

A Dynamic Programming Approach to the Study of Protein Sequence Variations

ARTHUR W. CHOU

Bioinformatics Program

Department of Mathematics and Computer
Science

Clark University

950 Main Street, Worcester, MA 01610, USA

E-Mail: achou@clarku.edu

Web: <http://www.cs.clarku.edu/~achou/>

SHAN LU

Laboratory of Nucleic Acid Vaccines

Department of Medicine

University of Massachusetts Medical School

364 Plantation Street, Worcester, MA 01605,

USA

E-Mail: shan.lu@umassmed.edu

Abstract: We propose a dynamic programming method to design efficient algorithms to analyze the genetic variation of gene of interest from different isolates, to search for the pattern and rule of changes in their DNA/protein sequences. In many cases we can achieve linear time ($O(n)$) time bound for the worst case time complexity, instead of the cubic time ($O(n^3)$) for a brute-force approach, where n is the length of the sequence. We apply our algorithms to the analysis of N-linked glycosylation sites of all published gp120 variable regions (V1 to V5) of the envelope glycoproteins of HIV-1 virus and find that there is a strong positive correlation between the length of the region and the number of glycosylation sites in the V1 and V4 loops.

Key-Words: bioinformatics, dynamic programming, protein sequences, HIV virus, envelope glycoprotein.

1. Introduction

Dynamic programming is an important paradigm for the design of algorithms [3]. It explores the sub-structures of problem instances and combines the solutions to sub-problems efficiently to form a solution to the original problem. It is similar to the divide-and-conquer strategy of algorithm design, but differs in two major ways: (1) there are significant overlaps of (solutions to) sub-problems, and a straightforward recursive divide-and-conquer type of algorithm re-computes the solutions to sub-problems redundantly; as a result its time complexity becomes unnecessarily high. (2) We therefore unwind the recursion and track the solutions to sub-problems carefully by storing them in a table, and then synthesize the local solutions into a global one by an efficient table tracing.

Due to the fact that phenomena of sequences can often be studied by combining the information about their sub-sequences, sequence analysis of DNA/Protein sequences provides a fertile ground for the application of dynamic programming strategies. They are used extensively in sequence alignment, Hidden Markov Models, and phylogenetic tree constructions [2, 4]. In this paper we use such a

strategy to study sequence variations in different regions of a sequence and among various subtypes of sequences. As an example, we apply it to study the glycosylation sites of the variable regions (regions V1 to V5) of the envelope (Env) glycoprotein gp120 of the human immunodeficiency virus-1 (HIV-1) virus.

Historically antibody responses have proven critical in inducing protective immunity against a wide range of infectious diseases. Despite efforts in the last two decades, there has been little progress in generating broadly cross-reactive neutralizing antibodies through immunization against HIV-1. The development of a safe and effective vaccine to prevent the transmission of HIV-1 remains a major challenge. It is most likely due to the diversity of the HIV-1 subtypes and the high frequency of mutation of this virus. A detailed analysis on the structure and function of the envelope glycoprotein (Env) from primary HIV-1 isolates is indispensable for the development of HIV-1 vaccines [7]. HIV-1 Env plays an important role in evading any potential humoral immune response and allowing the virus to establish a persistent infection within the host cell. Immunization with the HIV-1 Env glycoprotein gp120 results in production of antibodies that

neutralize T-cell line-adapted (TCLA) viruses, but has only marginal activity against primary isolates of HIV-1[6]. The gp120 cores of both primary and the laboratory-adapted TCLA strains are very similar. Their sequence variability is concentrated in the variable regions, V1 through V5. Therefore, the variable regions appear to be the major targets for the neutralizing antibody responses and are regarded as essential in the neutralization resistance of HIV-1 primary isolates. One important feature for the variable regions is their high glycosylation [5, 7]. In this paper we analyze the distribution of the glycosylation sites of these variable regions and demonstrate that there is a strong positive correlation between the size of the V1 and V4 loops and the level of glycosylation. In another paper [1], we reported on the status of the conjecture that the levels of glycosylation may have contributed to the diversity of the immunogenicity among various primary HIV-1 Env, and the result of experiments showing that mutated HIV-1 Env antigens with selected removal of N-glycosylation sites at the tip of V1 loop indeed induced higher antibody responses against the autologous HIV virus. Our goal is to better understand the structure-function relationship of HIV-1 Env antigens from primary viral isolates, in order to facilitate the development of effective HIV vaccines that could induce broad neutralizing antibody responses.

