J. Biomedical Science and Engineering, 2010, 3, 785-790 **JBiSE**
doi:10.4236/jbise.2010.38104 Published Online August 2010 (http://www.SciRP.org/journal/jbise/).

Categorizing HIV-1 subtypes using an ant-based clustering algorithm

David King*, Wei Hu

Department of Computer Science, Houghton College, Houghton, USA.
Email: David.King@Houghton.edu

Received 28 May 2010; revised 17 June 2010; accepted 22 June 2010.

ABSTRACT

Human Immunodeficiency Virus (HIV) is especially difficult to treat due to its rapid mutation rate. There are currently eleven different genomic subtypes of HIV-1, as well as a number of recombinant subtypes. An area of study in bioinformatics is the development of algorithms to identify the subtypes of HIV-1 genomes. Ant-based algorithms have the ability to find global solutions in optimizations problems, and are also able to process complex data efficiently. We proposed a new technique named Ant Colony Anchor Algorithm (ACAA), using anchors of training data on a topographic map to categorize HIV-1 sequences based on ant-based clustering. We used three sets of sequences from the POL region of the HIV-1 genome. We categorized these three dataset with the Subtype Analyzer (STAR), a current HIV-1 categorization algorithm, and the ACAA. We found that the ACAA returned higher accuracy values of 83.2%, 67.1%, and 53.5% for our three datasets respectively, than the STAR's 47.3%, 49.4% and 18%. The results of the ACAA are the average results of 20 runs of the algorithm. We also observed the performance of the algorithm on specific subtypes, and observed that while the STAR and ACAA performed with similar accuracy on several subtypes (A, B, and C in particular), the ACAA had a significant advantage over the STAR in others, especially in categorizing recombinant subtypes.

Keywords: Ants; Clustering; HIV; Classification; Subtype; STAR; ATTA; Bioinformatics

1. INTRODUCTION

1.1. HIV-1 Subtype Categorization

Human Immunodeficiency Virus (HIV), which causes Acquired Immunodeficiency Syndrome (AIDS), falls into two broad categories: HIV-1 and HIV-2. HIV-2 is largely constrained to West Africa, and has a relatively lower transmission rate. HIV-1, on the other hand, has a very high transmission rate, and accounts for most HIV infections on a global scale.

Treatment of HIV-1 is a difficult process because of the virus' exceptionally high mutation rate. When DNA is replicated in the reproductive cycle of the virus, anywhere between 1 in 1000 and 1 in 10000 nucleotides will be improperly transcribed. This has lead to an exceptionally diverse pool of viruses, genetically speaking. HIV-1 has been classified into 11 subtypes based on genomic patterns and variations (subtypes A-H, J, N, and O).

Additionally, there are recombinant types of these viruses, which contain combined DNA from two or more viruses. These recombinant subtypes are caused by the infection of a single cell by multiple viruses [1].

There is neither a cure for HIV-1 nor a preventative vaccine. Antiretroviral drugs have been shown to be an effective treatment. These drugs interfere with the DNA of the virus. As such, the genetic makeup of the virus has tremendous effect on its resistance to the treatments. Because the subtype has such a significant impact on the effectiveness of the drugs, categorization of different subtypes of HIV-1 is critical to the treatment process.

The development of an algorithm to effectively categorize HIV-1 is of prime interest in this area of study. One current categorization algorithm is the STAR (Subtype Analyzer), developed by University College London Centre for Virology. The STAR functions by maintaining profiles of these 11 categories. These profiles are calculated based on aligned training sequence data. Each profile is a two-dimensional frequency matrix, where the number of rows in the matrix is the length of the training sequences in amino acid positions, and the number of columns is 20—one for each possible amino acid. These profiles are calculated based on training sequence data, where each position in a subtype profile is the frequency of that particular amino acid at that position in all training sequences of that subtype. A global profile matrix is

also calculated, using the amino acid frequency from all training sequences rather than just a single subtype [1].

Each test sequence to be categorized receives a Z-score based on each subtype profile. The test sequence is categorized as the subtype which gives the score closest to the mean. If the sequence is not within 1.5 standard deviations of the mean for any of the subtypes, the virus is categorized into the lump category, 'unassigned' [1]. This is not an innately helpful category; however neither is it innately harmful. It is preferred to leave a sequence unclassified rather than misclassified.

