# CLASSIFICATION OF DENGUE VIRUS SEROTYPES BY VISUALIZATION TOOL

BOONYARAT VIRIYASAKSATHIAN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR
THE DEGREE OF MASTER OF ENGINEERING
(BIOMEDICAL ENGINEERING)
FACULTY OF GRADUATE STUDIES
MAHIDOL UNIVERSITY
2010

Thesis
entitled

# CLASSIFICATION OF DENGUE VIRUS SEROTYPES BY VISUALIZATION TOOL

……………….……………..…..………
Miss Boonyarat Viriyasaksathian
Candidate

……………………….…….…..………
Lect. Yodchanan Wongsawat,
Ph.D. (Electrical Engineering)
Major advisor

……………………….…….…..………
Lect. Prapat Suriyaphol,
Ph.D. (Biochemistry)
Co-advisor

……………………….…….…..………
Lect. Norased Nasongkla,
Ph.D. (Polymer Science)
Co-advisor

……………………….…….…..………
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

……………………….…….…..………
Asst.Prof.Jackrit Suthakorn,
Ph.D. (Robotics)
Program Director
Master of Engineering Program in
Biomedical Engineering
Faculty of Engineering
Mahidol University

Thesis
entitled

# CLASSIFICATION OF DENGUE VIRUS SEROTYPES BY VISUALIZATION TOOL

was submitted to the Faculty of Graduate Studies, Mahidol University
for the degree of Master of Engineering (Biomedical Engineering)
on
July 30, 2010

…………….…………..……..………
Miss Boonyarat Viriyasaksathian
Candidate

…………………….…..………………
Lect. Songpol Ongwattanakul,
Ph. D. (Electrical and Computer Engineering)
Chair

…………………….…..……………..……
Lect. Yodchanan Wongsawat,
Ph.D. (Electrical Engineering)
Member

…………………….…..……………...…
Lect. Prapat Suriyaphol,
Ph.D. (Biochemistry)
Member

…………………….…...………………..
Lect. Norased Nasongkla,
Ph.D. (Polymer Science)
Member

…………………….…..………………..
Lect. Yuttapong Jiraraksopakun,
Ph.D. (Electrical and Computer Engineering)
Member

…………………….…..……………...…
Prof. Banchong Mahaisavariya,
M.D., Dip Thai Board of Orthopedics
Dean
Faculty of Graduate Studies
Mahidol University

…………………….…..……………...…
Asst. Prof. Rawin Raviwongse, Ph.D.
Dean
Faculty of Engineering
Mahidol University

# ACKNOWLEDGEMENTS

I would like to take this opportunity to offer my thanks to all people for their contribution to the development and completion of this thesis. First and foremost, I would like to thank my academic advisor, Dr. Yodchanan Wongsawat. He guided me step by step from beginning until the end of this study, taught me how to write academic papers and made me a better programmer. Besides thesis work, he also gave many valuable opportunities for supporting my attendance at various conferences. I deeply thank my co-advisor, Dr. Prapat Suriyaphol for his suggestion throughout my Master study. His suggestion improved my research to be efficient and be appropriate direction. Without their encouragement and constant guidance, I could not have finished this dissertation. Many thanks go to Dr. Norased Nasongkla and Dr. Yuttapong Jiraraksopakun for suggestion on my thesis writing.

I would also thank my committee: Dr. Songpol Ongwattanakul, Dr. Yodchanan Wongsawat, Dr. Prapat Suriyaphol, Dr. Norased Nasongkla and Dr. Yuttapong Jiraraksopakun for their kindness and valuable suggestions.

I feel thankful for all my teachers, friends and staffs at Department of Biomedical Engineering, Mahidol University for all the support during my coursework and through my academic years. I also thank my friends at Department of Computer Engineering, Prince of Songkla University, for their suggestion and encouragement. My special thanks go to all Dr. Prapat Lab members for their entire helps. Also thank you to Miss Wonphorn Kandhavivorn for suggestion on my thesis writing.

Finally, I am especially thankful to my family for their unconditional support and understanding throughout these years.

Boonyarat Viriyasaksathian

CLASSIFICATION OF DENGUE VIRUS SEROTYPES BY VISUALIZATION TOOL

BOONYARAT VIRIYASAKSATHIAN 5137363 EGBE/M

M.Eng. (BIOMEDICAL ENGINEERING)

THESIS ADVISORY COMMITTEE: YODCHANAN WONGSAWAT, Ph.D. (Electrical Engineering), PRAPAT SURIYAPHOL, Ph.D. (Biochemistry), NORASED NASONGKLA, Ph.D. (Polymer Science)

## ABSTRACT

Software for analysis and interpretation of nucleotide sequences can display the result in different ways: color-coded bar, letter to letter and graphical display, *etc*. However, there is still no efficient software that can simultaneously classify the serotypes of Dengue virus and visually illustrate the nucleotide sequences of Dengue virus. Therefore, this research aimed to study and develop software, called CADViST, for analyzing the nucleotide sequences of Dengue virus with a novel graphical representation, called UnitX. The tasks of the thesis were divided into two parts: (1) design a graphic to represent the nucleotide sequences of Dengue virus, then (2) develop software that provides the following functions: (2.1) developing a nucleotide sequence search engine of the Dengue virus, (2.2) assigning the serotype to the unknown Dengue virus sequence, and (2.3) displaying the Basic Local Alignment Search Tool (BLAST) search result.

This thesis designed UnitX to display the relationship between amino and keto functions in graphical display. For the purpose of verification, UnitX was employed to classify the nucleotide sequences of Dengue virus. All 2,184 sample sequences, from GenBank database of the National Center for Biotechnology Information (NCBI), USA, comprise four serotypes of Dengue virus sequence: DENV1 to 4. Each serotype contains 952, 737, 405 and 90, nucleotide sequences, respectively. The experimental result concluded that UnitX was able to efficiently classify the nucleotide sequences of Dengue virus serotype. Hence, UnitX was able to be used as the graphical representation of the CADViST software for providing a facility to users in nucleotide analysis. For assigning the serotype, one-tenth of each serotype was randomly selected for constructing the template sequence via multiple sequence alignment, called ClustalW. BLAST was employed to provide the serotype to the unknown Dengue virus sequence. The results showed that the template sequence representing each serotype could efficiently distinguish the different serotypes of Dengue virus.

KEY WORDS: BIOINFORMATICS / CLASSIFICATION / DENGUE VIRUS / SEROTYPE / VISUALIZATION / BLAST

106 pages

การจำแนกซีโรทัยป์ของไวรัสเด็งกี่ด้วยโปรแกรมคอมพิวเตอร์แบบวิชวลไลเซชั่น
CLASSIFICATION OF DENGUE VIRUS SEROTYPES BY VISUALIZATION TOOL

บุญรัตน์ วิริยะศักดิ์เสถียร 5137363 EGBE / M

วศ.ม. (วิศวกรรมชีวการแพทย์)

คณะกรรมการที่ปรึกษาวิทยานิพนธ์ : ยศชนัน วงศ์สวัสดิ์, Ph.D. (Electrical Engineering), ประพัฒน์ สุริยผล Ph.D. (Biochemistry), นรเศรษฐ์ ณ สงขลา, Ph.D. (Polymer Science)

## บทคัดย่อ

ซอฟต์แวร์สำหรับการวิเคราะห์และแปลผลลำดับเบสของนิวคลีโอไทด์สามารถแสดงผลได้หลายแบบ เช่น การแสดงผลเป็นแถบสี เป็นตัวอักษรแบบตำแหน่งต่อตำแหน่ง และการแสดงผลแบบกราฟิก เป็นต้น แต่ยังไม่มีซอฟต์แวร์ที่มีประสิทธิภาพในการจำแนกซีโรทัยป์ของเชื้อไวรัสเด็งกี่และทำให้เห็นภาพรวมของสายนิวคลีโอไทด์ทั้งเส้นของเชื้อไวรัสเด็งกี่ ดังนั้น โครงการวิจัยในวิทยานิพนธ์นี้ จึงมุ่งเน้นการศึกษาและพัฒนาซอฟต์แวร์โดยตั้งชื่อว่า CADViST สำหรับใช้ในการวิเคราะห์ลำดับเบสของนิวคลีโอไทด์ของเชื้อไวรัสเด็งกี่ด้วยการนำเสนอข้อมูลในรูปแบบกราฟิกที่ถูกพัฒนาขึ้นมาใหม่ชื่อ UnitX เป้าหมายของงานวิจัยแบ่งออกเป็น 2 ส่วน ดังนี้คือ (1) ออกแบบกราฟิกที่นำมาใช้ในการแสดงลำดับเบสของนิวคลีโอไทด์ของเชื้อไวรัสเด็งกี่ (2) พัฒนาซอฟต์แวร์ให้มีฟังก์ชันการทำงานดังนี้คือ (2.1) เป็นซอฟต์แวร์ที่ใช้ในการสืบค้นหาข้อมูลลำดับเบสของนิวคลีโอไทด์ของเชื้อไวรัสเด็งกี่ (2.2) เป็นซอฟต์แวร์ที่ใช้ในการจำแนกซีโรทัยป์ของลำดับเบสของนิวคลีโอไทด์ของเชื้อไวรัสเด็งกี่ (2.3) เป็นซอฟต์แวร์ที่ใช้แสดงผลจากโปรแกรม Basic Local Alignment Search Tool (BLAST)

งานวิจัยนี้นำเสนอ UnitX ไปใช้ในการแสดงความสัมพันธ์ของปริมาณระหว่างหมู่ฟังก์ชันอะมิโนและคีโตในรูปของกราฟิก เพื่อทดสอบความสามารถของ UnitX UnitX จึงถูกนำไปใช้ในการจำแนกซีโรทัยป์ของลำดับนิวคลีโอไทด์ของเชื้อไวรัสเด็งกี่ นิวคลีโอไทด์ตัวอย่างจำนวน 2,148 ตัวอย่าง ได้มาจาก ฐานข้อมูล GenBank ของ National Center for Biotechnology Information (NCBI) ประกอบด้วย 4 ซีโรทัยป์ คือ ซีโรทัยป์ 1 ถึง 4 แต่ละซีโรทัยป์ประกอบด้วย 952, 737, 405 และ 90 ตามลำดับ ผลจากการทดลองสรุปได้ว่า UnitX สามารถนำมาใช้ในการจำแนกซีโรทัยป์ทั้ง 4 ของเชื้อไวรัสเด็งกี่ได้ เมื่อได้ผลดังนี้ จึงนำ UnitX ไปใช้เป็นหน่วยแสดงผลของซอฟต์แวร์ CADViST ซึ่งเป็นซอฟต์แวร์ที่ถูกพัฒนาเพื่ออำนวยความสะดวกแก่ผู้ใช้ให้สามารถวิเคราะห์ลำดับนิวคลีโอไทด์ได้ง่ายขึ้น 10% ของแต่ละซีโรทัยป์ถูกสุ่มมาทำ Multiple Sequence Alignment เพื่อสร้างสายนิวคลีโอไทด์ของแต่ละ ซีโรทัยป์ด้วยโปรแกรม ClustalW โปรแกรม BLAST ถูกนำมาใช้เพื่อจำแนกซีโรทัยป์ของเชื้อไวรัสเด็งกี่ ผลลัพธ์แสดงให้เห็นว่า 10% นิวคลีโอไทด์ใน 4 ซีโรทัยป์ที่นำมาใช้สามารถระบุซีโรทัยป์ได้ถูกต้องทั้งหมด

106 หน้า

# CONTENTS

## CONTENTS (cont.)

**Page**

## CONTENTS (cont.)

Page

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF FIGURES (cont.)

**Figure** **Page**

# LIST OF FIGURES (cont.)

**Figure** **Page**

# CHAPTER I
# INTRODUCTION

## 1.1 Introduction

Dengue virus is a major threat to health in tropical countries around the world, including Thailand. By this reason, there is a need to develop Dengue virus research. Signal processing methods are used in various fields of research. One of the newest of these methods is in the field of genomics. The idea of genomic signal processing approach is to convert a four letter alphabet, each representing a different nucleotide: A, C, G and T, to a numerical sequence before digital signal processing (DSP) is applied for visualization and analysis.

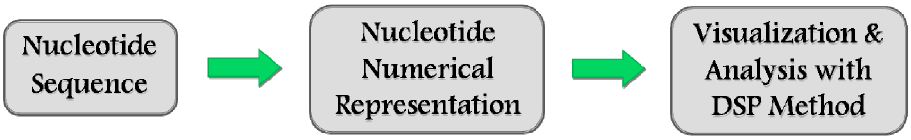

Nucleotide Sequence
Nucleotide Numerical Representation
Visualization & Analysis with DSP Method

**Figure 1.1** Steps of genomic signal processing [1]

In this thesis, visualization tool for analyzing nucleotide sequence of Dengue virus, called UnitX, is proposed. The proposed UnitX can map the four-letter alphabets (A, T, C and G) of DNA sequence to the corresponding numerical values with DSP methods. The software that can employ the proposed visualization tool, called CADViST, is also developed. The benefit of this developed software is to graphically display simultaneously with analyzing the Dengue virus sequence data. A process map provides an overall graphical view of the sequence and quickly analyzes the data. The present version of proposed software was developed in C# language with .Net Framework 3.5. Since C# is a modern object oriented language, its source code tend to be more universally used than other languages such as C and C++ languages. Hence the proposed software can be easily modified by including more open source or in house developed mathematical modeling. With .Net Framework, the proposed software is also easy to use via graphic user interface (GUI).

## 1.2 Thesis objective and scope

The aim of this study is to develop the prototype software for visualization and analysis nucleotide sequence of Dengue virus.

The objectives needed to be achieved for this thesis are:

(1) To design the optimal representation for nucleotide sequence of Dengue virus.

(2) To develop the CADViST software which can perform three kinds of functions, i.e. (a) nucleotide sequence search engine, (b) assigning serotype to the unknown Dengue virus sequence and (c) displaying BLAST search results.

## 1.3 Thesis Organization

This thesis begins with the background on Dengue virus. This chapter also reviews the existing visualization tools for biological sequence analysis and discusses advantages and disadvantages of each tool. Chapter 3 describes the methodology for developing the proposed software called CADViST. Results and discussion are described in Chapter 4 and 5, respectively. Finally, Chapter 6 concludes this thesis and indicates future research direction.

# CHAPTER II
# LITERATURE REVIEW

The literature review is divided into five sections. Section 1 provides background on Dengue virus. Section 2 illustrates the existing visualization tools for biological sequence analysis. Section 3 provides the idea of sequence comparison. Section 4 explains BLAST algorithm for which *CADViST* uses as a supplementary tool to perform sequence database search. Section 5 describes the multiple sequence alignment with Clustal programs used as part of *CADViST* development process.

## 2.1 Background of Dengue virus

### 2.1.1 Biological characteristics of Dengue virus

Dengue virus is a member of the Flavivirus genus within the family Flaviridae. There are four distinct serotypes of Dengue virus, DENV-1 through DENV-4. The transmission of Dengue virus to humans occurs via the bite of Aedes mosquitoes [2].

### 2.1.2 Clinical presentation of Dengue virus infection

Dengue virus infection has three main clinical criteria: (1) Undifferentiated Fever (UF), (2) Dengue Fever (DF) and (3) Dengue Haemorrhagic Fever (DHF). According to WHO case definition, clinical characteristics of DHF are (1) high fever, (2) haemorrhagic tendency, (3) thrombocytopenia and (4) Haemoconcentration equal or greater than 20% relative to baseline or evidence of plasma leakage. Some patients, the temperature may be higher than 40 ℃ for 2 to 7 days. During the acute stage of illness it is not possible to differentiate DF from DHF. The hallmark of DHF that differentiates it from DF is plasma leakage as a result of increased vascular permeability and abnormal hemostasis. The patient suffering from Dengue may develop shock, called Dengue Shock Syndrome (DSS). For defining DSS all of above four criteria of DHF and additionally, evidence of circulatory failure by rapid and weak pulse, narrow pulse pressure (less than 20 mmHg) and fall of blood pressure, abdominal pain and hypotension. Patients, who progress to DSS, get into critical condition usually at the time or soon after the fall of temperature following the symptoms described in DHF and the additional abdominal pain. Shock occurs two to six days after the start of symptoms. During shock the pulse and blood pressure become undetectable. If the patients are not treated immediately they may die within 12-24 hours [2].