2. Computational Model

Given a DNA/protein sequence s , we are interested in the occurrences of certain *pattern* in the sequence. For example, in a protein sequence of the HIV-1 Env, the pattern can be the potential N-linked glycosylation site (the amino acid sequences NXS and NXT). We assume that the locations of the elements (amino acid) in the sequence are labeled from 1 to n , where n is the length of the sequence; that is, we consider the sequence s as an array of amino acid symbols (Here we leave the slot at location 0 blank or store the length of the sequence in it, to be consistent with scientists' naming convention.) The pattern appears at certain locations in the sequence, and the locations of the occurrences of the pattern are denoted by s_1, s_2, \dots, s_m , where s_k denotes the starting location of the k -th occurrence of the pattern and m the total number of occurrences. Note that $\{s_k\}$ is an increasing sequence. These locations are stored in an array L : $L[i] = s_i$. We leave $L[0]$ blank.

Example. Take the *A1 consensus sequence* from Gao et al [5]. The glycosylation sites appear at location: 85, 129, 134, 141, 145, 173, 184, Therefore, $s_1=85, s_2=129, s_3=134, \dots$, and so on.

To study the variability of the pattern in different regions of a sequence, we need to examine their local variation in different segments of varying length. The local variations are captured by inspecting *stretches* of increasing length; say, from length 3 to a meaningful upper bound, which could be equal to the length of the sequence. A *stretch* is, by definition, a contiguous block of elements of the sequence. A complete description of the sequence variation can thus be obtained by combining the information about the pattern inside stretches of varying length. The goal of this computational study is to create an efficient data structure to store the information and to support subsequent analysis of the global behavior of the variability throughout the sequence.

The dynamic programming strategy is ideal for the study of the occurrences of patterns in the stretches: two contiguous stretches a and b of length l_1 and l_2 respectively form a stretch $a \circ b$ of length $(l_1 + l_2)$. If the characteristics of the pattern in the combined stretch $a \circ b$ can be *inferred* from that of its two components a and b , then an efficient dynamic programming scheme can be set up to compute the characteristics of stretches of varying length. In this paper we focus on a simple characteristic: *the number of occurrences of a pattern in each stretch* (e.g., the number of glycosylation sites in stretches of varying length of the gp 120 region of the HIV Env protein) and the *distribution or concentration* of the pattern throughout the sequence. To set up such an efficient scheme, we first convert the information about the locations of the pattern to that of the distances between them. We define a *distance matrix* D as follows:

$$D[i, j] = s_j - s_i \text{ if } j > i; D[i, j] = 0 \text{ otherwise; (1)}$$

where $i = 1, 2, \dots, m$, and $j = 1, 2, \dots, m$, and “ m ” is the total number of occurrences of the pattern in the sequence s . Notice that we leave the 0-th row and the 0-th column blank in a typical two-dimensional array implementation.

Example. For the A1 consensus sequence above, we have the matrix $D[i, j]$ as follows:

i \ j	1	2	3	4	...	m-1	m
1	0	44	49	56	...		(m-1)-th diagonal
2	0	0	5	12			(m-2)-th diagonal
3	0	0	0	7			⋮
4	0	0	0	0			2 nd diagonal
⋮							⋮
m-1							1 st diagonal
m							

Figure 1. Distance Matrix

It is clear that

$$D[i, j] = D[i, k] + D[k, j] \text{ for any } k, i < k < j. \quad (2)$$

Inside this table (**Figure 1**) we can define the k -th diagonal to be the entries of the form

$$D[i, i+k] \text{ for } i = 1, 2, \dots, (m-k), \quad (3)$$

$k = 1, 2, \dots, (m-1)$, as marked on the table above (Figure 1). The 0-th diagonal is the entries $D[i, i] = 0$ for $i = 1, 2, \dots, m$. In view of (2), we see that once the 1st diagonal entries, $D[i, i+1]$, $i = 1, 2, \dots, (m-1)$, are computed, the rest can be computed inductively:

$$D[i, i+k] = D[i, i+(k-1)] + D[i+(k-1), i+k] \quad (4)$$

for $k = 2, 3, \dots, (m-1)$, sequentially.