There are other algorithms, similar in nature to the STAR, which have also been turned towards the purpose of subtype categorization. The SUDI algorithm[1], developed by Los Alamos National Laboratory, uses phylogenic analysis to determine subtype. The HIV Genotyping algorithm[2], developed by NCBI uses both a similarity search as well as a bootscanning similarity process to categorize subtypes. Stanford's HIVseq[3] algorithm categorizes based only on a simple similarity search [2].

1.2. Ant-Based Clustering Algorithms

Ant-based algorithms fall under a category of algorithms which model their behavior from ants in the wild. These algorithms have found effective applications in bioinformatics because of their ability to find global solutions in optimization problems, as well as for their ability to efficiently process complex data quickly [3].

Ant-based clustering algorithms refer specifically to those algorithms which model the clustering behavior of ants in the wild. The behavior of these social insects displays a high level of swarm intelligence. In this instance, the habits of many relatively simple individuals form a pattern of behavior with distinct order and purpose. In the same way that real ants will cluster objects into like piles within a space (e.g., within a colony, eggs will be clustered into one chamber, while food will be clustered into another), the algorithm will cluster like pieces of data into the same area on a two-dimensional topographic map, forming clusters. Clustering data on a two dimensional map can greatly reduce the dimensionality which needs to be analyzed, allowing for much more efficient analysis and evaluation than would be possible with the raw data [4].

Ant-based clustering algorithms were originally proposed in 1991 by Deneubourg *et al.* [5], and were designed to emulate the sorting and clustering behaviors of ants. Early algorithms of this variety were ineffective, but showed enough potential for clustering data that new improvements continued to occur [3].

One such improvement, called ATTA (Adaptive Time-dependent Transporter Ants), was proposed by Julia Handl [3], and is effective at clustering like data points on a two dimensional map. A sample output of the ATTA can be seen in **Figure 1**. The ATTA utilized formulae which were altered from those used in previous algorithms. Several other changes were made to the process of the algorithm which increased the ability of the algorithm to produce a distinct clustering [6].

The goal of this study is to develop a new algorithm based on the ATTA to categorize HIV-1 subtypes, and to compare the performance of our algorithm with that of the STAR (http://www.vgb.ucl.ac.uk/starn.shtml).

2. DATA

2.1. HIV-1 Sequence Data

An analysis of the STAR algorithm used sequence data which includes the entire protease region of the HIV-1 genome and 340 amino acids of the reverse transcriptase (RT), making for a total sequence length of 439. The protease and RT regions of HIV-1 play a role in the reproductive cycle of the virus. Common antiretroviral treatments interfere with the functions of these regions. Because of their medical significance, sequence data of

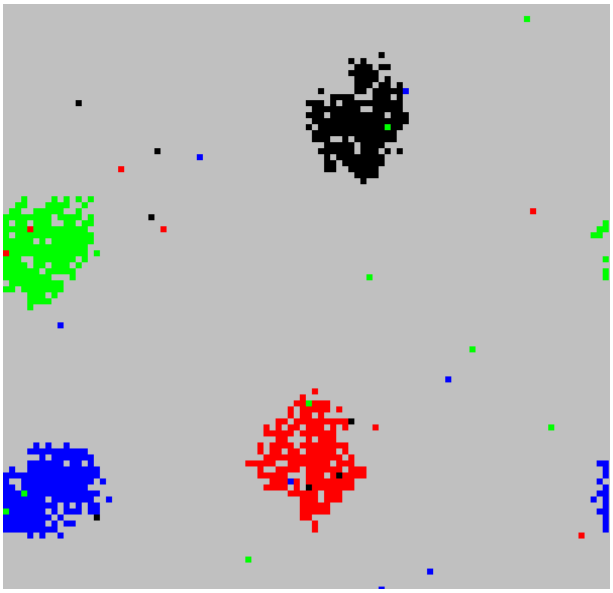

Figure 1. ATTA. This is visual representation of the ATTA. To generate this plot, 800 two-dimensional vectors were generated, each containing values close (within a vector distance of 1) to one of the points [2, 2], [2, 8], [8, 2] or [8, 8]. The points were assigned colors based on which of the four points they were associated with, then clustered using the ATTA.