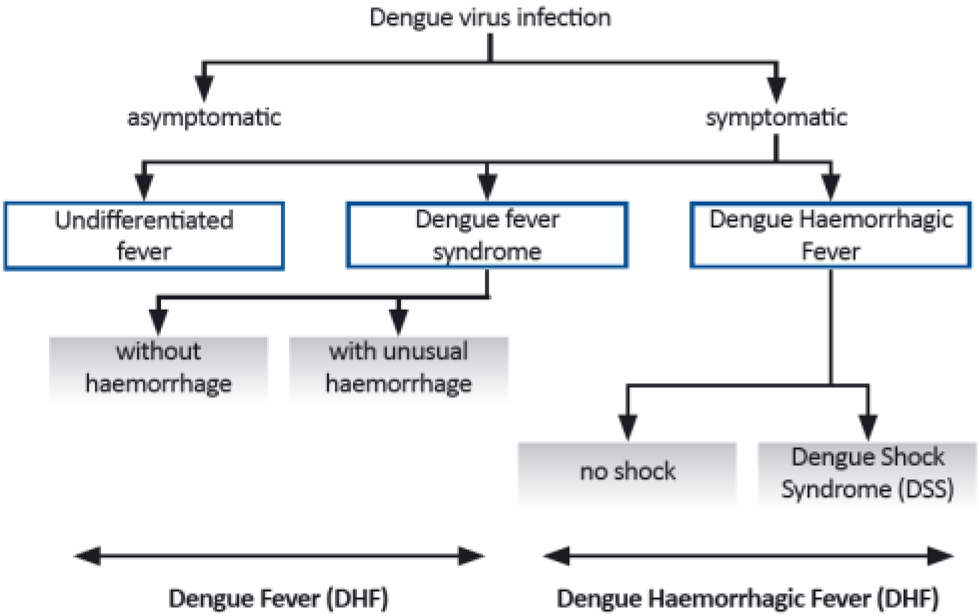


**Figure 2.1** Clinical spectrum of Dengue infection [3]

### 2.1.3 Dengue in the Western Pacific Region

Dengue virus infection has become a major problem in public health. Dengue information in Western Pacific Regional Office (WPRO), there is 213,190 Dengue cases and 671 Dengue deaths reported in 27 out of 37 countries and territories in the region. Countries that were hardest hit by Dengue outbreaks in 2008 include Viet Nam, Malaysia and Philippines. Dengue outbreaks appear to be more frequent not only in endemic countries of the Asia sub-region of the Western Pacific region but also in countries of Pacific sub-region where it was often known to have contracted off-island. Pacific island countries reported experiencing Dengue outbreaks in 2008. Many of them reported high incidence rates of Dengue cases. These include Palau, American, Samoa, Kiribati, New Caledonia, Samoa and Fiji [3].

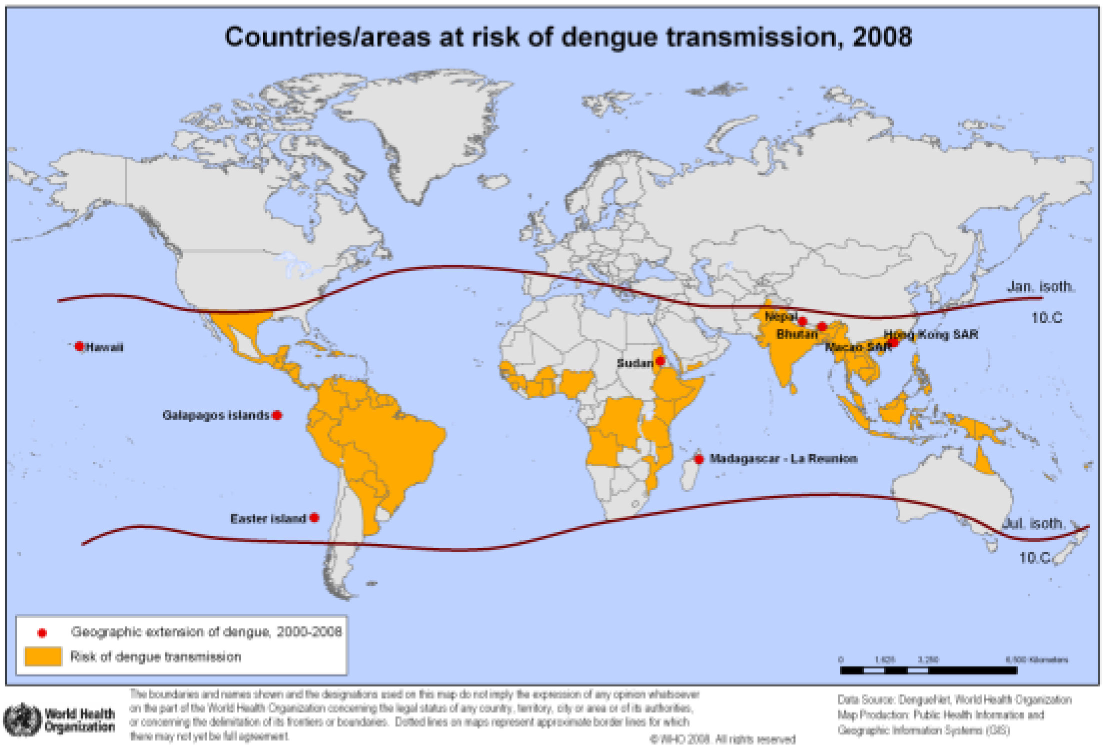

Countries/areas at risk of dengue transmission, 2008
Jan. isoth.
10.C
Nepal
Bhutan
Hong Kong SAR
Macao SAR
Hawaii
Sudan
Galapagos islands
Madagascar - La Reunion
Easter island
Jul. isoth.
10.C
● Geographic extension of dengue, 2000-2008
🟧 Risk of dengue transmission
0 1,625 3,250 6,500 Kilometers
World Health Organization
The boundaries and names shown and the designations used on this map do not imply the expression of any opinion whatsoever on the part of the World Health Organization concerning the legal status of any country, territory, city or area or of its authorities, or concerning the delimitation of its frontiers or boundaries. Dotted lines on maps represent approximate border lines for which there may not yet be full agreement.
© WHO 2008. All rights reserved
Data Source: DengueNet, World Health Organization
Map Production: Public Health Information and Geographic Information Systems (GIS)

**Figure 2.2** Distribution of Dengue [4]

### 2.1.4 Structure and function

Dengue virus genome is a single stranded RNA approximately 10.7 kb. According to Fig. 2.3, it can be divided into (1) three structural proteins: C, prM/M and E; (2) seven non-structural proteins: NS1, NS2A, NS2B, NS3, NS4A, NS4B and NS5; and (3) short noncoding regions on both the 5’ and 3’ end [2], [5].

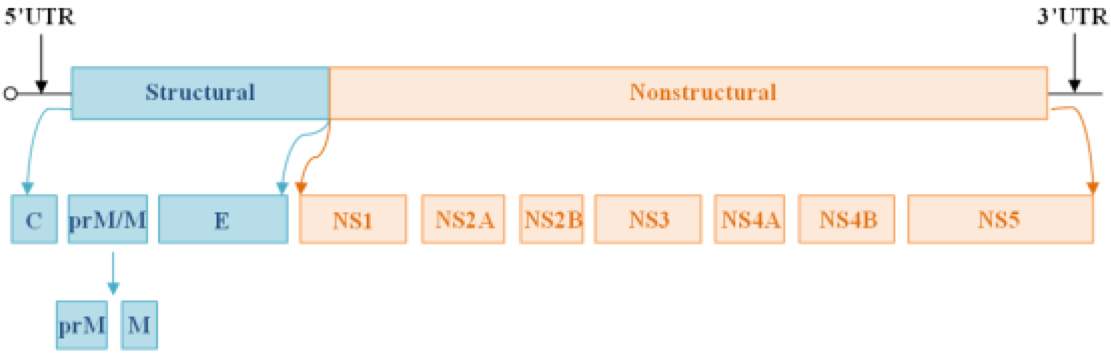


**Figure 2.3** The protein position of Dengue virus sequence [5]

#### 2.1.4.1 Structural protein

Three structural protein of Dengue virus consists of C, prM/M and E protein. (1) Capsid (C) protein of Dengue virus is a small structural protein which contains 100 amino acids. C protein is synthesized as the first protein. This protein is produced in cytoplasm. The peripheral N-terminal of C protein is hydrophobic amino acids. Hence, this protein is permeable through lipid composition in endoplasmic reticulum membrane. This protein is a component of the nucleocapsid. Some part is transported into nucleus. Its function is unknown. (2) Envelope (E) protein of Dengue virus consists of 495 amino acids. E protein is a major protein on the membrane surface of flavivirus. The function of E protein is to control the receptor binding and fusion with the host cell. (3) Dengue virus pre-membrane (prM) glycoprotein consists of 166 amino acids. A major function of this protein is to prevent E protein from undergoing acid-catalyzed fusion. Both prM and E proteins synthesized at rough endoplasmic reticulum will bind with each other to from heterodimers. The interaction of prM and E proteins contributes to the outer viral membrane surface. Dengue virus membrane (M) protein is cleavage product of prM protein which consists of 75 amino acids. It locates on the surfaces of extracellular virions. Its role is unknown [2], [5].

#### 2.1.4.2 Non-Structural protein

Seven non-structural proteins consist of NS1 to NS5 proteins. (1) The non-structural 1 (NS1) protein of which glycoprotein consists of 352 amino acids is essential for viral viability. It involves in recognition of a host immune response in order to against a viral infection. It also plays an essential role in viral RNA replication. (2) The non-structural 2A (NS2A) protein is an integral membrane protein of which hydrophobic consists of 218 amino acids. NS2A protein is the product of cleavage at the NS1/2A and NS2/2B proteins. It is involved in RNA replication but its precise biological function is unknown. (3) The non-structural 4B (NS2B) protein consists of 130 amino acids and also locates at membrane due to a hydrophobic part of the peripheral N- and C-terminal of NS2B protein. Moreover, NS2B protein is a cofactor of NS3 protein. The non-structural 3 (NS3) proteins protease, which consists of 618 amino acids.(4) This protein also acts as multifunctional enzymes such as serine protease, RNA helicase, RNA-stimulated nucleoside 5' – triphosphatase (NIPase) and RNA 5' – triphosphatase (RTPase). (5) The non-structural 4A (NS4A) protein is an integral membrane protein that consists of 150 amino acids: the peripheral N-terminal is hydrophilic and C-terminal is hydrophobic. The non-structural 4B (NS4B) protein is an integral membrane protein that consists of 248 amino acids. The combination of NS4A, NS4B and NS4A can inhibit interferon signaling. (6) It involves in RNA replication but its precise biological function is unknown. (7) The non-structural 5 (NS5) protein, consists of 900 amino acids, is the largest of the ten flavivirus proteins. It acts as bifunctional enzymes, which are 2' -O-methyltransferase (MTase), transfers methyl at the 5'-cap structure, and RNA-dependent RNA polymerase (RdRP) enzyme, RNA synthesis [2], [5].

#### 2.1.4.3 Different nucleotide sequence in Dengue virus

Dengue virus consists of a single stranded RNA genome of approximately 11 kilobases in length. The genome lengths of all four serotypes are different. Dengue virus 2 has maximum length of nucleotide sequence (including 10,723 nucleotide sequences). Dengue virus 4 has minimum length of nucleotide sequence (including 10,644 nucleotide sequences). Each serotype of Dengue virus has 68 - 76 % of identical nucleotide sequence. An identification serotype of Dengue virus process bases on Envelop glycoprotein, Membrane protein and Nonstructural Protein NS1 genes [2].

## 2.2 Existing visualization tools for biological sequence analysis

Genomic information is represented by character strings, in which each character is representing by an alphabet. For DNA sequence, it consists of the alphabets A, T, C and G; represented adenine, thymine, cytosine and guanine, respectively. The idea of visualization tool for genomic sequence is to map a genomic symbolic sequence into a graphical form. Each tool is developed based on some specific tasks which can be categorized into three approaches, i.e. base vector, sequential, and Fourier transform (FT) approaches.

### 2.2.1 Base vector approach

The method represents four kinds of bases by four unit vectors. In the literature reviews, many authors have proposed a variety pattern of the vectors representing the nucleotides in the sequence. Existing examples mainly include two and three dimensional graphical representations.

#### 2.2.1.1 Three dimensional graphical representations

The graphical approach representing DNA sequences was first reported in 1982 by Eugene Hamori and John Ruskin at Tulane University in Louisiana, called H Curve. In this method, the information content of a nucleotide sequence is mapped into a three dimensional space curves [6]. The unit vectors representing four nucleotides A, G, C and T are shown in Fig. 2.4.

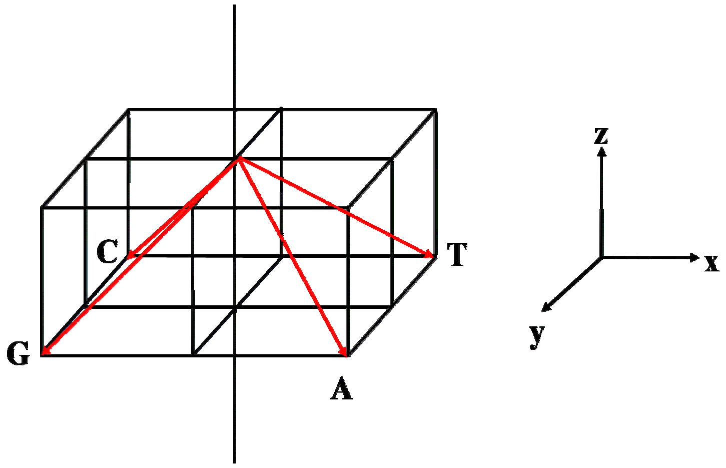


**Figure 2.4** The unit vectors designed by Eugene Hamori and John Ruskin of the four DNA nucleotides A, G, T and C [6]

In 1991, Zhang CT and Zhang R proposed graphical representation and numerical characterization for DNA sequences, called Z curve. The method maps the bases of DNA sequences and plots in three dimensional spaces. Z curve comprises of three components, $x_n$, $y_n$, $z_n$, that can be completely described the DNA sequence being studied by the three independent distributions. The component $x$ displays the distribution of purine/pyrimidine (R/Y) bases along the DNA sequence. The component $y$ displays the distribution of amino/keto (M/K) bases along the DNA sequence. The component $z$ displays the distribution of strong H bond / weak H bond (S/W) bases along the DNA sequence. Z curve transforms DNA sequence into a series of nodes $P_0$, $P_1$, $P_2$, ..., $P_N$ whose coordinates $x_n$, $y_n$, $z_n$ ($n$ = 0, 1, 2, …, N) where $N$ is the length of DNA sequence being studied, where $A_n, C_n, G_n$ and $T_n$ are the cumulative occurrence number of A, C, G and T, respectively, in the subsequence from 1st base to the $n$-th base in the sequence being studied by defined $A_0 = C_0 = G_0 = T_0 = 0$

$$x_n = (A_n + G_n) - (C_n + T_n)$$

$$y_n = (A_n + C_n) - (G_n + T_n)$$

$$z_n = (A_n + T_n) - (C_n + G_n)$$

According to the above equation, the subsequence constituted from 1-st base to the $n$-th base in the sequence being studied, when the numbers of purine bases (A and G) are greater than pyrimidine base (C and T), $x_n > 0$, otherwise, $x_n < 0$, and when the numbers of purine and pyrimidine bases are identical, $x_n = 0$. Similarly, when amino bases (A and C) are greater than keto bases (G and T), $y_n > 0$, otherwise, $y_n < 0$, and when the numbers of amino and keto bases are identical, $y_n = 0$. Also, when weak H bond bases (A and T) are greater than strong H bond bases (G and C), $z_n > 0$, otherwise, $z_n < 0$, and when the numbers of weak and strong H bond bases are identical, $z_n = 0$ [7], [8], [9]. James J Cai and his coworkers developed this tool in the MATLAB package, called MBEToolbox [10].