Define a k -tuple of locations $\{s_i\}$ to be a sequence of k consecutive locations $\{s_i, s_{i+1}, \dots, s_{i+(k-1)}\}$, denoted by $[s_i \dots s_{i+(k-1)}]$, and define the *length of the k -tuple* to be the length of the contiguous block occupied by that k -tuple; that is,

$$\text{length}([s_i \dots s_{i+(k-1)}]) = s_{i+(k-1)} - s_i + 1. \quad (5)$$

Then the k -th diagonal contains the information about the lengths of the $(k+1)$ -tuples:

$$D[i, i+k] = \text{length}([s_i \dots s_{i+k}]) - 1. \quad (6)$$

Thus, to compute and number of occurrences of the pattern in the stretches of varying length or to perform statistical analysis of them, we don't need to scan the sequence block by block, from the beginning to the end, which is too time-consuming. Instead, we read the information off the diagonal lines of this $m \times m$ table $D[i, j]$ and store the tuples in a simple

hash table and use these data structures to facilitate the data analysis, as we will describe next.

3. Algorithm and Analysis

Given the distance matrix D defined by (1) and the information stored in it: (2) – (6) (as shown on **Figure 1**), we first describe the data structure and the algorithm to store the k -tuples of different length (defined in (5)) throughout the sequence s . From that we can derive information about the distribution and the concentration of the pattern.

For a fixed length l , to find all occurrences of the pattern (e.g., glycosylation sites) that can be covered by stretches of that length, we proceed as follows:

Procedure *diagonal_search* (l);

- (a) start from the upper right corner of the table D ;
- (b) for each k from $(m-1)$ down to 1
 - search the table along the k -th diagonal from top down;
 - if an entry at index (p, q) : $D[p, q] \leq l-1$ then output the $(k+1)$ -tuple $[s_p \dots s_q]$ and
- (c) ignore the whole quadrant $\{D[i, j]; i < p, j < q\}$ for the subsequent searches;

Because of (5) (6) above and the pruning procedure in step (c), this algorithm outputs all the “maximal” tuples of locations of the pattern that lie within stretches of length l . Moreover, the pruning procedure in step (c) avoids unnecessary searches (and outputs) of smaller tuples contained in the maximal tuples, and speed up the search considerably.

Analysis of Time Complexity. For a sequence s of length n , a brute-force sequential search for patterns using a moving window of size l would cost $O(l \cdot n)$ time, whereas the time complexity of the construction of the table and the *diagonal_search* procedure is no more than $O(n + m^2)$. Here we ignore the cost of outputting the tuples in both cases. The $O(n)$ factor in $O(n + m^2)$ comes from the cost of a linear search of the sequence s to find the locations of the pattern: $\{s_1, s_2, \dots, s_m\}$; that is, to construct the array of locations L . In Practice, “ m ”, the number of occurrences of the pattern in the sequence s , is much less than “ n ”, the length of the sequence. Thus we achieve linear time if $m^2 = O(n)$. Summing over all stretch lengths l from 1 to n , we see that the brute-force method for detecting the occurrences of the pattern for stretches of all lengths costs time complexity $O(n^3)$, but our construction of the

distance matrix $D[i, j]$, together with the following *bucket hashing* data structure, accomplishes the task in time proportional to $(n + m^2)$:

Procedure *bucket_hashing*

- (a) create an array of lists, called *Bucket*, where *Bucket*[i] stores all tuples $[s_p \dots s_q]$ of length i , $i = 1, 2, \dots, n$.
- (b) for each k from $(m-1)$ down to 1
 for each entry $D(p, q)$ on the k -th diagonal from top down
- (c) append the $(k+1)$ -tuple $[s_p \dots s_q]$ to the list: *Bucket*[$D(p, q)+1$];

This way *Bucket*[i] stores all tuples of length i . Notice that step (c) guarantees that the tuples are stored in the same sequential order as they appear on the sequence, since we traverse the diagonals from top down. To obtain all the occurrences of the pattern within stretches of length l , we need only examine *Bucket*[i] for $i \leq l$. This data structure supports data storage and retrieval efficiently.