[1] http://www.hiv.lanl.gov/content/hiv-db/SUDI/sudi.html
[2] http://www.ncbi.nih.gov/projects/genotyping/
[3] http://hivdb.stanford.edu/

protease and RT has a high availability, making them prime for use in categorization algorithms [1]. In this study, we used three datasets, each covering different subtypes and different portions of these genomic regions.

Dataset 1 contained sequences of the same 439 amino acid region as in Myers *et al*. [1]. We include data from 8 subtypes (**Table 1**). We obtained these sequences from the Los Alamos National Laboratory HIV Sequence Database. Each subtype contained no more than 50 sequences and no less than 35.

Dataset 2 contained sequences of a shorter length: 99 amino acids from the protease, and only 240 amino acids from the RT (339 amino acids total, nucleotide positions 2253 to 3270). These were also downloaded from the Los Alamos National Laboratory HIV Sequence Database. 10 subtypes were covered, with sequence numbers between 35 and 50 (**Table 1**).

Dataset 3 contained sequences of the protease genomic region (99 amino acids, ranging from nucleotide positions 2253 to 2550). These were downloaded from the Stanford drug resistance database. This dataset contained 10 subtypes (**Table 1**). Again, each data type had no more than 50 and no less than 35 sequences.

Because misaligned sequences can significantly reduce the effectiveness of sequence analysis, we performed a multiple alignment on all sequences in each dataset using the ClustalW2 algorithm.

In each dataset, every subtype was separated into training and test data. For the purposes of consistency, we used 30 sequences as training data for every subtype. All other sequences (between 5 and 20 in number) were used as test data to be categorized.

All datasets are available on request from the author.

2.2. HIV-1 Sequence Encoding

Each dataset was a collection of sequenced HIV-1 genomes. Each sequence consisted of a list of amino acids. In order to properly analyze these sequences, we converted each into a binary vector. Because there are twenty different possible amino acids, we mapped each amino acid to a unique twenty-dimensional binary vector. For example, we mapped the amino acid Alanine to the point [1, 0, 0, 0, 0...], while we mapped Arginine to [0, 1, 0, 0, 0...]. After all the amino acids in a sequence were converted, we concatenated these individual vectors into an ordered, high-dimensional sequence vector, with dimensionality of 20N, where N is the number of amino acids in the sequence. In this manner, each unique amino acid was represented in the sequence vector.

3. METHODS

3.1 The ATTA Algorithm

The ATTA takes as input a collection of vectors, called documents—a term used by the ATTA. These documents are distributed randomly across a two dimensional grid called a topographic map. The vector data of the documents bears no relation to their initial coordinates on the topographic map. The goal of the algorithm is to cluster the documents on the topographic map such that the documents which are similar to one another will be located near one another on the map.

The ants are simple entities which have the ability to wander freely across the map. They are able to pick up, carry, and drop documents as the algorithm runs. The probability that a document is picked up or dropped by an ant is dependent on the nearby documents.

For each iteration of the algorithm, a random ant from the colony is chosen. The ant moves to a different location on the map within a user-defined distance (our maximum movement distance was 50). At the beginning of the algorithm, this movement is randomly generated, but later is based partially on previous movements of that ant. The ant attempts to drop the document it is carrying based on a probability calculated from the *pdrop* formula. If the drop action is successful, the ant will choose a new document at random, and attempt to pick with a probability calculated from the *ppick* formula. If this operation is unsuccessful, the ant will choose another document at random, and try again. This continues until the ant has successfully picked a new document [3].

The *ppick* and *pdrop* formulae are defined as:

$$ppick = \begin{cases} 1.0 & if \quad f^*(i)^2 \le 1 \\ \dfrac{1}{f^*(i)^2} & else \end{cases}$$

$$pdrop = \begin{cases} 1.0 & if \quad f^*(i)^2 \ge 1 \\ f^*(i)^4 & else \end{cases}$$

Table 1. This table contains the subtypes used for each dataset, as well as the number of test data sequences for each subtype. Any subtype with the heading N/A was not included in that dataset.