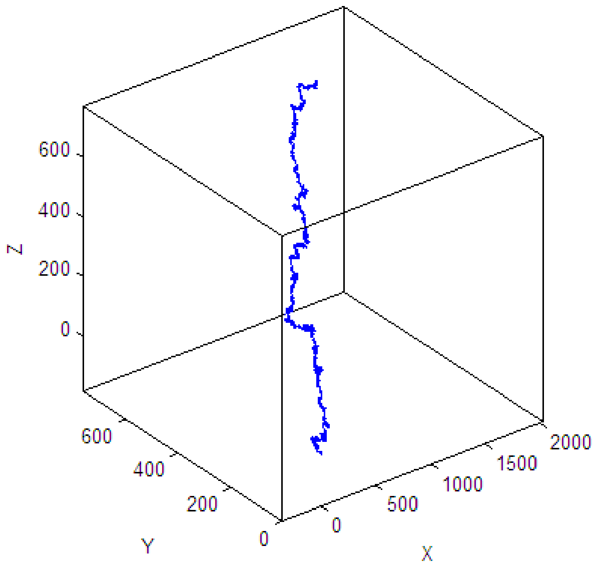


**Figure 2.5** The three dimension Z curves for AF180818 Dengue virus type 1, complete genome

### 2.2.1.2 Two dimensional graphical representations

The method is based on choosing the cardinal four directions in (*x*, *y*) coordinate system to represent the four bases in DNA sequences. The two dimensional graphical representation was first proposed by Gates in 1986, and then rediscovered independently by Nandy and Leong and Morganthaler [11]. In the Gates axes system plot cytosine (C) with the positive *x* direction, thymine (T) with the positive *y* direction, guanine (G) with the negative *x* direction and adenosine (A) with the negative *y* direction [11], [12]. The Nandy axes system associates G with the positive *x* direction, C with the positive y direction, A with negative *x* direction and T with negative *y* direction [11], [13]. In the Leong and Morgenthaler axes system, A is associated with positive *x* direction, T with positive *y* direction, C with negative *x* direction, and G with negative *y* direction [11]. However, all of these methods have their own weakness.

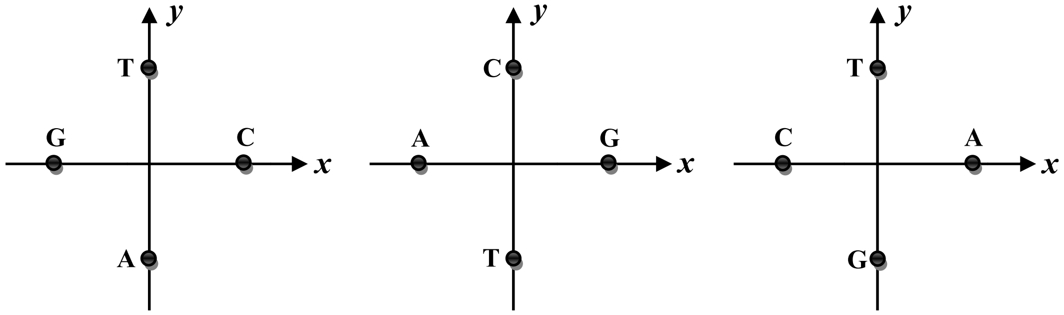
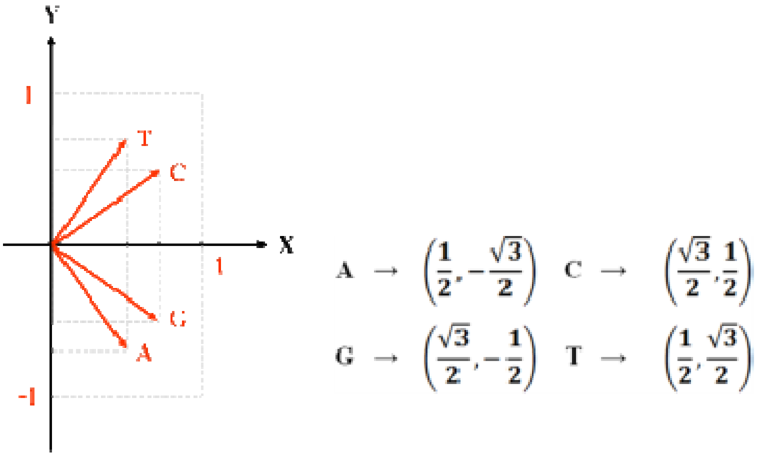


**Figure 2.6** The unit vectors presented by (a) Gates, (b) Nandy and (c) Leong and Morganthaler in the Cartesian coordinate plane [11]

For instant, according to Gates' graphical representation, the sequences AGTCA, AGTCAG, *etc*. have the same graphical representation. In 2003, Stephen S. -T Yau and his coworkers solve the problem by modified Gates' method. They constructed a pyrimidine-purine graph on two quadrants of the Cartesian coordinate system, with pyrimidines (T and C) in the first quadrant and purines (A and G) in the fourth quadrant [14]. The unit vectors representing four nucleotides A, G, C and T are shown in Fig. 2.7.



$$A \rightarrow \left(\frac{1}{2}, -\frac{\sqrt{3}}{2}\right) \quad C \rightarrow \left(\frac{\sqrt{3}}{2}, \frac{1}{2}\right)$$

$$G \rightarrow \left(\frac{\sqrt{3}}{2}, -\frac{1}{2}\right) \quad T \rightarrow \left(\frac{1}{2}, \frac{\sqrt{3}}{2}\right)$$

**Figure 2.7** The unit vectors presented by Yau in the Cartesian coordinate plane [14]

### 2.2.2 Sequential approach

The method use to identify similar base pair by pair. In bioinformatics, the most commonly used algorithms for database similarity searching are ***B***asic ***L***ocal ***A***lignment ***S***earch ***T***ool, or BLAST. BLAST was first reported by Stephen Altschul and coworker in 1990. The method represents the result of database similarity searching by alignment between a query sequence and sequences in the database being searched. BLAST uses heuristic algorithm for performing local alignments searches to find high scoring sequence alignments between query sequence and sequences in the database. Using heuristic algorithm, BLAST finds homologous sequences by locating short matches between two sequences, not comparing either sequence in its entirety. Each initial match is called “seed”. When a match is found, the search process is suspended, and BLAST tries to extend initial seed match in both directions [15].

### 2.2.3 FT Approach

The method applies Fourier Transforms (FT) to view sequence of nucleotides in frequency domain. Fourier analysis of genomic sequences, the major signal in coding regions of genomic sequences is the three-base periodicity [16], [17]. In 2001, Dimitris Anastassiou applies Fourier Transform to convert four symbolic DNA sequence consisting of letters A, T, C and G into a visual representation that highlights periodicities of co-occurrence of DNA patterns. The spectrogram of biomolecular sequences that provide local frequency information for all four bases is defined by using three primary colors: red, green and blue. Even though this method yields an impressive graphical representation, the computational complexity is fairly high [17].

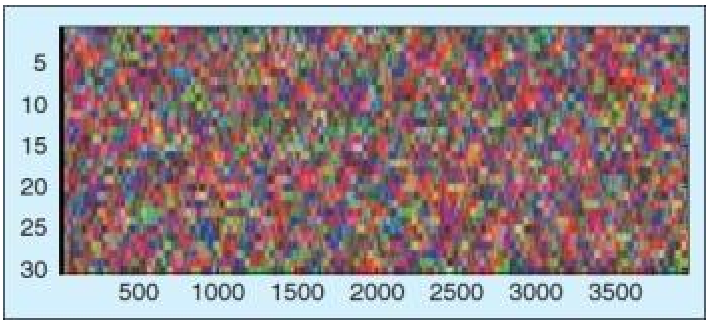


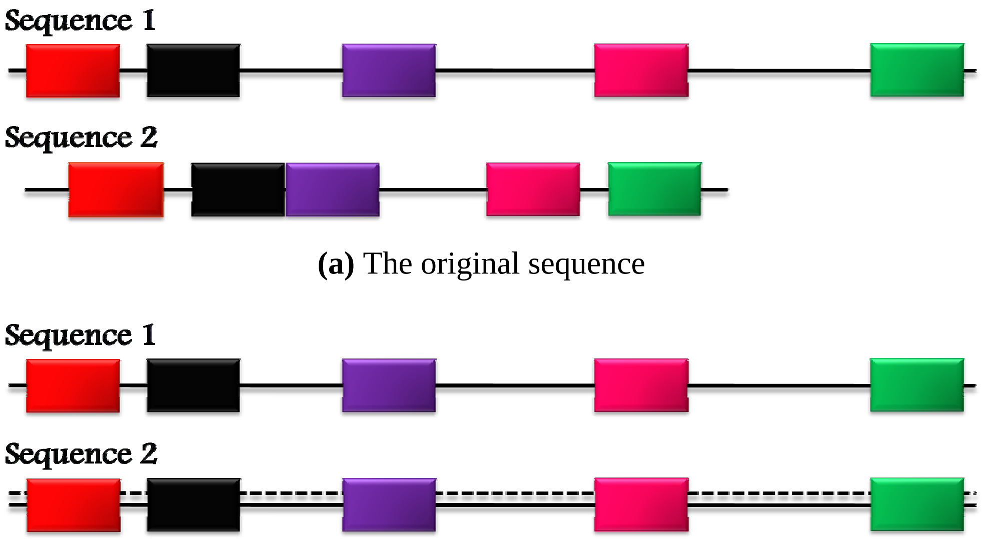
**Figure 2.8** Color spectrogram of DNA stretch [17]

## 2.3 Sequence comparison

Sequence comparison can be defined as the problem of finding “which parts of the sequences are similar”, and “which parts are different?”. The algorithms related to this area are: (1) the algorithms for comparing two sequences and (2) multiple sequence alignment.

### 2.3.1 Algorithms for comparing two sequences

The idea of aligning two sequences of possibly different sizes is to break them into smaller pieces by inserting spaces (-) so that identical subsequences are eventually aligned in a one-to-one correspondence. At last, the sequences end up with the same size (see Figs. 2.9) [18].



**(a)** The original sequence

**(b)** Aligned results

**Figure 2.9** The idea of aligning two sequences [18]

This work concentrates on efficient algorithm for comparing query sequence with all database sequences. BLAST is fast search algorithm that is widely used to find homologous sequence by comparing a query sequence with all the sequences in database. BLAST services can be accessed through a web service at the NCBI. User is also able to run BLAST search on their local computer and built into other tools. The main advantage of standalone BLAST is to be able to create your own search database. By its merits, the first version of *CADViST* employs standalone

BLAST+ application to find database sequences related to the query. BLAST algorithm is described in more detail in section 2.4.

### 2.3.2 Algorithms for multiple sequences alignment

Multiple sequence alignment is a comparison of a group of related sequences. The method is used to arrange regions of distributed similarity over the data sets of interest. Generally, a group belong to the same organism should have similar sequences. Applying multiple sequence alignment techniques is the way of arranging the biological sequence to identify the region of similarity over sets of sequence being studied. Multiple sequence alignment can be performed automatically using a variety of existing programs. Among many techniques, Clustal programs are one of the most widely used programs for carrying out automatic multiple alignment sets of nucleotide sequences [19].

In this thesis, *CADViST* development process employs multiple sequence alignment with ClustalW program to find one representative sequence of each Dengue virus serotype. The ClustalW algorithm is described in more detail in section 2.5.

## 2.4 BLAST algorithm

Nowadays, a number of software tools are available such as the BLAST which is provided by National Center for Biotechnology Information. This tool is used for representing the result of genome alignment positions. Within the alignment, there are three possible outcomes for each genome pair in an alignment: match, mismatch or gap. Matches represent unchanged in any alignment. Mismatches indicate that a substitution has occurred in at least one of alignment. Gaps (-) indicate that an insertion occurred in one of the sequences [20]. The following example illustrates an alignment between the sequences S1 = ACAAGACAGCGT and S2 = AGAACAAG GCGT. It should be noted that, in this thesis, we employ the software BLASTN which is another version of BLAST that is used for searching the nucleotide sequence.

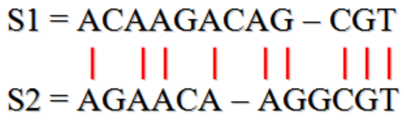


**Figure 2.10** Alignment of two sequences

BLAST identifies homologous sequence using heuristic algorithm which initially finds short matches between two sequences. Thus, the method does not take the entire sequence. After initial match, BLAST attempts to start local alignments from these initial matches. This means that BLAST does not guarantee the optimal alignment, it may miss matches. According to this strategy, it is expected to find most matches, but sacrifices complete sensitivity in order to increase speed. BLAST works by breaking a query sequence into a series of word. These words are compared to words from sequences in the database. Once a match is found, the sequence are aligned and scored. The basic BLAST procedure is illustrated in following step (see Fig. 2.12) [21], [22].

**Step1:** Filter query sequence to remove low complexity regions.

The regions with low complexity sequence are unusual composition that can create problems in sequence similarity searching. Normally, the low complexity region can be recognized by visual inspection. For instance, the nucleotide sequence AAATAAAAAAAATAAAAAAT has low complexity. Filtering can eliminate these poor matches. For nucleotide queries, DustMasker program is an application used to filter low complexity part [23].

**Step2:** Find all query and target matches as follow:

**(A)** Extract a word from the query, using a sliding window,

For nucleotide searches, the default number of query word per window length (word size) is 11 and can be raised (word size =15) or reduced (word size = 7). When a query is used to search a database, the first step in BLAST algorithm is to divide the query into a series of a smaller sequence (words) of a particular length (word size). It should be noted that lower word size yields more accurate but slower search.

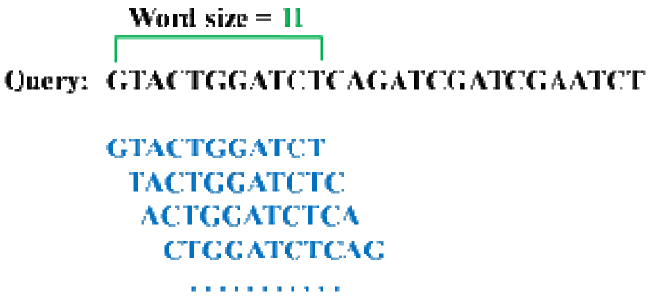


**Figure 2.11** The default word length for DNA/RNA comparisons [24]

**(B)** Find an exact match of the word in the target sequence (if no match: return to Step2A, if a match: continue),

**(C)** Extend match in both directions,

**(D)** Calculate an E-value for the final match,

**(E)** Save the matches whose E-value is better than given threshold,

In BLAST search, E-value or expectation value is parameter that describes quality of alignment expected to be found, assuming that the database sequences are random. An E-value of 10 means that if you scramble your sequences up and search again in a random set of sequences, you would expect to be found as good as the one you are trying to interpret about ten times. The default threshold for E-value is 10. For further information on calculating E-value, please see [25], [26].

**F:** Repeat step 2A-2E until all possible query words have been tried.

**Step4:** Rank the matches by their E-values.

The search results are described by one line summaries called “Descriptions”. The description lines are sorted by increasing E-value, so the most significant alignments (lowest E-value) are at the top.

**Step5:** Print out the top matches.

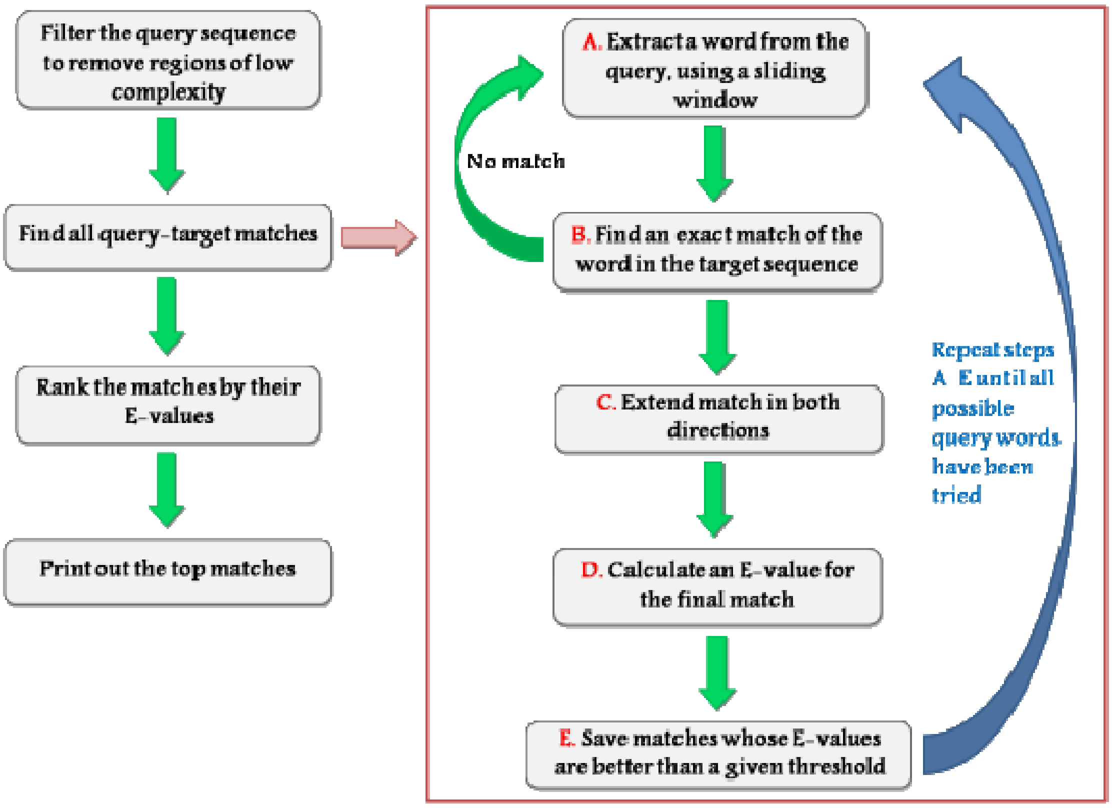
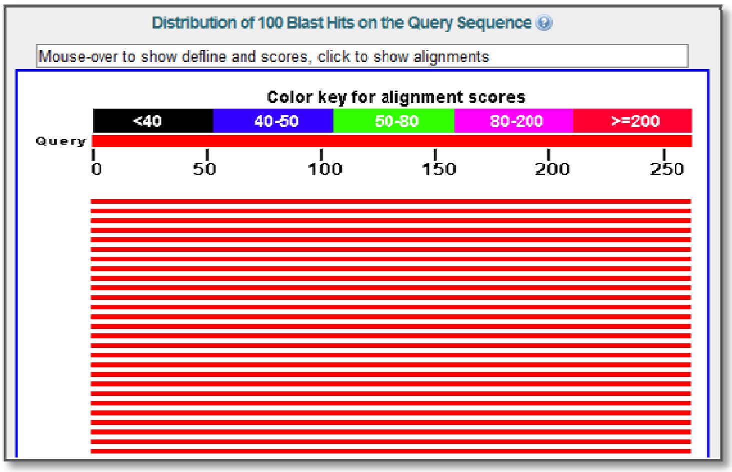

Filter the query sequence to remove regions of low complexity
Find all query-target matches
Rank the matches by their E-values
Print out the top matches
A. Extract a word from the query, using a sliding window
No match
B. Find an exact match of the word in the target sequence
C. Extend match in both directions
D. Calculate an E-value for the final match
E. Save matches whose E-values are better than a given threshold
Repeat steps A E until all possible query words have been tried

**Figure 2.12** The step of BLAST Algorithm [25], [26]

The default output of BLAST search, in which most users are familiar with the so-called "tradition BLAST report", consists of three major sections: (1) a graphical overview of the results (see Fig 2.13), (2) a series of one-line descriptions of each database sequence found to match the query sequence (see Fig 2.14) and (3) the alignment for each database sequence matched (see Fig 2.15) [20].

