVMs | One-versus-rest SVMs | Least Hamming distance | ISort |
|---|---|---|---|---|
| Toxin-AAM | 0.141 | 0.398 | **0.007** | **0.003** |
| Multi-class SVMs | | 0.137 | **0.003** | **3.627E-05** |
| One-versus-rest SVMs | | | **0.009** | **0.006** |
| Least Hamming distance | | | | 0.415 |
| ISort | | | | |

# 6. REFERENCES


[1] H Terlau and B M Olivera. Conus venoms: a rich source of novel ion channel-targeted peptides. Physiol. Rev., 84:41-68, 2003.

[2] S Mouhat, B Jouirou, A Mosbah, M D Waard, and J M Sabatier. Diversity of folds in animal toxins acting on ion channels. Biochem. J., 378:717-726, 2004.

[3] J M McIntosh and R M Jones. Cone venom: from accidental stings to deliberate injection. Toxicon, 39:1447-1451, 2001.

[4] R M Jones and G Bulaj. Conotoxins – new vistas for peptide therapeutics. Curr. Pharm. Des., 6:1249-1285, 2000.

[5] W Rajendra, A Armugam, and K Jeyaseelan. Toxins in anti-nociception and anti-inflammation. Toxicon, 44:1-17, 2004.

[6] S Mondal, R Bhavna, R M Babu, and S Ramakumar. Pseudo amino acid composition and multi-class support vector machines approach for conotoxin superfamily classification. Journal of Theoretical Biology, 243:252-260, 2006.

[7] W R Pearson. Rapid and sensitive sequence comparisons with fastp and fasta. Methods Enzymol., 183:63–98, 1985.

[8] S F Altschul, W Gish, W Miller, E W Myers, and D J Lipman. A basic local alignment search tool. J. Mol. Biol., 215:403-410, 1990.

[9] S F Altschul, T L Madden, A A Schäffer, J Zhang, Z Zhang, W Miller, and D Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. Nucleic Acid Research, 25:3389-3402, 1997.

[10] R I Sadreyev. Compass server for remote homology inference. Nucleic Acids Res., pages 653-658, 2007.

[11] R C Edgar and K Sjolander. A comparison of scoring functions for protein sequence profile alignment. Bioinformatics, 20:1301-1308, 2004.

[12] J Soding. Protein homology detection by hmm-hmm comparison. Bioinformatics, 21:951-960, 2005.

[13] S Han. Fold recognition by combining profile-profile alignment and support vector machine. Bioinformatics, 21:2667-2673, 2005.

[14] H Rangwala and G Karypis. Profile-based direct kernels for remote homology detection and fold recognition. Bioinformatics, 21:4239-4247, 2005.

[15] D Kim. Prospect ii: protein structure prediction program for genomescale applications. Protein Eng., 16:641-650, 2003.

[16] C J Yona and G Ku. The distance-profile representation and its application to detection of distantly related protein families. BMC Bioinformatics, 6:282, 2005.

[17] T Ohlson and A Elofsson. Profnet, a method to derive profile-profile alignment scoring functions that improves the alignments of distantly related proteins. BMC Bioinformatics, 6:253, 2005.

[18] K C Chou. Prediction of protein cellular attributes using pseudoamino acid composition. Proteins, 44:246-255, 2001.

[19] K C Cai and Y D Chou. Prediction of protease types in a hybridization space. Biophys. Res. Commun., 339:1015-1020, 2006.

[20] H Nakashima, K Nishikawa, and T Ooi. The folding type of a protein is relevant to the amino acid composition. J. Bio. Chem., 99:152-162, 1986.

[21] P Y Chou. Prediction of protein structural classes from amino acid composition. Springer, pages 549-586, 1989.

[22] K Cramer and Y Singer. On the algorithmic implementation of multiclass kernel-based vector machines. J. Mach. Learn. Res., 2:265-292, 2001.

[23] F. Tekaia, E. Yeramian, and B. Dujon. Amino acid composition of genomes, lifestyles of organisms, and evolutionary trends: a global picture with correspondence analysis. Gene, 297(4):51–60, 2002.

[24] S. Roy, D. Martinez, O. Platero, T. Lane, and M. Werner-Washburne. Exploiting amino acid composition for predicting protein-protein interactions. PLoS One, 11(4), 2009.

[25] A Bairoch, B Boeckmann, S Ferro, and E Gasteiger. Swiss-prot: juggling between evolution and stability. Brief. Bioinform., 5:39-55, 2004.

[26] W Li, L Jaroszewski, and A Godzik. Clustering of highly homologous sequences to reduce the size of large protein databases. Bioinformatics, 17:282-283, 2001.

[27] V N Vapnik. Statistical Learning Theory. Adaptive and learning systems for signal processing, communications, and control. Wiley, 1998.

[28] N Cristianini and J Shawe-Taylor. An introduction to Support Vector Machines. Cambridge University Press, 2000.

[29] J Ma, M N Nguyen, and J C Rajapakse. Gene classification using codon usage and support vector machines. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 6(1):134-143, 2009.

[30] C.W. Hsu, C.C. Chang, and C.J. Lin. A practical guide to support vector classification. Dept of Comp. Sci., National Taiwan Uni., 2003.

[31] L Nanni and A Lumini. A genetic approach for building different alphabets for peptide and protein classification. BMC Bioinformatics, 9:45, 2008.

[32] L Hao and L Qian-Zhong. Predicting conotoxin superfamily and family by using pseudo amino acid composition and modified mahalanobis discriminant. Biochemical and Biophysical Research Communications, 354:548-551, 2007.

[33] C Chih-Chung and L Chih-Jen. Libsvm : a library for support vector machines. Software, 2000.

[34] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I.H. Witten. The weka data mining software An update. SIGKDD, 11(1), 2009.

[35] T F Smith and M S Waterman. Identification of Common Molecular Subsequences. J.mol.Biol., 147:195-197, 1981.

[36] N M Zaki, S Lazarova-Molnar, W El-Hajj, and P Campbell. Protein-protein interaction based on pairwise similarity. BMC Bioinformatics, 10, 2009.