Subtype	A	B	C	D	F	F1	G	J	K	O	U	AE	AG
Dataset 1	20	20	20	20	N/A	7	19	N/A	N/A	N/A	N/A	20	20
Dataset 2	20	20	20	20	N/A	20	20	N/A	N/A	8	10	20	20
Dataset 3	20	20	20	20	20	N/A	20	20	20	N/A	N/A	20	20

where

$$f^*(i) = \begin{cases} \frac{1}{\sigma^2}\sum_J \left(1 - \frac{\delta(i,j)}{\alpha}\right) & \textbf{if } f^*(i) > 0 \wedge \forall j \left(1 - \frac{\delta(i,j)}{\alpha}\right) > 0 \\ 0 & \text{otherwise} \end{cases}$$

where i is the candidate document, σ is the neighborhood size, J is the set of documents in the neighborhood, j is the current document in J, $\delta(i, j)$ is the vector distance between document i and document j, and α is a user-defined constant. It is important to note that since the vector information of all documents is constant, $\delta(i, j)$ need only be calculated once for all possible document pairs. Because the algorithm merely references pre-calculated values, it is able to process even high dimensional vectors very efficiently.

The value of f^* becomes larger as i becomes more similar with the other elements in J. As such, *ppick*, which has an inverse relation to f^*, becomes smaller as f^* increases, leading to a dramatically decreased likelihood that the element is picked up when it is nearby similar elements. By contrast, *pdrop* skyrockets when f^* increases, leading to a very high likelihood that an element will be dropped amongst like elements. These formulae are specific to the ATTA, and are revised from those used in previous ant-based clustering algorithms [7].

In this manner, after a large number of successive iterations, the documents with similar vectors are likely to have similar Euclidian positions, forming clusters of like documents on the map.

The output of the ATTA is a clustering of the input vectors. The quality of these clusters is measured by a Pearson Correlation, which evaluates the correlation between the distances of two documents on the topographic map and the distances between the vectors of the two documents [7].

3.2 Modifications to the ATTA

The ATTA is a clustering algorithm, not originally intended for categorization purposes. Where the ATTA takes in all documents at once, and clusters them all in the same manner, categorization algorithms build a classifier based on training data, and then apply the classifier to the test data to categorize. Using the ATTA as a base, we built a categorization algorithm to categorize input test sequences using a classifier build from training sequences.

Our classifier takes the form of anchors of training data in fixed positions on the topographic map. We assign each subtype of training data a unique square region on the topographic map, called the anchor area. The Euclidian positions of the training documents are fixed.

This anchoring technique ensures that the training documents of each subtype are well separated, and will not move through the clustering phase of the algorithm. As a result, the ants will tend to drop test documents of a given subtype near the pre-defined anchor of the same subtype. We call this the Ant Colony Anchor Algorithm (ACAA). A sample output of the ACAA is visible in **Figure 2**. The pseudo-code for the ACAA is available in Section 1 of the supplementary file.

3.3 Subtype Categorization Based on Clustering

When running the ACAA on HIV-1 sequences, we converted each sequence to a binary vector as in Subsection 2.2. Each document in the ACAA represents one HIV-1 sequence.

After the clustering process of the ACAA is finished, we defined a local area for each test document. The radius of this local area was one half the width of the anchor areas. Each test document was classified as the subtype which has the highest representation of training documents within the test document's local area. This is a modified version of the classification strategy used in the study by Lee *et al*. [8].