### 2.4.1 Graphical overview of the results

Fig. 2.13 shows graphical overview of BLAST results. Thick red numbered bar at the top of the figure represents the query sequence. The colored bars below the query sequence represent the matched database sequences. The textbox is for displaying the information on matching database sequences, when moving mouse over the colored bars, the definition line for that sequence is shown in this textbox [20], [27].



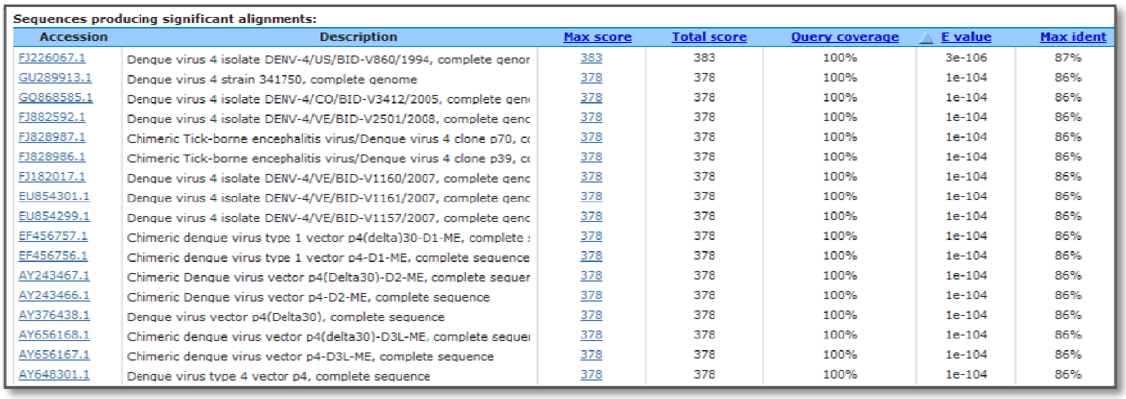
Distribution of 100 Blast Hits on the Query Sequence

Mouse-over to show defline and scores, click to show alignments

**Figure 2.13** NCBI BLAST output form: graphical overview of the results [28]

### 2.4.2 Best match listing

This section provides details on the matched database sequences in a table. Each line composes of seven fields. The first column indicates the name of sequence found in BLAST search. The second column indicates the definition of each database sequence. The third column indicates the highest alignment score from that database sequence. The fourth column indicates the total of alignment scores of all segments from that database sequence. The fifth column indicates the percent of query that has been matched by that database sequence. The sixth column indicates E-value, the measure of quality of match. The last column indicates the number of identical residues in the query and database sequences [20], [27].



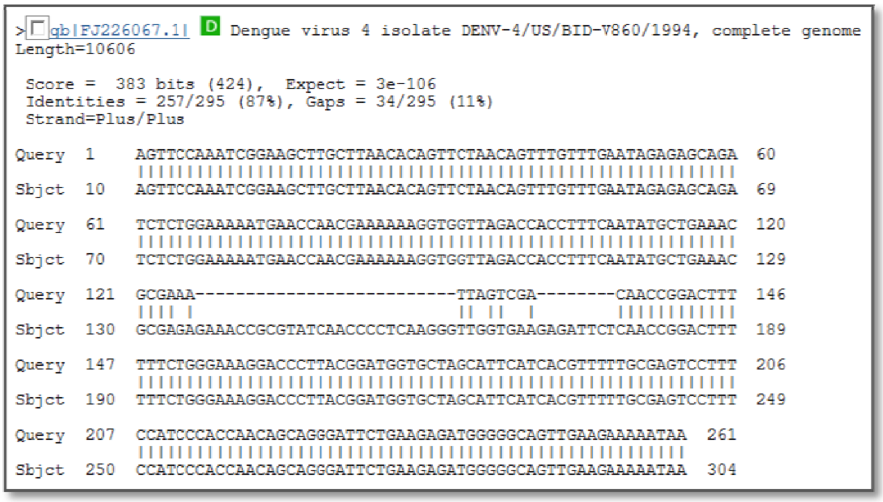
Sequences producing significant alignments:

| Accession | Description | Max score | Total score | Query coverage | E value | Max ident |
|---|---|---|---|---|---|---|
| FJ226067.1 | Dengue virus 4 isolate DENV-4/US/BID-V860/1994, complete genor | 383 | 383 | 100% | 3e-106 | 87% |
| GU289913.1 | Dengue virus 4 strain 341750, complete genome | 378 | 378 | 100% | 1e-104 | 86% |
| GQ868585.1 | Dengue virus 4 isolate DENV-4/CO/BID-V3412/2005, complete gen | 378 | 378 | 100% | 1e-104 | 86% |
| FJ882592.1 | Dengue virus 4 isolate DENV-4/VE/BID-V2501/2008, complete genc | 378 | 378 | 100% | 1e-104 | 86% |
| FJ828987.1 | Chimeric Tick-borne encephalitis virus/Dengue virus 4 clone p70, cc | 378 | 378 | 100% | 1e-104 | 86% |
| FJ828986.1 | Chimeric Tick-borne encephalitis virus/Dengue virus 4 clone p39, cc | 378 | 378 | 100% | 1e-104 | 86% |
| FJ182017.1 | Dengue virus 4 isolate DENV-4/VE/BID-V1160/2007, complete genc | 378 | 378 | 100% | 1e-104 | 86% |
| EU854301.1 | Dengue virus 4 isolate DENV-4/VE/BID-V1161/2007, complete genc | 378 | 378 | 100% | 1e-104 | 86% |
| EU854299.1 | Dengue virus 4 isolate DENV-4/VE/BID-V1157/2007, complete genc | 378 | 378 | 100% | 1e-104 | 86% |
| EF456757.1 | Chimeric dengue virus type 1 vector p4(delta)30-D1-ME, complete : | 378 | 378 | 100% | 1e-104 | 86% |
| EF456756.1 | Chimeric dengue virus type 1 vector p4-D1-ME, complete sequence | 378 | 378 | 100% | 1e-104 | 86% |
| AY243467.1 | Chimeric Dengue virus vector p4(Delta30)-D2-ME, complete sequer | 378 | 378 | 100% | 1e-104 | 86% |
| AY243466.1 | Chimeric Dengue virus vector p4-D2-ME, complete sequence | 378 | 378 | 100% | 1e-104 | 86% |
| AY376438.1 | Dengue virus vector p4(Delta30), complete sequence | 378 | 378 | 100% | 1e-104 | 86% |
| AY656168.1 | Chimeric dengue virus vector p4(delta30)-D3L-ME, complete seque | 378 | 378 | 100% | 1e-104 | 86% |
| AY656167.1 | Chimeric dengue virus vector p4-D3L-ME, complete sequence | 378 | 378 | 100% | 1e-104 | 86% |
| AY648301.1 | Dengue virus type 4 vector p4, complete sequence | 378 | 378 | 100% | 1e-104 | 86% |

**Figure 2. 14** NCBI BLAST output form: best match listing [28]

### 2.4.3 Local sequence alignment

This section provides the full definition of database sequences and the length of matched sequences. The following is the statistical measures of the match. Finally, the actual alignment is shown with the query on top row and the database match is labeled as “Sbjct”, in the following row [20], [27].



```
>□gb|FJ226067.1| D Dengue virus 4 isolate DENV-4/US/BID-V860/1994, complete genome
Length=10606

 Score =  383 bits (424),  Expect = 3e-106
 Identities = 257/295 (87%), Gaps = 34/295 (11%)
 Strand=Plus/Plus

Query  1    AGTTCCAAATCGGAAGCTTGCTTAACACAGTTCTAACAGTTTGTTTGAATAGAGAGCAGA  60
            ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  10   AGTTCCAAATCGGAAGCTTGCTTAACACAGTTCTAACAGTTTGTTTGAATAGAGAGCAGA  69

Query  61   TCTCTGGAAAAATGAACCAACGAAAAAAGGTGGTTAGACCACCTTTCAATATGCTGAAAC  120
            ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  70   TCTCTGGAAAAATGAACCAACGAAAAAAGGTGGTTAGACCACCTTTCAATATGCTGAAAC  129

Query  121  GCGAAA--------------------------TTAGTCGA--------CAACCGGACTTT  146
            |||| |                          || ||  |        ||||||||||||
Sbjct  130  GCGAGAGAAACCGCGTATCAACCCCTCAAGGGTTGGTGAAGAGATTCTCAACCGGACTTT  189

Query  147  TTTCTGGGAAAGGACCCTTACGGATGGTGCTAGCATTCATCACGTTTTTGCGAGTCCTTT  206
            ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  190  TTTCTGGGAAAGGACCCTTACGGATGGTGCTAGCATTCATCACGTTTTTGCGAGTCCTTT  249

Query  207  CCATCCCACCAACAGCAGGGATTCTGAAGAGATGGGGGCAGTTGAAGAAAAATAA  261
            |||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  250  CCATCCCACCAACAGCAGGGATTCTGAAGAGATGGGGGCAGTTGAAGAAAAATAA  304
```

**Figure 2.15** NCBI BLAST output form: local sequence alignment [28]

## 2.5 Multiple alignment in Clustal programs

According to Fig 2.16, the basic multiple alignment in Clustal programs consist of three main stages: (1) perform pairwise alignment on all sequences to calculate distance matrix, (2) use distance matrix to calculate guide tree, and (3) the sequences are progressively aligned using the branching order in the guide tree [19].

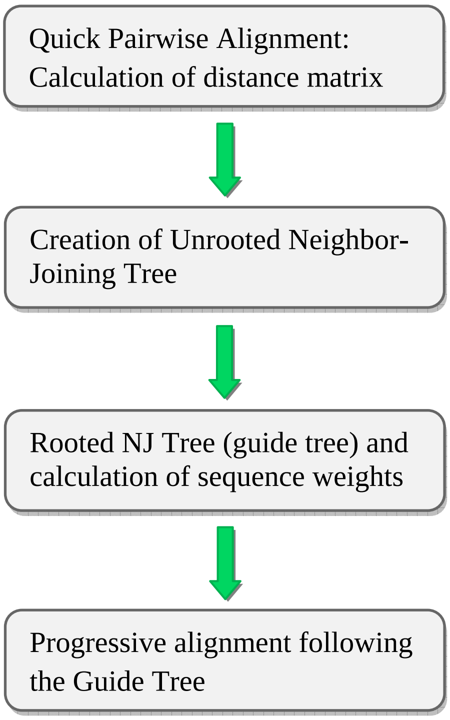

Quick Pairwise Alignment: Calculation of distance matrix
Creation of Unrooted Neighbor-Joining Tree
Rooted NJ Tree (guide tree) and calculation of sequence weights
Progressive alignment following the Guide Tree

**Figure 2.16** Overview of ClustalW procedure [19]

### 2.5.1 Calculating distance matrix

The procedure at this stage is to calculate the distance (percent identity) of each pair of sequences. The first stage is to calculate identity scores by comparing each sequence with each other. These scores are calculated from the number of match score divided by the number of position compared. Then, these score are converted into distance by dividing by 100 and subtracting from 1 to give number of differences per site. In the original Clustal programs, the assigned match score is 1 and mismatch score is 0. For instance, the distance matrix of sequences: S1 = ATCG, S2 = ATCT, S3 = AGGC and S4 = GCAA can be calculated as follows:

| | | | |
|---|---|---|---|
| **ATCG** | **ATCG**<br>\| \| \|<br>**ATCT** | Identity Score = 3/4 | Distance = 1-(0.75/100)<br>= 1-0.0075<br>= 0.9925 |
| | **ATCG**<br>\|<br>**AGGC** | Identity Score = 1/4 | Distance = 1-(0.25/100)<br>= 1-0.0025<br>= 0.9975 |
| | **ATCG**<br><br>**GCAA** | Identity Score = 0/4 | Distance = 1-(0/100)<br>= 1-0<br>= 1 |
| **ATCT** | **ATCT**<br>\|<br>**AGGC** | Identity Score = 1/4 | Distance = 1-(0.25/100)<br>= 1-0.0025<br>= 0.9975 |
| | **ATCT**<br><br>**GCAA** | Identity Score = 0/4 | Distance = 1-(0/100)<br>= 1-0<br>= 1 |
| **AGGC** | **AGGC**<br><br>**GCAA** | Identity Score = 0/4 | Distance = 1-(0/100)<br>= 1-0<br>= 1 |

**Table 2.1** The original distance matrix [29]

| | **S1** | **S2** | **S3** |
|---|---|---|---|
| **S2** | 0.9925 | | |
| **S3** | 0.9975 | 0.9975 | |
| **S4** | 1 | 1 | 1 |

### 2.5.2 Calculating guide tree

The procedure at this stage is to produce rooted tree used to guide the final multiple alignment process. The Neighbour-Jointing (NJ) algorithm (Saitou and Nei 1987) is used to group sequence. The idea of NJ is to produce a modification matrix that takes into account the net divergence rate of the entire tree, so nodes close together but far apart from all other nodes are not necessarily joined directly. The equation below is a model for calculating the modification matrix. In this equation $d$ is the original distance matrix. $R_i$ is the net divergence matrix for each sequence is calculated as the sum of each node to all other nodes. $N$ is the set of leaves in the tree. The number of leaves is $|N|$ [30], [31].

$$R_i = \frac{1}{|N|-2}\sum\nolimits_{m\in N} d_{im} \qquad \text{... (Equation. 2.1)}$$

$$D_{ij} = d_{ij} - (R_i + R_j) \qquad \text{... (Equation. 2.2)}$$

According to the original distance matrix, the resulting tree is an unrooted tree with branch lengths proportional to estimated divergence along each branch as shown in Fig.2.17.

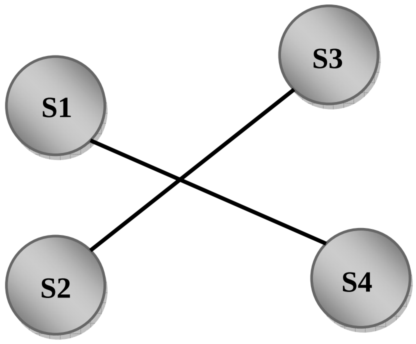


**Figure 2.17** The initial branch point [29]

In order to clearly explain the algorithm of the Neighbour Joining method for reconstructing rooted tree and computing the length of the branches of this tree, we use the information in section 2.5.1 as the example.