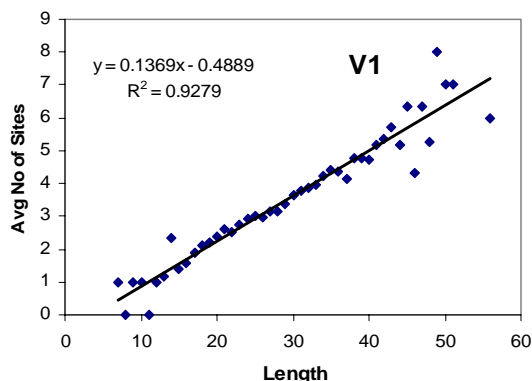
4. Results and Discussion

We implemented the algorithm in Java* and first applied it to study the 9 consensus sequences of primary HIV-1 gp120 glycoproteins compiled by Gao et al [5], as listed in **Table 1** in the *Appendix*. The distribution of N-linked glycosylation sites on these sequences indicated that the V1 region was the most variable region with a wide variation on the number of N-linked glycosylation sites (ranging from 3 to 7 sites), followed by the V4 region (3 to 6 sites) and the other regions varied by only 1 or 2 glycosylation sites (**Table 1**). An interesting observation from our analysis was that, not only the glycosylation sites concentrate more on the V1 and V4 variable regions, there was also a highly linear relationship between stretch length and the number of glycosylation sites in these stretches of such length, for the V1 and V4 regions; that is, the longer the stretches, the more the number of glycosylation sites in them. This phenomenon does not occur anywhere else in the sequences. For the V1 and V4 regions, for stretches of length 15 to 25, the number of glycosylation sites vary from 2 to 4; for stretches of length 26 to 45, the number varies between 4 and 7; where as for other regions, the number varies between 1 and 2. Over the whole sequence, for

stretches of length 45, the average number of glycosylation sites is about 2.5.

In order to see whether this is a general phenomenon, we proceeded to perform the analysis on a large number of samples: we downloaded all published gp120 sequences from the Los Alamos HIV Database (<http://www.hiv.lanl.gov>), a total of 4084 complete gp120 sequences. To obtain the variable regions, we first performed the multiple sequence alignment of them using CLUSTALW (version 1.8), then composed text mining programs to extract the variable regions V1 through V5 from the alignment. Similar data analysis was applied to these variable regions to study the relationship between the stretch length and the number of glycosylation sites for such length. Now because the sample size is large enough and the lengths of variable regions already span a wide range (**Table 2**), we only plot the results for the variable regions (instead of stretches within such regions.)

From this we found not only that the glycosylation occurs significantly more in variable regions V1 and V4 regions, but also that for V1 and V4 there was a remarkably tight linear relationship between the segment length and the average number of glycosylation sites for that length: under linear plot in Excel, the R^2 values were 0.93 and 0.95 for V1 and V4 regions, respectively. Please see **Figure 2** below and **Table 2** in the *Appendix*.



continued onto next page.

* These programs are available upon request.

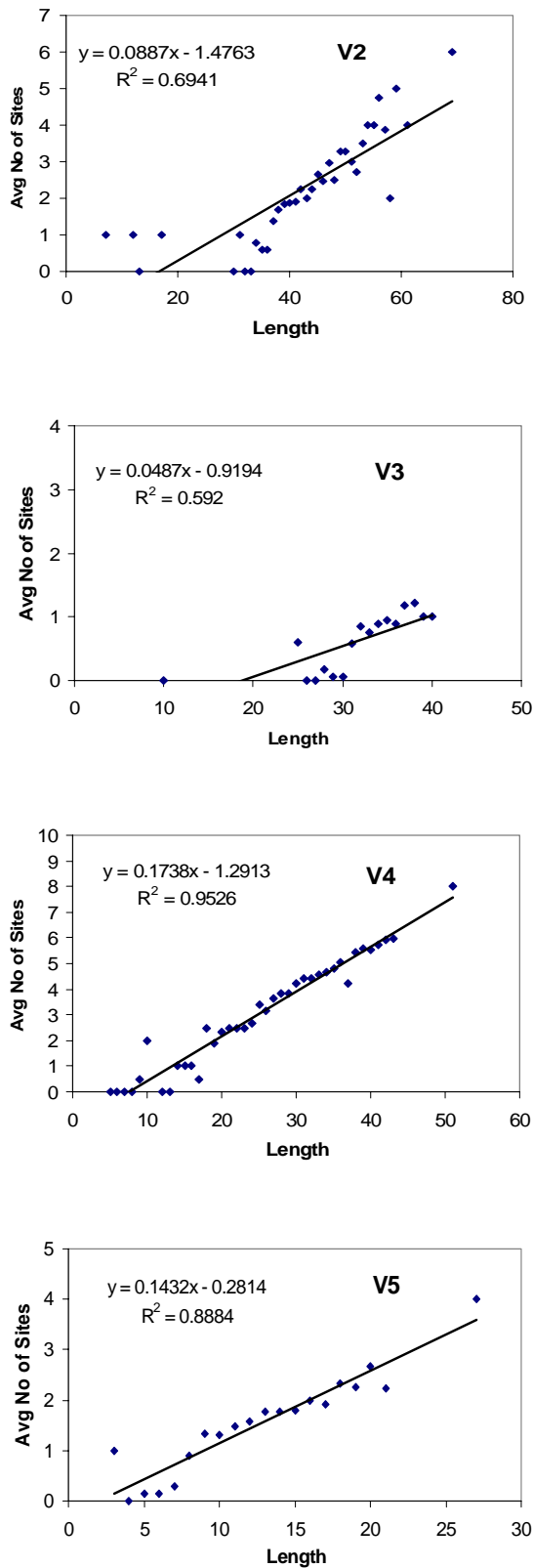


Figure 2. The average number of glycosylation sites plotted against each length of the variable region, for V1 – V5.