Because the ACAA is non-deterministic, we utilized repetition to stabilize the results of the categorization. We recorded each test sequence's predicted subtype in each repetition of the ACAA. There are a total of 20 repetitions in our experiment—further repetitions did not

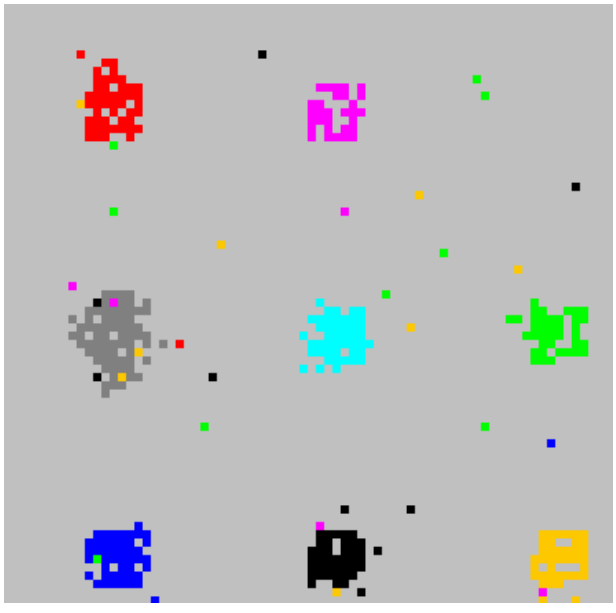

Figure 2. ACAA. This figure clearly shows the anchors of training data on the topographic map, with test data clustering around the training anchors of appropriate type.

improve performance significantly. The final categorization is based on majority votes.

4. RESULTS

We categorized our test data using the ACAA and the STAR's online interface. For each dataset, we maintained a record of the number of sequences correctly classified, the number incorrectly classified, and the number left unclassified.

The STAR is deterministic in nature, and so the categorization was only run a single time. Because the ACAA is non-deterministic in nature, we ran the algorithm on each dataset twenty times, taking the average values of the twenty runs. We also recorded the individual run which produced the highest rate of correct categorization (**Table 2**).

The ACAA yields a more accurate classification than the STAR for all datasets. For the ACAA, the lengthier Dataset 1 yields the best results, while Dataset 2 still gives reasonable accuracy. The STAR was able to categorize both Datasets 1 and 2 with approximately the same level of accuracy. In both the STAR and the ACAA, the categorization of Dataset 3 was not very successful, although the ACAA did categorize with a much higher accuracy than the STAR.

In all datasets, we observe that the ratio of inaccuracy to unclassified sequences is higher in our algorithm than in the STAR. In addition, in Datasets 2 and 3, we observe a higher number of misclassified sequences in the ACAA than in the STAR.

The statistical significance of these results is undeniable—every accuracy rating of the ACAA had a P-value of 0.0. The accuracies generated by the STAR had P-values of 0.0 for Datasets 1 and 2, and 0.004 for Dataset 3.

The statistical significance was determined by generating 10,000 randomized classification predictions for each dataset, and comparing the accuracy of each against the accuracy of the prediction generated by the ACAA. The statistical significance was the percentage of randomly generated predictions with accuracies greater than those produced by the ACAA.

In addition to analysis on each specific dataset, we also observed distinct differences in the ability of both algorithms to categorize specific subtypes. To measure this, we maintained records of the accuracy for each specific subtype in each dataset (**Table 3**).

In so doing, we observed that there are clear differences in how certain subtypes categorize. Typically the subtypes A-D are easier to categorize than others. The ACAA gives a more accurate classification on recombinant subtypes, especially AE. Other subtypes, particularly J, K, and U were problematic to classify in both instances.

We observed in both algorithms that, while the accuracy for some subtypes increased as sequence length increased, there were some subtypes which were more accurately categorized when the sequence length was shorter—the G subtype, for instance.

Overall, however, we find that the ACAA gives a more consistently accurate categorization for the subtypes used in this study than the STAR gives for all sequence lengths.

Table 2. Results of Analysis. This table shows a synopsis of the performance of the ACAA and STAR. Accuracy is the percentage of sequences correctly classified, inaccuracy is the percentage incorrectly classified, and unclassified is the number of sequences which were not assigned a category. The ACAA results are the average values from 20 runs of the algorithm, with the values in parentheses being the results of the run which.