**Stage1:** Compute the Net divergence matrix R for each sequence by Eq. 2.1

$R_{S1}$ = (0.9925 + 0.9975 + 1) / (4 - 2) = 1.495

$R_{S2}$ = (0.9925 + 0.9975 + 1) / (4 - 2) = 1.495

$R_{S3}$ = (0.9975 + 0.9975 + 1) / (4 - 2) = 1.4975

$R_{S4}$ = (1+1+1) / (4 - 2) = 1.5

**Stage2:** Compute the modification matrix D by Eq. 2.2

$D_{S1S2} = d_{S1S2} – (R_{S1} + R_{S2})$

$D_{S1S2}$ = 0.9925 – (1.495 + 1.495)

$D_{S1S2}$ = -1.9975

$D_{S1S3} = d_{S1S3} (R_{S1} + R_{S3})$

$D_{S1S3}$ = 0.9975 – (1.495+1.4975)

$D_{S1S3}$ = -1.995

$D_{S1S4} = d_{S1S4} – (R_{S1} + R_{S4})$

$D_{S2S3} = d_{S2S3} – (R_{S2} + R_{S3})$

$D_{S2S3}$ = 0.9975 – (1.495+1.4975)

$D_{S2S3}$ = -1.995

$D_{S2S4} = d_{S2S4} – (R_{S2} + R_{S4})$

$D_{S2S4}$ = 1 – (1.495 + 1.5)

$D_{S2S4}$ = -1.995

$D_{S3S4} = d_{S3S4} – (R_{S3} + R_{S4})$

$D_{S1S4} = 1 - (1.495 + 1.5)$ $D_{S3S4} = 1 - (1.4975 + 1.5)$

$D_{S1S4} = -1.995$ $D_{S3S4} = -1.9975$

| | **S1** | **S2** | **S3** |
|---|---|---|---|
| **S2** | -1.9975 | | |
| **S3** | -1.995 | -1.995 | |
| **S4** | -1.995 | -1.995 | -1.9975 |

**Stage3:** According to the original distance matrix, joining S1 and S2 is one of the minima of the original distance matrix; joining S3 and S4 is also the other minimum. S1 and S2 are the closest. To construct a tree group them under a new node called X. Then compute the distances to the new node X. A new node X is the branches join S1, S2 and the rest of the tree, defined the lengths of the tree branch from X to S1 and S2 by Eq. 2.3 and 2.4, respectively. These distances are the lengths of the new branches. And the distance from X to each other node use Eq.2.5 to compute.

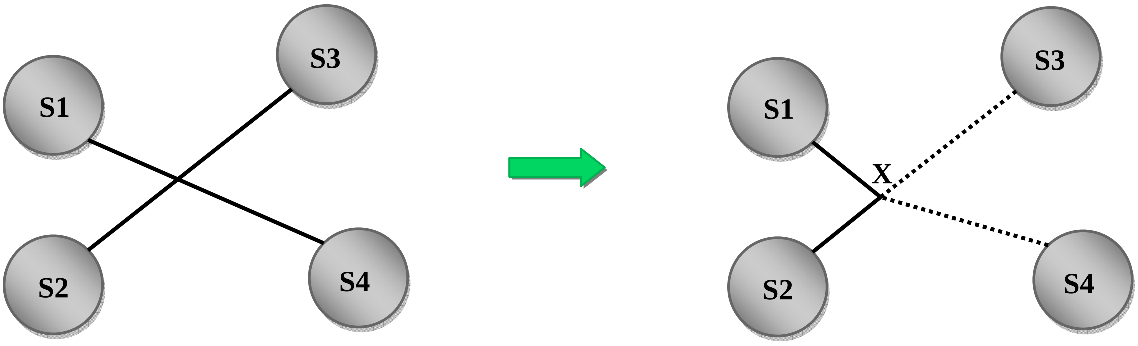


**Figure 2.18** Nodes joining S1 and S2

**Explanation of Equation 2.3 – 2.4:** Define a new node u whose three branches join *i*, *j* and the rest of the branch tree. Define the lengths of the tree branch from u to i and j as:

$$d_{iu} = \frac{d_{ij}}{2} + \left(\frac{R_i - R_j}{2(N-2)}\right) \qquad \text{... (Equation 2.3)}$$

$$d_{ju} = d_{ij} - d_{iu} \qquad \text{... (Equation 2.4)}$$

**Explanation of Equation 2.5:** Define the distance from u to each other node (k ≠ i or j) as:

$$d_{ku} = (d_{ik} + d_{jk} - d_{ij})/2 \quad \text{... (Equation 2.5)}$$

$d_{S1X} = (d_{S1S2} / 2) + ((R_{S1} - R_{S2}) / (2(N - 2))$
$= (0.9925 / 2) + ((1.495 - 1.495) / (2(4 - 2))$
$= 0.49625$

$d_{S2X} = d_{S1S2} - d_{S1X} = 0.9925 - 0.49625$
$= 0.49625$

$d_{S3X} = (d_{S1S3} + d_{S2S3} - d_{S1S2}) / 2 = (0.9975 + 0.9975 - 0.9925) / 2$
$= 0.50125$

$d_{S4X} = (d_{S1S4} + d_{S2S4} - d_{S1S2}) / 2 = (1 + 1 - 0.9925) / 2$
$= 0.50375$

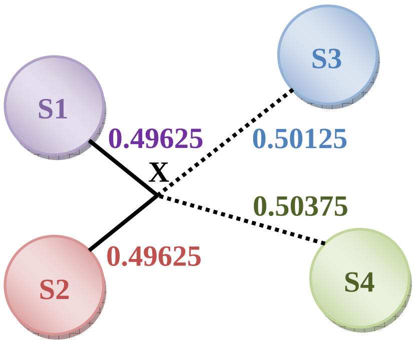


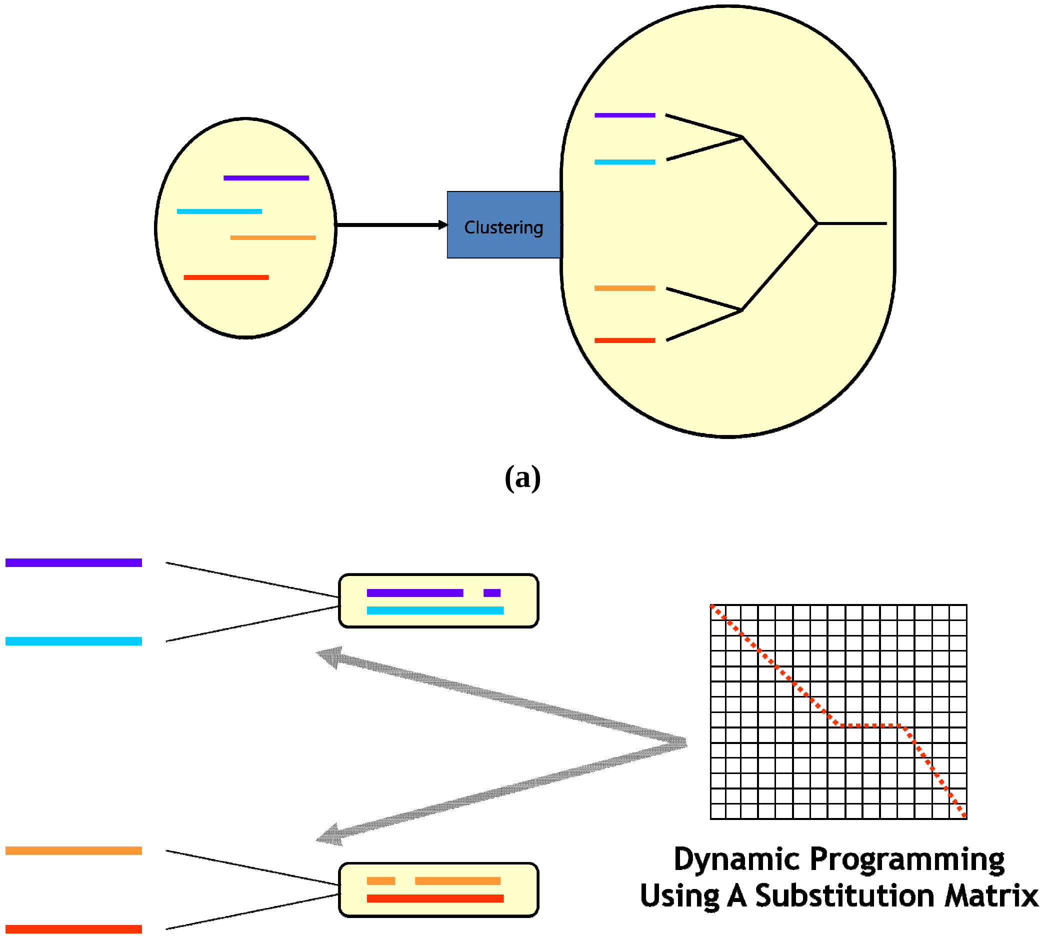
**Figure 2.19** A rooted tree with branch lengths

**Stage4:** According to Eq. 2.3 – 2.5, remove distances to nodes i and j from the original matrix, and increase N by 1.

**Stage5:** If more than two nodes remain, go back to stage1. Otherwise the tree is fully defined.

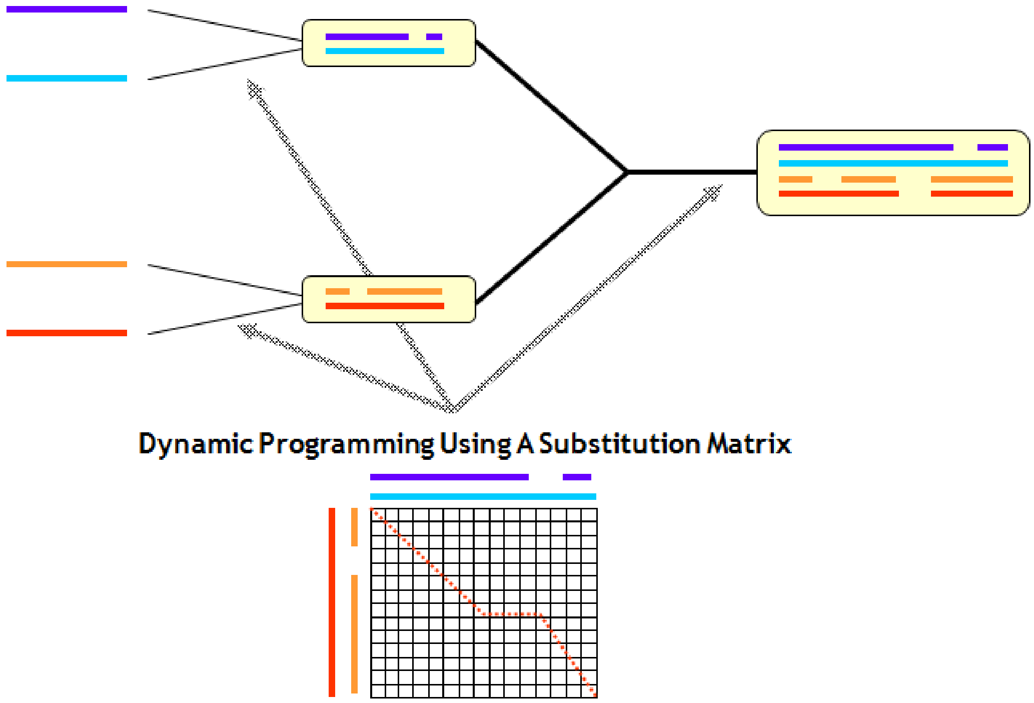
### 2.5.3 Progressive alignment

The procedure at this stage is to use a series of pairwise alignments to align larger and larger group of sequences according to branching order in the guide tree. Basically, most progressive alignment methods rely on dynamic programming to perform multiple alignment beginning with the most related sequences and then progressively adding less related sequences to the initial alignment [19], [32].



(a)

(b)



**(c)**

**Figure 2.20** Three stages of progressive sequence alignment: All progressive alignment requires three stages: **(a)** a first stage computes a distance value between each pair of input sequences to calculate distance matrix giving the divergence of each pair of each sequences; **(b)** a second stage calculates guide tree based on distance matrix; and **(c)** final stage built multiple sequence alignment (MSA) by adding the sequences sequentially to growing MSA according to the guide tree [19], [32].

### 2.5.4 Dynamic programming

Dynamic programming for pairwise alignment consists of three steps: (1) Initialization, (2) Matrix fill (scoring) and (3) Traceback (alignment) [32].

#### 2.5.4.1 Initialization step

The first step in dynamic programming algorithm is to create a matrix with ($M$ + 1) columns and ($N$ + 1) rows where $M$ and $N$ is the size of the aligned sequences. For instance, the two sequences to be globally aligned are S1 = GAATTCAGTTA and S2 = GGATCGA. Scoring sequences for alignment values are defined as: match score = 1, mismatch score = 0 and gap penalty for each insertion or deletion = 0. The score matrix D is initialized by placing each character of sequence

S1 above each column in the matrix, beginning with the second column, and placing each character of sequence S1 before each row, beginning with the second row [33].

| | | G | A | A | T | T | C | A | G | T | T | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | |
| G | | | | | | | | | | | | |
| G | | | | | | | | | | | | |
| A | | | | | | | | | | | | |
| T | | | | | | | | | | | | |
| C | | | | | | | | | | | | |
| G | | | | | | | | | | | | |
| A | | | | | | | | | | | | |

**Figure 2.21** Initialization of dynamic programming matrix [33]

#### 2.5.4.2 Matrix filling step

The filling step of the matrix element starts in the left top corner of the matrix that requires the score from the diagonal, vertical and horizontal neighbor cells. For each position in the matrix is the maximum global alignment score D(i, j) [33]. The value at position (i, j) is the maximum score at position (i, j). Let the scoring scheme D(i, j) be defined as follows:

**(1)** D(i-1, j-1) = D(i-1, j-1) + (Match or Mismatch score in the diagonal)

**(2)** D(i, j-1) = D(i, j-1) + (Gap score in sequence S1)

**(3)** D(i-1, j) = D(i-1, j) + (Gap score in sequence S2)

D(i, j) = Maximum of [D(i-1, j-1), D(i, j-1), D(i-1, j)]

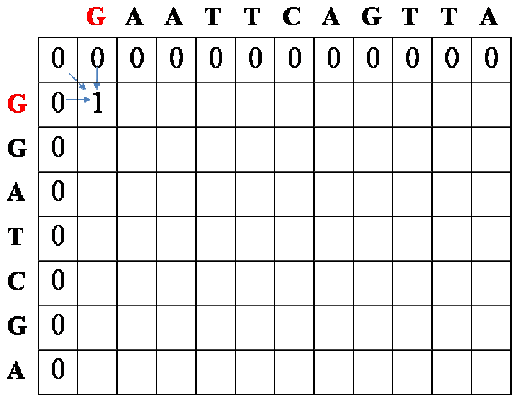
This example assumes that there is no gap opening or gap opening, or gap extension penalty. So the first row and the first column of the matrix can be initially filled with 0.

| | | G | A | A | T | T | C | A | G | T | T | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 0 | | | | | | | | | | | |
| G | 0 | | | | | | | | | | | |
| A | 0 | | | | | | | | | | | |
| T | 0 | | | | | | | | | | | |
| C | 0 | | | | | | | | | | | |
| G | 0 | | | | | | | | | | | |
| A | 0 | | | | | | | | | | | |

**Figure 2.22** Dynamic programming matrix with base case filled in [33]

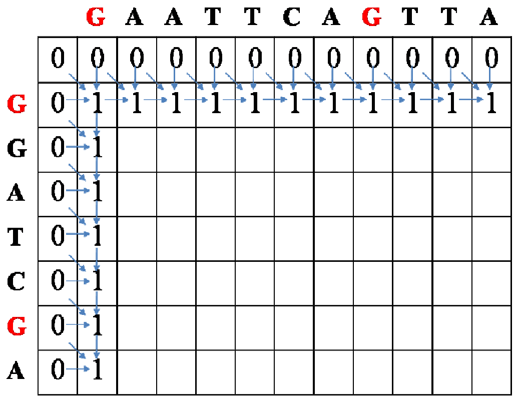
In position (1, 1) of the scoring matrix D(1, 1) is the maximum score of D(0, 0), D(1, 0) and D(0, 1).

$$D(1, 1) = \text{Max } [1, 0, 0] = 1$$



| | | G | A | A | T | T | C | A | G | T | T | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 0 | 1 | | | | | | | | | | |
| G | 0 | | | | | | | | | | | |
| A | 0 | | | | | | | | | | | |
| T | 0 | | | | | | | | | | | |
| C | 0 | | | | | | | | | | | |
| G | 0 | | | | | | | | | | | |
| A | 0 | | | | | | | | | | | |