Possible explanations of these observations are:

(1) The level of glycosylation, especially in V1 and V4 regions, might contribute to the variability of the variable regions and hence the immunogenicity of the primary HIV-1 Env isolates. It might play a role in their weak neutralizing activity by inaccessibility or absence of relevant epitopes on various primary HIV-1 Env.

(2) Because V1/V2 domain functions as a cover for the underlying CD4 binding site, it is conceivable that heavy glycosylation around V1 region provides further survival advantages for HIV-1 virus.

Further analysis of sequence variability for various clades of HIV-1 Env and carefully designed experiments need to be conducted to understand the relationship between glycosylation and immunogenicity. They improve our understanding of the structure-function relationship of HIV-1 Env antigens and may facilitate the future development of effective HIV vaccines.

References:

- [1] A. Chou, S. Lu and et al, Levels of N-linked glycosylation of the V1 loop of HIV-1 envelope glycoproteins and their relationship to the immunogenicity of HIV Env from primary viral isolates, 2007 (in preparation).
- [2] P. Clote and R. Backofen, *Computational Molecular Biology: An Introduction*, Wiley, 2000.
- [3] T. H. Cormen et al, *Introduction to Algorithms*, MIT Press, 2001.
- [4] R. Durben et al, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, 1999.
- [5] F. Gao et al, Molecular cloning and analysis of functional envelope genes from human immunodeficiency virus type 1 sequence subtypes A through G, *Journal of Virology*, Vol. 70, 1996, pp. 1651-67.
- [6] Mascola, J. R. et al, Immunization with envelope subunit vaccine products elicits neutralizing antibodies against laboratory-adapted but not primary isolates of human immunodeficiency virus type 1. The National Institute of Allergy and Infectious Diseases AIDS Vaccine Evaluation Group. *J. Infect. Dis.*, 1996 173:340-348.
- [7] S. Zolla-Pazner, Identifying epitopes of the HIV-1 that induce protective antibodies, *Nature Reviews Immunology*, Vol.4, March 2004, pp. 199-210.

Appendix.

Table 1. Distribution of N-linked Glycosylation sites on HIV-1 gp120 glycoproteins. Variable regions are named V1 to V5; constant regions are named C1 to C5.

HIV-1 isolates		Distribution of potential N-glycosylated sites in HIV-1 gp120										
<u>clade</u>	<u>Env clone</u>	<u>C1</u>	<u>V1</u>	<u>V2</u>	<u>C2</u>	<u>V3</u>	<u>C3</u>	<u>V4</u>	<u>C4</u>	<u>V5</u>	<u>C5</u>	<u>total</u>
A1	92RW020.5	1	3	2	6	1	2	6	1	2	0	24
A2	92UG037.8	1	4	3	7	1	3	4	1	1	0	25
B	92US715.6	2	6	2	7	1	4	4	1	2	0	29
C1	92BR025.9	1	3	3	8	1	2	4	1	0	0	23
C2	93MW965.26	1	5	3	6	1	3	4	2	1	0	26
D	92UG021.16	1	4	2	7	0	3	4	1	2	0	24
E	93TH976.17	1	7	1	6	0	2	3	1	1	0	22
F	93BR020.17	1	4	2	6	1	3	3	2	2	0	24
G	92UG975.10	1	4	1	6	1	4	3	2	2	0	24
B	HXB2	1	3	2	8	1	3	4	1	1	0	24

Table 2. Summary of sequence analysis of gp120 and variable regions in 4084 published HIV-1 Env sequences. “Size Range” is the range for the lengths of the variable regions V1 – V5. “Glycan Range” is the range for the numbers of glycosylation sites in the corresponding regions. R² values are those of the linear plots in Figure 2.

	Size Range	Glycan Range	R ² value
V1	7-56	0-8	0.93
V2	7-69	0-6	0.69
V3	10-40	0-3	0.59
V4	5-51	0-8	0.95
V5	3-27	0-4	0.89