	STAR accuracy	*STAR inaccuracy*	*STAR unclassified*	*ACAA accuracy*	*ACAA inaccuracy*	*ACAA unclassified*
Dataset 1	47.3%	16.4%	36.3%	83.2% (85.6%)	6.9% (6.1%)	10.1% (8.2%)
Dataset 2	49.4%	16.9%	33.7%	67.1% (70.2%)	19.9% (16.9%)	13.0% (12.9%)
Dataset 3	18%	17%	65%	53.5% (57%)	27.3% (30.5%)	19.2% (19.5%)

Table 3. Subtype specific Results. This table shows the accuracy for each subtype for both the ACAA and the STAR. Results of the ACAA are the average of the same twenty runs as used in **Table 2**.

Subtype	**A**	**B**	**C**	**D**	**F**	**F1**	**G**	**J**	**K**	**O**	**U**	**AE**	**AG**
Dataset 1													
ACAA	86.7%	97.7%	95%	78.2%	N/A	6.4%	46.8%	N/A	N/A	N/A	N/A	100%	100%
STAR	55%	95%	100%	5%	N/A	0%	21%	N/A	N/A	N/A	N/A	0%	80%
Dataset 2													
ACAA	82%	94.7%	49.5%	71.7%	N/A	55%	4.5%	N/A	N/A	100%	1%	100%	99%
STAR	75%	80%	55%	75%	N/A	0%	25%	N/A	N/A	100%	0%	0%	90%
Dataset 3													
ACAA	56.5%	91.2%	85.5%	37%	17.2%	N/A	75.7%	27.2%	8%	N/A	N/A	65%	71.2%
STAR	60%	0%	80%	15%	0%	N/A	25%	5%	0%	N/A	N/A	0%	0%

5. SUMMARY

We proposed to use ant-based clustering as a method to categorize HIV-1 sequences according to subtype. We developed the ACAA algorithm to utilize the approach of anchoring test data on a topographic map in order to categorize test data.

We ran our algorithm on three distinct datasets, containing varied HIV-1 subtypes and sequence lengths. For comparison, we also ran the STAR algorithm, a previously developed algorithm for HIV-1 subtype classification, on each dataset.

We have demonstrated increased accuracy of the ACAA algorithm over the STAR algorithm based on HIV-1 subtype classification. Our results imply that the ACAA is a viable alternative for the algorithmic classification of binary vector-based data.

6. ACKNOWLEDGEMENTS

We would like to thank Julia Handl for providing the original java code of the ATTA, and also for her quick and helpful responses to our enquiries about the code.

We would also like to thank Thomas Leitner of the Los Alamos National Laboratories HIV Database, who also responded to queries in an expedient and helpful manner.

REFERENCES

[1] Myers, E.R., *et al*. (2005) A statistical model for HIV-1 sequence classification using the subtype analyzer (STAR). *Bioinformatics*, **21(17)**, 3535-3540.

[2] Oliveira, T., *et al*. (2005) An automated genotyping system for analysis of HIV-1 and other microbial sequences. *Bioinformatics*, **21(19)**, 3797-3800.

[3] Handl, J. (2003) Ant-based methods for tasks of clustering and topographic mapping: Improvements, evaluation and comparison with alternative methods. Ph.D. Thesis, Friedrich-Alexander University, Erlangen-Nürnberg.

[4] Chen, L., *et al*. (2004) An adaptive ant colony clustering algorithm. *Proceedings of the* 3*rd International Conference on Machine Learning and Cybernetics*, Shanghai, 26-29 August 2004, 1387-1392.

[5] Deneubourg, J.L., *et al*. (1991) The dynamics of collective sorting: Robot-like ants and ant-like robots. *Proceedings of the* 1st *International Conference on Simulation of Adaptive Behavior*: *From Animals to Animats*, MIT Press, Cambridge, **1**, 356-365.

[6] Handl, J., *et al*. (2004) Strategies for increased robustness of ant-based clustering. *Engineering Self-Organising Systems* (*Lecture Notes in Computer Science*), **2977**, 90-104.

[7] Handl, J., *et al*. (2004) Ant-based clustering and topographic mapping. *Artificial Life*, **12(1)**, 35-61.

[8] Lee, M., *et al*. (2007) An ant-based clustering system for knowledge discovery in DNA chip analysis data. *Proceedings of World Academy of Science*, *Engineering and Technology*, **32**, 261-266.