**Figure 2.23** The first element of the matrix that is filled in D(1, 1) [33]



| | | G | A | A | T | T | C | A | G | T | T | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| G | 0 | 1 | | | | | | | | | | |
| A | 0 | 1 | | | | | | | | | | |
| T | 0 | 1 | | | | | | | | | | |
| C | 0 | 1 | | | | | | | | | | |
| G | 0 | 1 | | | | | | | | | | |
| A | 0 | 1 | | | | | | | | | | |

**Figure 2.24** Initialization of all elements along the second row second column [33]

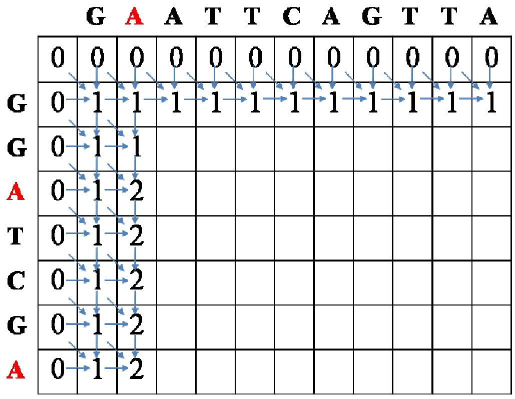
In column 2, the cell in row 2 will be assigned by the maximum value of 1 (mismatch score), 1 (horizontal gap score) or 1 (vertical gap score). So its value is 1. At the position column 2 row 3, the character in both sequences is A. So its value is the maximum score of 2 (match score), 1 (horizontal gap score) or 1 (vertical gap score), its value is 2.

$$D(2, 2) = \text{Max}[1, 1, 1] = 1 \text{ and } D(2, 2) = \text{Max}[2, 1, 1] = 2$$

In any row except the last one, moving along the position column 2 row 4, its value will be the maximum of 1 (mismatch score), 1 (horizontal gap score), or 2 (vertical gap score) so its value is 2. The value in the final row will be 2 since it is the maximum score of 2 (match score), 1 (horizontal gap score) or 2 (vertical gap score).

$$D(2, 4) = D(2, 5) = D(2, 6) = \text{Max}[1, 1, 2] = 2$$

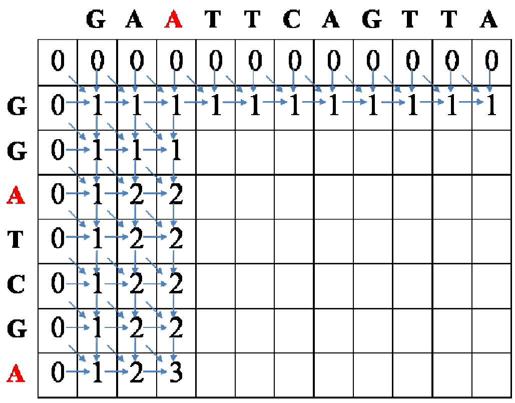
$$D(2, 7) = \text{Max}[2, 1, 2] = 2$$



| | | G | A | A | T | T | C | A | G | T | T | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| G | 0 | 1 | 1 | | | | | | | | | |
| A | 0 | 1 | 2 | | | | | | | | | |
| T | 0 | 1 | 2 | | | | | | | | | |
| C | 0 | 1 | 2 | | | | | | | | | |
| G | 0 | 1 | 2 | | | | | | | | | |
| A | 0 | 1 | 2 | | | | | | | | | |

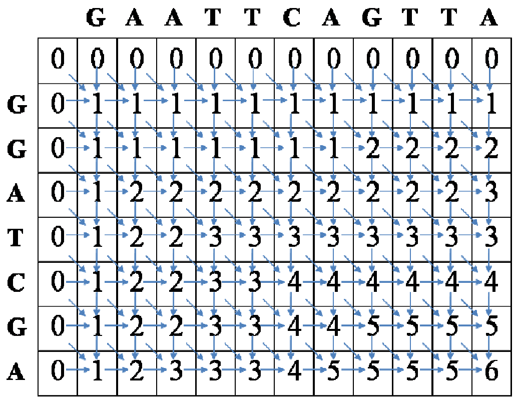
**Figure 2.25** Initialization of all elements along the third column [33]

Using the same techniques as described for column 3, Fig. 2.26 illustrates the values for column 3.



| | | G | A | A | T | T | C | A | G | T | T | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| G | 0 | 1 | 1 | 1 | | | | | | | | |
| A | 0 | 1 | 2 | 2 | | | | | | | | |
| T | 0 | 1 | 2 | 2 | | | | | | | | |
| C | 0 | 1 | 2 | 2 | | | | | | | | |
| G | 0 | 1 | 2 | 2 | | | | | | | | |
| A | 0 | 1 | 2 | 3 | | | | | | | | |

**Figure 2.26** Initialization of all elements along the forth column [33]



| | | G | A | A | T | T | C | A | G | T | T | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| G | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 |
| A | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 |
| T | 0 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| C | 0 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 |
| G | 0 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 5 | 5 | 5 | 5 |
| A | 0 | 1 | 2 | 3 | 3 | 3 | 4 | 5 | 5 | 5 | 5 | 6 |

**Figure 2.27** Complete score matrix with trace arrows [33]

### 2.5.4.3 Traceback step

Traceback step is the algorithm to determine the actual alignment. Traceback step start from the lower corner and trace back to the upper left. Each arrow represents one character at the end of each aligned sequence. (1) A horizontal move puts a gap in the left sequence. (2) A vertical move puts a gap in the top sequence. (3) A diagonal move uses one character from each sequence. The current alignment comprised of two conditions to be checked: (1) character comparison in both sequence and (2) the relationship between three neighbors of matrix cell (i, j): the neighbor above (i-1, j), diagonal neighbor (i-1, j-1) and neighbor left (i, j-1). If the first condition is true, disregard the second one. The current alignment is defined as follows:

**First condition:** If the position in the calculated alignment that matches with the same character, put "|" in the current alignment.

**Second condition:** Relationship between three neighbors of matrix cell (i, j):

**(1)** If the maximum score is D(i,j-1), put gap ("-") in left sequence.

**(2)** If the maximum score is D(i-1,j), put gap ("-") in top sequence.

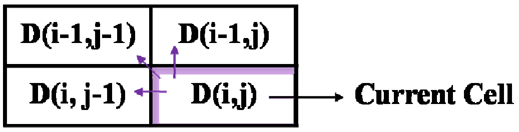
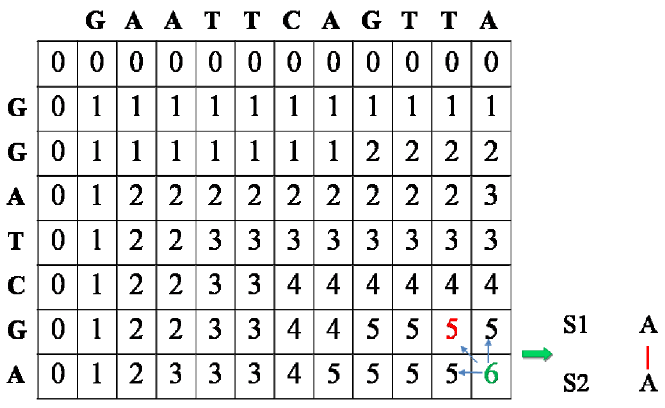


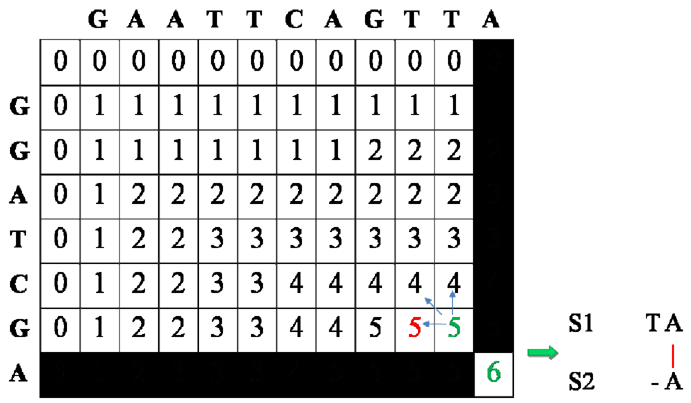
**Figure 2.28** Three neighbor cell at position (i, j)

After the matrix has been completely filled, the traceback step begins in the (M, N) position in the matrix. The current alignment is the result of three neighbors of matrix cell (M, N): the neighbor above (M-1, N), diagonal neighbor (M-1, N-1) and neighbor left (M, N-1). According to Fig. 2.29 (a), the value of current cell is 6. So the possible result is the diagonal match neighbor. Then decrease the position of current cell (i, j) by 1. Now the value of current cell is 5. According to the alignment as described in the above step, so the current alignment produces a gap to the left sequence. Following the traceback step until the position of (i, j) is (1, 1); that means the traceback is completed. The solution of possible alignment is shown in Figs. 2.29 [33].



| | | G | A | A | T | T | C | A | G | T | T | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| G | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| G | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 |
| A | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 |
| T | 0 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| C | 0 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 |
| G | 0 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 5 | 5 | 5 | 5 |
| A | 0 | 1 | 2 | 3 | 3 | 3 | 4 | 5 | 5 | 5 | 5 | 6 |

**(a)**

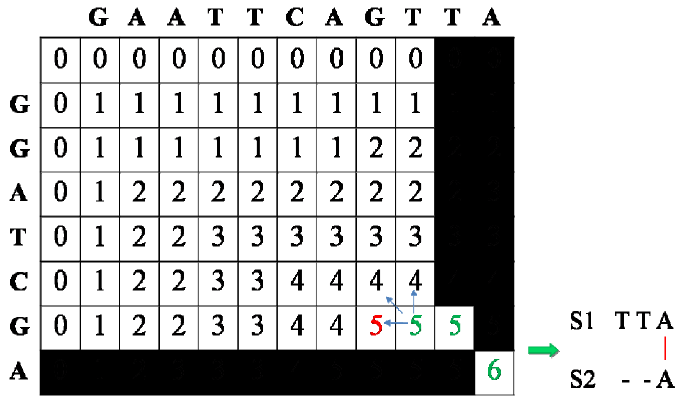


| | | G | A | A | T | T | C | A | G | T | T | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| G | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| G | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | |
| A | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | |
| T | 0 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | |
| C | 0 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | |
| G | 0 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 5 | 5 | 5 | |
| A | | | | | | | | | | | | 6 |

S1 TA

S2 -A

**(b)**

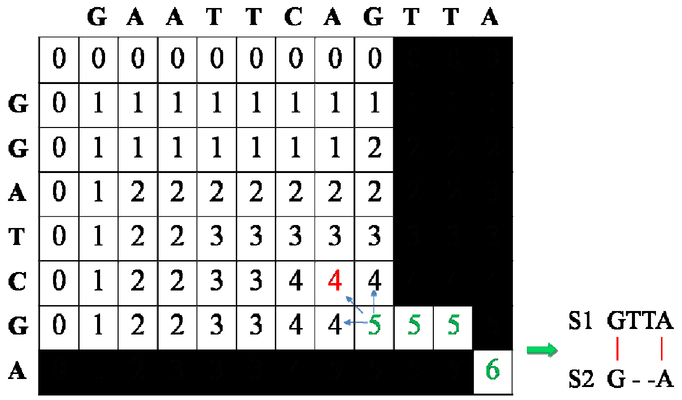


| | | G | A | A | T | T | C | A | G | T | T | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| G | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | |
| G | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | | |
| A | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | |
| T | 0 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 | | |
| C | 0 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 4 | 4 | | |
| G | 0 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 5 | 5 | 5 | |
| A | | | | | | | | | | | | 6 |

S1 TTA

S2 --A

**(c)**

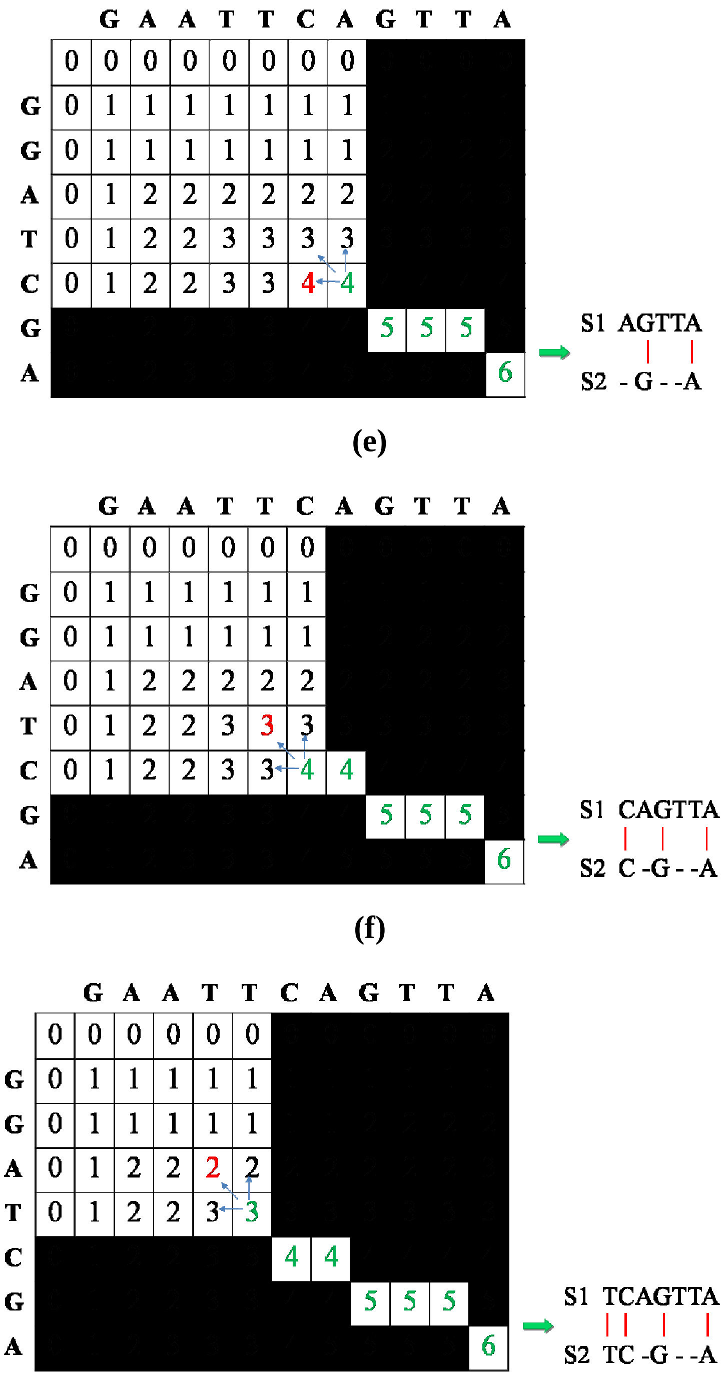


| | | G | A | A | T | T | C | A | G | T | T | A |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | |
| G | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | | | |
| G | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 2 | | | |
| A | 0 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | | | |
| T | 0 | 1 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | | | |
| C | 0 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 4 | | | |
| G | 0 | 1 | 2 | 2 | 3 | 3 | 4 | 4 | 5 | 5 | 5 | |
| A | | | | | | | | | | | | 6 |

S1 GTTA

S2 G--A

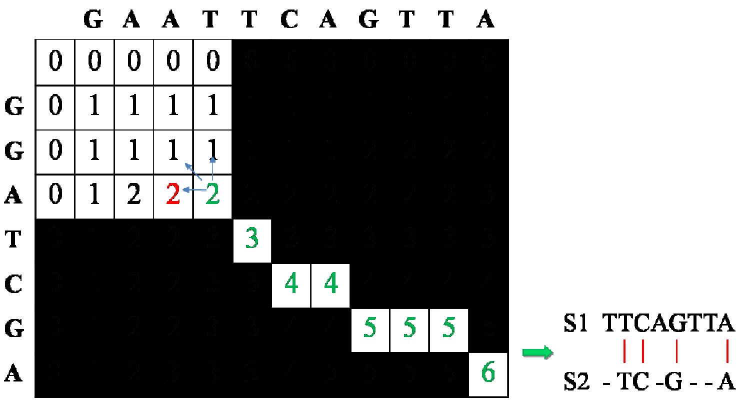
**(d)**


G A A T T C A G T T A
G G A T C G A
S1 AGTTA
S2 - G - -A
S1 CAGTTA
S2 C -G - -A
S1 TCAGTTA
S2 TC -G - -A

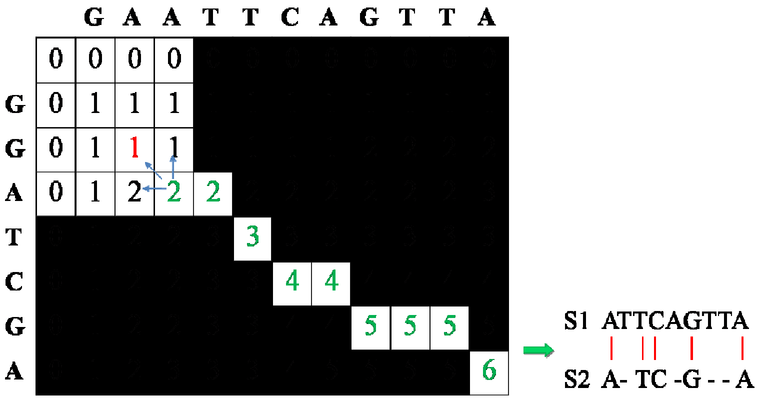
(e)

(f)

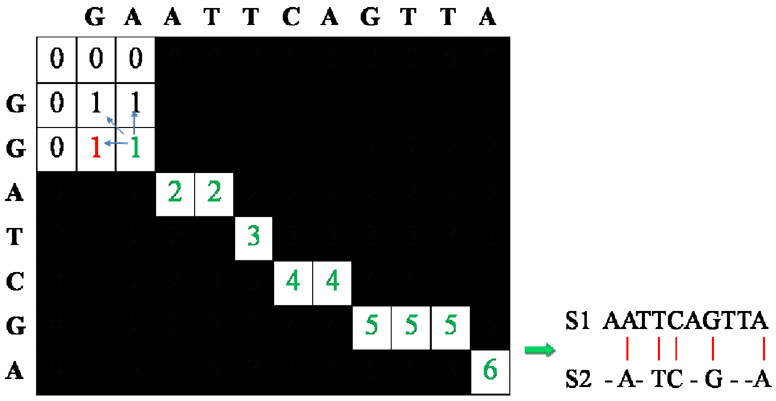
(g)


G A A T T C A G T T A
S1 TTCAGTTA
S2 - TC -G - - A

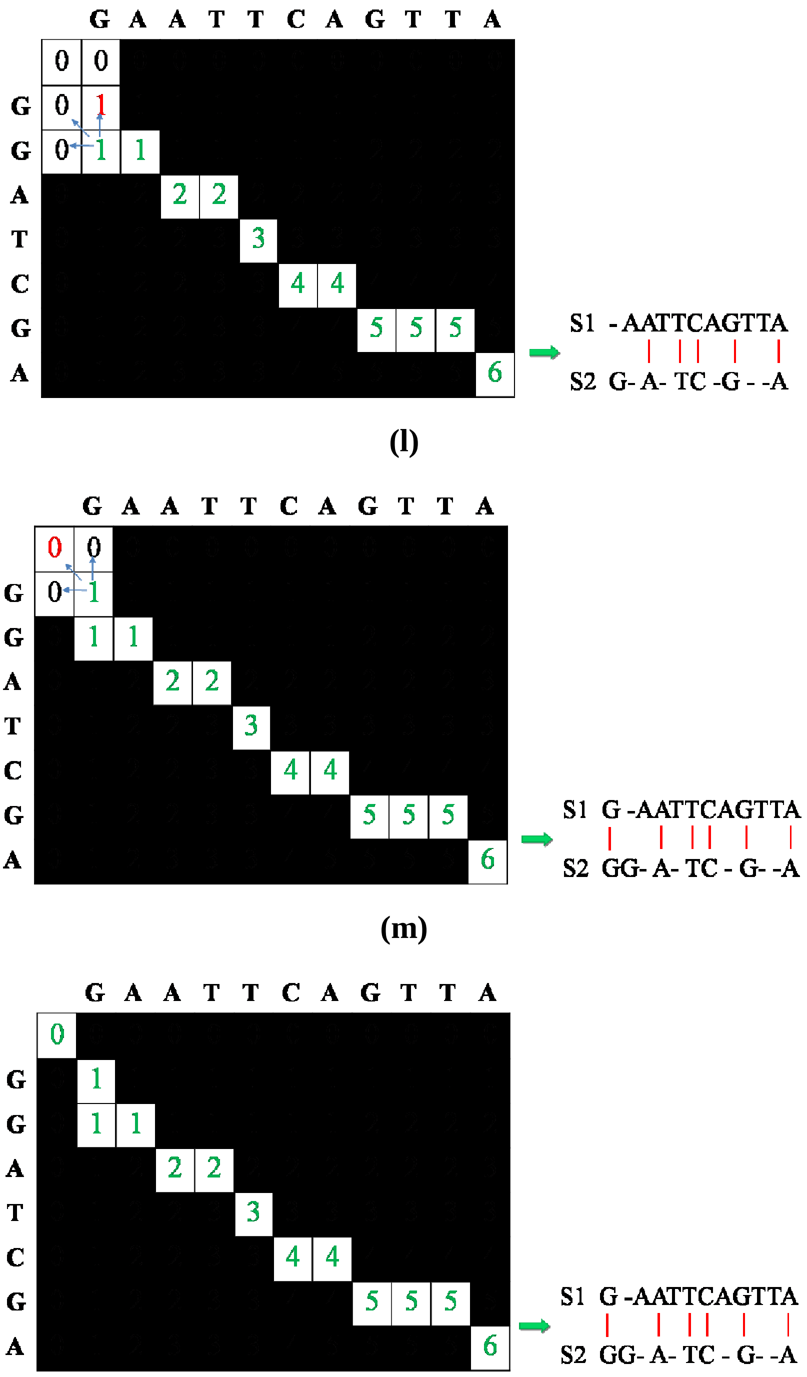
**(h)**


G A A T T C A G T T A
S1 ATTCAGTTA
S2 A- TC -G - - A

**(i)**


G A A T T C A G T T A
S1 AATTCAGTTA
S2 - A- TC - G - -A

**(j)**



(l)

(m)

(n)

**Figure 2.29** A graphical procedure of traceback solution [33]

# CHAPTER III
# MATERIALS AND METHODS

## 3.1 Introduction

BLAST is one of the most popularly used tools in bioinformatics to search and compare biological sequences. Fig. 3.1 illustrates the overall image of ***CADViST*** process. The idea of ***CADViST*** is to extract the essential information from BLAST output to visualize and analyze via graphical display. Therefore, in the proposed software, ***CADViST***, we propose the graphical representation especially for nucleotide sequence of Dengue virus, called UnitX, as the alternative tool to visualize and interpret the result obtained from BLAST. UnitX will also be used as the key feature for analysis tools of ***CADViST***.

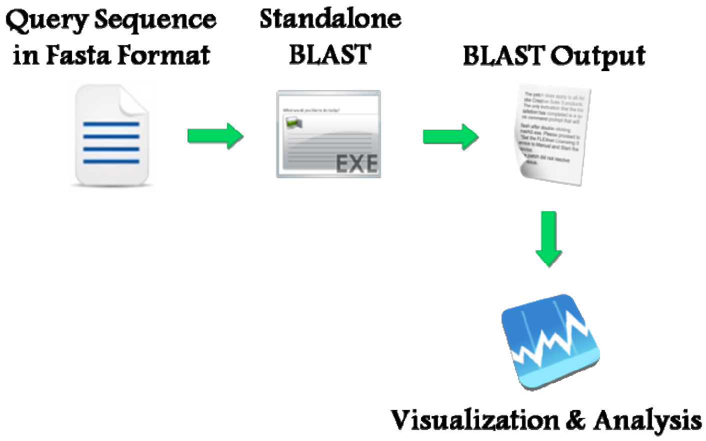


**Figure 3.1** Idea of ***CADViST***

## 3.2 The search of efficient graphical representative

The choice of selecting the graphical representation can be varied based on the characteristics of genomic sequences of interest. Since the goal of this research is to find the suitable graphical representation for nucleotide sequence of Dengue virus and the sequences can be of arbitrary lengths in huge database, therefore, computational complexity of the graphical representation needs to be low and base locations need to be easy to visualize. By comparing with many graphical representations of genomic sequence in the literature, this research proposes the novel graphical representation base on the modification of Yau’s method, called UnitX projection vector. UnitX is the new graphical representation especially for Dengue virus sequence analysis based on cumulative amount amino and keto bases.

### 3.2.1 UnitX Representation of nucleotide sequence for Dengue virus

In this thesis, we propose graphical representative to represent nucleotide sequence of Dengue virus, called UnitX. This method applies the idea of unit vector designed by Yau and his coworker also presented in section 2.2.1.2. UnitX represents the distribution of amino/keto bases along the sequence on two quadrants of Cartesian coordinate system, the first quadrant is amino (C and A) and keto (G and T) in the forth quadrant [34]. The vectors which have unit length in x-direction represented four nucleotides, i.e. adenines (A), guanine (G), thymine (T) and cytosine (C), are illustrated in Fig. 3.2:

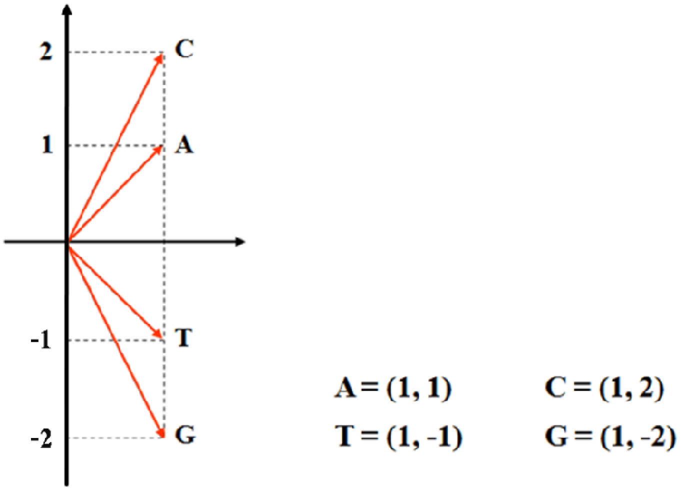


**Figure 3.2** Vector of UnitX: The vectors (1, 1), (1, 2), (1, -1), (1, -2) is used to represent four nucleotide A, C, T and G, respectively [34].

By given the numbers of nucleic acid occurring in the sequence, the $(x_n, y_n)$ coordinate projection onto $X$ and $Y$ axes can be demonstrated as follows:

$$x_{n-1} + \ (C_n + A_n + T_n + G_n) = x_n \quad \text{... (Equation. 3.1)}$$

$$y_{n-1} + \ (2C_n + A_n - T_n - 2G_n) = y_n \quad \text{... (Equation. 3.2)}$$

where $C_n, A_n, T_n$ and $G_n \in \{0, 1\}$ depend on the nucleotide type at position $n^{th}$.

Sections 3.2.2-3.2.4 show that UnitX is suitable for representing nucleotide sequence of Dengue virus. The idea is to find the relationship between unknown sequence and template represented each Dengue virus serotype via the correlation coefficient. The unknown sequence belongs to the serotype that yields the maximum correlation coefficient (see Fig 3.3).

The process on creating template is presented in section 3.2.2. Figs. 3.4 (a)-(d) show the template of all four Dengue virus serotypes. The classification process is presented in section 3.3.3. The performance of UnitX method is evaluated in section 3.2.4.

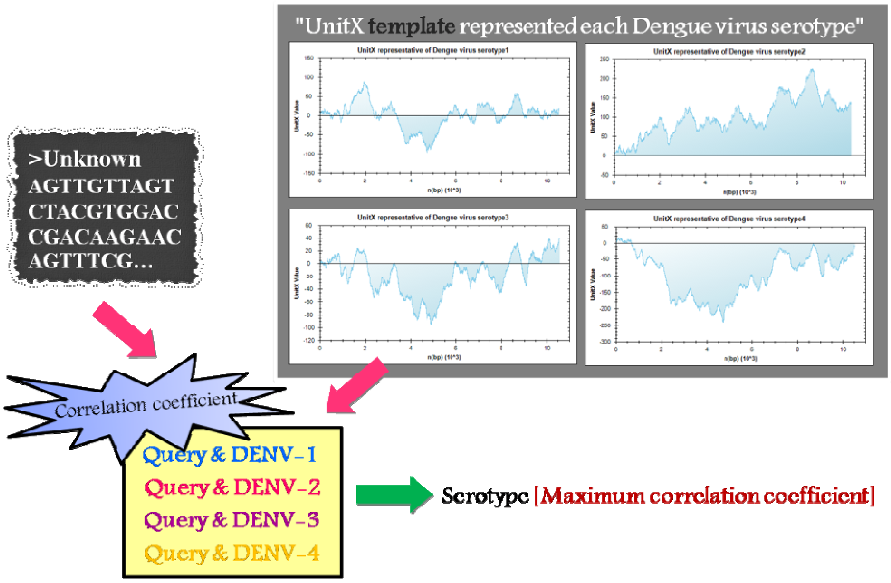

"UnitX template represented each Dengue virus serotype"
UnitX representative of Dengue virus serotype1
UnitX representative of Dengue virus serotype2
UnitX representative of Dengue virus serotype3
UnitX representative of Dengue virus serotype4
UnitX Value
n(bp) (10^3)
>Unknown
AGTTGTTAGT
CTACGTGGAC
CGACAAGAAC
AGTTTCG...
Correlation coefficient
Query & DENV-1
Query & DENV-2
Query & DENV-3
Query & DENV-4
Serotype [Maximum correlation coefficient]

**Figure 3.3** The idea to verify that UnitX is the suitable representation for nucleotide sequence of Dengue virus

### 3.2.2 Creating UnitX template to represent each Dengue virus serotype

The method used to construct UnitX representative of nucleotide sequences of each Dengue virus serotype can be summarized as follows:

**Step1:** Find $(X_n, Y_n)$ coordinate of each sequence by Equation. 3.1-3.2

**Step2:** Find $(X_n, Y'_n)$ coordinate of each sequence,

$$Y'_n = Y_n - \frac{\sum Y_n}{N}, \qquad \text{... (Equation 3.3)}$$

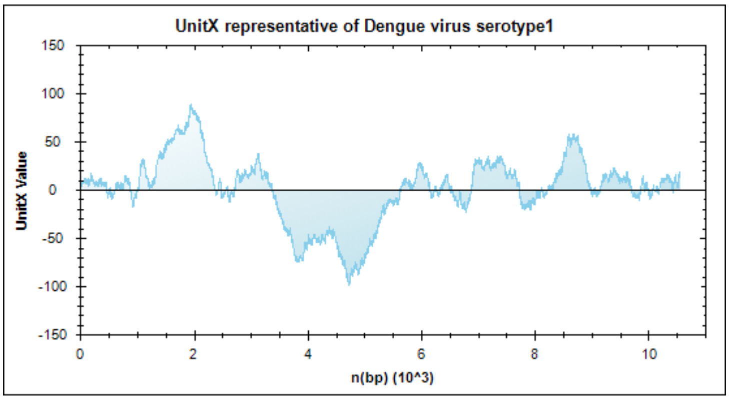
where $N$ is the number of base of the whole sequence. It should be noted that to reduce the variation of raw data, $Y_n$ is subtracted by $-\frac{\sum Y_n}{N}$.

**Step3:** Let $M$ be the number of training sequences for each Dengue virus serotype, the single optimal pattern of each serotype can be calculated as the centroid $(X'', Y'')$ of $\left(X_n^{(m)}, Y'^{(m)}_n\right)$ for all m= 1, 2, 3, ... , $M$ and $n$ = 1, 2, 3,…, $N$. The maximum value of $N$ is the minimum length of training sequences for each Dengue virus serotype.

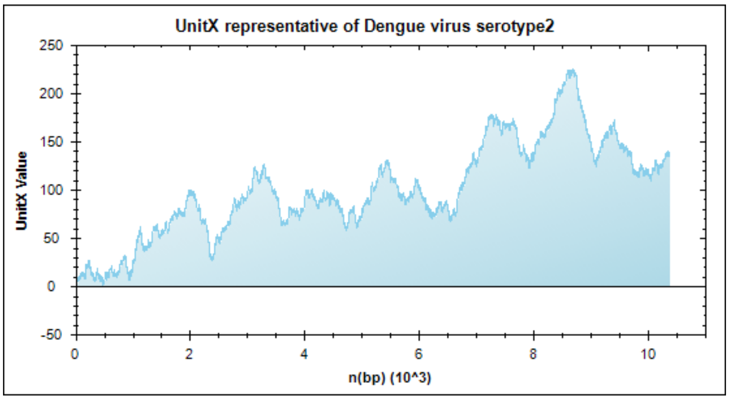
$$X'' = \frac{1}{M}\sum_{m=0}^{M} X_n^{(m)} \qquad \text{... (Equation 3.4)}$$

$$Y'' = \frac{1}{M}\sum_{m=0}^{M} Y'^{(m)}_n \qquad \text{... (Equation 3.5)}$$

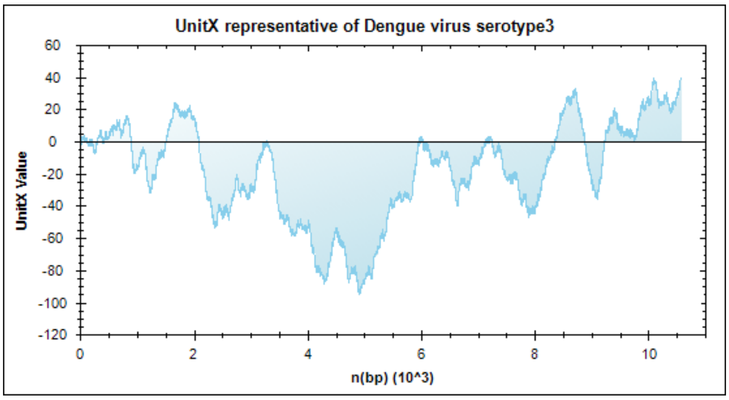
where $X_n^{(m)}$ and $Y'^{(m)}_n$ denote the coordinate $(\text{x}, y)$ of the m-th training sequence for each serotype [34].



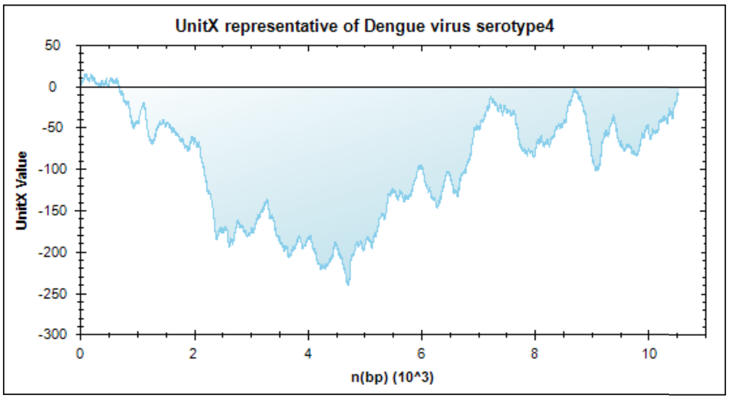
**(a)** UnitX representative of Dengue virus serotype 1



**(b)** UnitX representative of Dengue virus serotype 2



**(c)** UnitX representative of Dengue virus serotype 3



**(d)** UnitX representative of Dengue virus serotype 4

**Figure 3.4** UnitX representative of each Dengue virus serotype

### 3.2.3 Classification method

UnitX method can be used to automatically assign the serotype to the unknown Dengue virus sequence. The method to determine Dengue virus serotype uses the idea of measuring the similarity among the unknown Dengue virus sequence and UnitX template of all four Dengue virus serotypes. Their similarity is measured by correlation coefficients [34]. The classification process can be summarized as follows:

**Step1:** Follow Step1 and 2 of section 3.2.2

**Step2:** Compare the output of $Y'$ with the UnitX template $Y''$ of all four Dengue virus serotypes by measuring similarity via correlation coefficients. The unknown sequence belongs to the serotype that yields the maximum correlation coefficient.

### 3.2.4 Validation of UnitX method to classify all four Dengue virus serotype

To verify UnitX method, we employ this method to distinguish complete nucleotide sequence of all four Dengue virus serotypes. The sample sequences are obtained from GenBank database with keyword “Dengue virus complete genome”. All 2,184 sequences compose of four serotype each serotype contains 952, 737, 405 and 90 sequences, respectively. In this thesis, we estimate the performance of this method by four-fold cross validation, as shown in Table 3.1.

**Table 3.1** Summary of classification results by UnitX method [34]

| **Round** | **Number of testing sequences** | **Classification accuracy** |
|---|---|---|
| 1 | 1636 | 99.02 % |
| 2 | 1638 | 98.72 % |
| 3 | 1639 | 98.78 % |
| 4 | 1639 | 98.90 % |
| | | **98.86 %** |

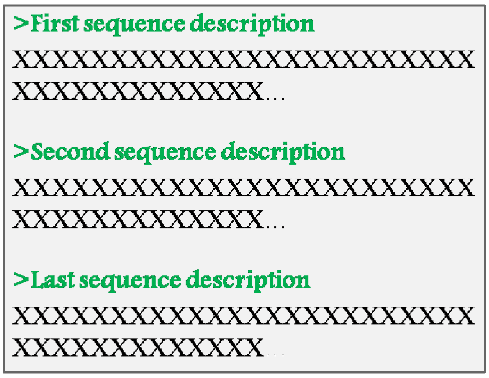
**<u>Note</u>:** The number of sample are not identical for each round because the data sample in each serotype is divided into 4 segments for four - fold cross validation and in fact it is not divisible by 4.

According to Table 3.1, we can obviously see that UnitX method can be efficiently used to classify the full genome of Dengue virus [34]. However, this method has a high computational complexity and a limited performance in processing partial sequence. The idea of CADVIST process is to search the best result with less computational time. In bioinformatics, the Basic Local Alignment Search Tool, or BLAST, is the existing simplification method to perform sequence similarity search. It is the available free software that can be used through the web interface or as standalone tool. CADVIST process employs standalone BLAST+ application to do BLAST search.

## 3.3 Selection database search program

According to the previous section, BLAST stands for Basic Local Alignment Search Tool. BLAST is the tool that uses heuristics algorithm to search for similarities between query sequence and all sequence in database. By heuristic algorithm, BLAST provides rapid sequence comparison but is not guaranteed to find the best alignment between query and database sequences. In the first version of CADVIST, we start with BLAST because it is faster than the existing sequence comparison tools and provide all the answers we needed. Besides, BLAST can be used to search more than one sequence at a time.

In this thesis, we employ standalone BLAST+ application to generate BLAST result. Standalone BLAST is the local installation of NCBI BLAST suite. The steps for downloading NCBI BLAST to local computer can be found from NCBI ftp site. The main advantage to run BLAST on your machine is to be able to create your own BLAST database. To create BLAST database, all the sequence should be assembled in one file in Fasta format or Multi-Fasta files containing all of the sequences you would like in the database. This file will be processed using program from NCBI, "*formatdb*", to build index for source database before this database can be searched by BLAST search command.



```
>First sequence description
XXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXX...

>Second sequence description
XXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXX...

>Last sequence description
XXXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXX
```

**Figure 3.5** Source file in Fasta format, Xs are nucleotide codes (A, T, G or C)

A sequence in Fasta format begins with one line description, followed by lines of sequence data. The first character of description line is greater-than (“>”) symbol. All lines of text should be shorter than 80 characters in length to facilitate viewing on screen (see Fig. 3.5).

In this thesis, we are focusing only on the nucleotide sequence of Dengue virus. The following “*formatdb*” command is use to create Dengue virus database from Fasta file called “All_Seqs.Fasta” containing all nucleotide sequences:

```
formatdb -p F -o T -i All_Seqs.Fasta
```

After executing “*formatdb*” command, three files will be produced from the source Fasta file. For nucleotide, the extensions are *nhr*, *nin* and *nsq*. To perform BLAST search, we use BLASTN to find the database sequence related to query sequence. First put query sequence in a file called “Query_Seq.Fasta”. Then we use “*blastn*” command to search for the nucleotide query. When the BLAST search is completed, the result is reported in text format. The following example command line is used for searching the input query file:

```
blastn.exe -db “All_Seqs.Fasta” -query “Query_Seq.Fasta” -outfmt 7
-num_descriptions 1 -num_alignments 1 -out “Result.txt”
```

## 3.4 Function of CADVIST

CADVIST consists of three main functions: (1) Dengue virus search engine, (2) UnitX visualization and analysis and (3) UnitX visualization referred to the user defined references. All of these functions can be described in more detail as below.

### 3.4.1 Dengue virus search engine

The idea of Dengue virus search engine is to locate partial matches on sequence data. This function employs standalone BLAST software to find the similarity score among the query sequence and Dengue virus nucleotide database. The results of tenth closest match show: (1) description of each database sequence found to match the query, (2) the start and end positions regard to database sequence found to match the query sequence and (3) the graphical visualization via UnitX displayed with database sequence found to match the query sequence. This function requires input in Fasta format. Once the input is inserted, the process inside CADVIST can be summarized as follows (see Fig 3.6):

**Step1:** Call the standalone BLAST+ program to generate BLAST output.

**Step2:** Extract the essential information from its output: (1) description of each database sequence found to match the query and (2) the matched nucleotide at the start and end positions regard to subject sequence found to match the query sequence.

**Step3:** Provide matched region among query and subject sequences via UnitX, then send the result to the display unit [35].

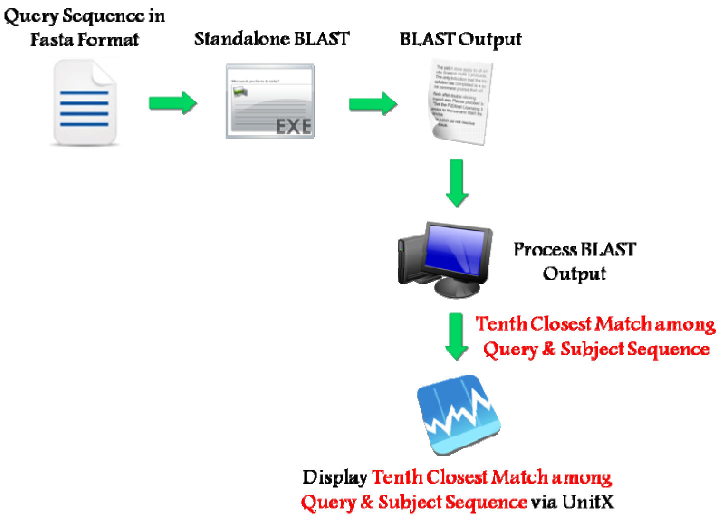

Query Sequence in Fasta Format
Standalone BLAST
EXE
BLAST Output
Process BLAST Output
Tenth Closest Match among Query & Subject Sequence
Display Tenth Closest Match among Query & Subject Sequence via UnitX

**Figure 3.6** Flowchart on the process of Dengue virus search engine

### 3.4.2 UnitX visualization and analysis

The idea of UnitX visualization and analysis is to visualize and analyze either partial or full length genomic sequences. This function employs standalone BLAST software to find the similarity score among the query sequence and optimal nucleotide sequence represented each serotype. The result provides (1) description of optimal nucleotide sequence found to match the query, (2) the start and end positions regard to the optimal nucleotide sequence that represents the result of the best match serotype and (3) the graphical visualization via UnitX displayed with the corresponding optimal nucleotide sequence. This function requires input in Fasta format. Once the input is inserted, the process inside *CADViST* can be summarized as follows (see Fig 3.7):

**Step1:** Call the standalone BLAST+ program to generate BLAST output.

**Step2:** Extract the essential information from its output: (1) description of optimal nucleotide sequence found to match the query and (2) the start and end positions regard to the optimal nucleotide sequence of best match serotype.

**Step3:** Provide matched region among query and optimal nucleotide sequence of best match serotype via UnitX, then send the result to display unit.

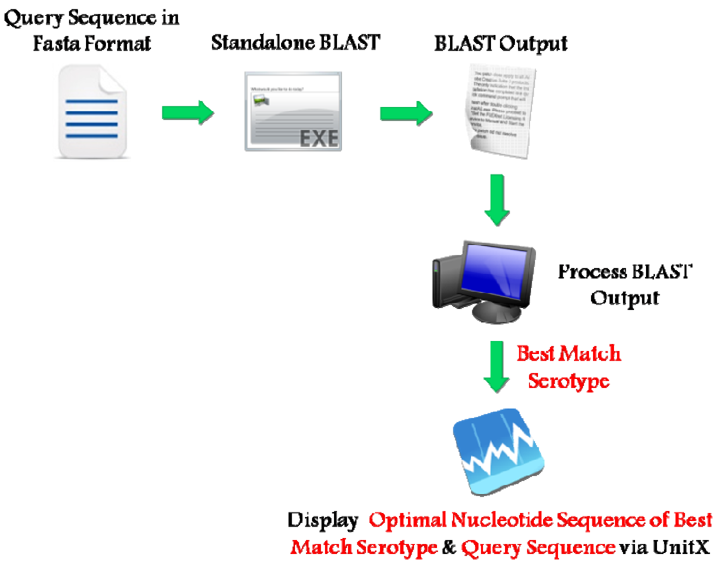

Query Sequence in Fasta Format
Standalone BLAST
EXE
BLAST Output
Process BLAST Output
Best Match Serotype
Display Optimal Nucleotide Sequence of Best Match Serotype & Query Sequence via UnitX

**Figure 3.7** Flowchart on the process of UnitX visualization and analysis

Furthermore, ***CADVIST*** can concurrently process multiple queries. In order to match a matching large number of queries, the attention is saving the execution time. The work of running BLAST search is done in Batch mode in order to increase speed of time performance. Accordingly, we employ BLAST in batch mode to search a large number of nucleotide sequences faster. For more details about finding optimal nucleotide sequence represented each Dengue virus serotype see in section 3.5.

### 3.4.3 UnitX visualization referred to user defined references

The idea of UnitX visualization referred to user defined references preferred user to display the Tabular output from BLAST+ application that is sorted by serotype and Subject Id. The results of this function provide: (1) description of each database sequence found to match the query, (2) the matched nucleotide at the start and end positions regard to the subject sequence found to match the query sequence and (3) the graphical visualization via UnitX displayed with the subject sequence found to match the query sequence. Consequently, the main advantage of this function is that the users can define their own reference database. This function requires three inputs: (1) the Tabular output from BLAST+ application, (2) query sequence in Fasta format and (3) reference sequence in Fasta format. Once the input is inserted, the process inside ***CADVIST*** can be summarized as follows (Fig. 3.8):

**Step1:** Extract the essential information from BLAST+ output: (1) description of each database sequence found to match the query and (2) the matched nucleotide at the start and end positions regard to subject sequence found to match the query sequence.

**Step2:** Sort query sequence by serotype and Subject ID.

**Step3:** Provide matched region among query and subject sequence via UnitX and then send the results to the display unit.

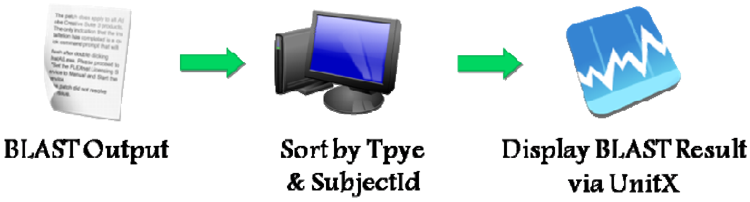


**Figure 3.8** Flowchart on the process of UnitX visualization referred to user defined references

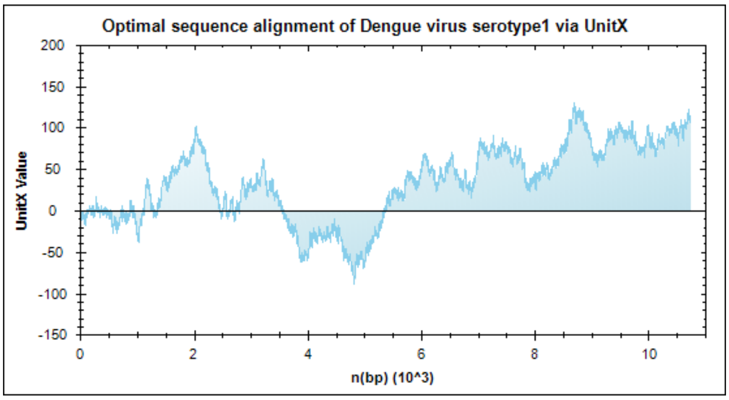
## 3.5 Creating the optimal nucleotide sequence to represent each Dengue virus serotype

One objective of this thesis is to find one representative nucleotide sequence for each Dengue virus serotype. Generally, the same group of organisms should have similar patterns of biological sequence. Since the sequences are not perfectly aligned, we cannot directly make the template by aligning the whole sequence. Therefore, we apply multiple sequence alignment to find sequence similarity that may be widely dispersed or hidden in the sequence. In this thesis, we random one-tenth of each serotype, then produce multiple sequence alignment by ClustalW (see section 2.5). The step to find the optimal representative of each Dengue virus serotype is as follows:

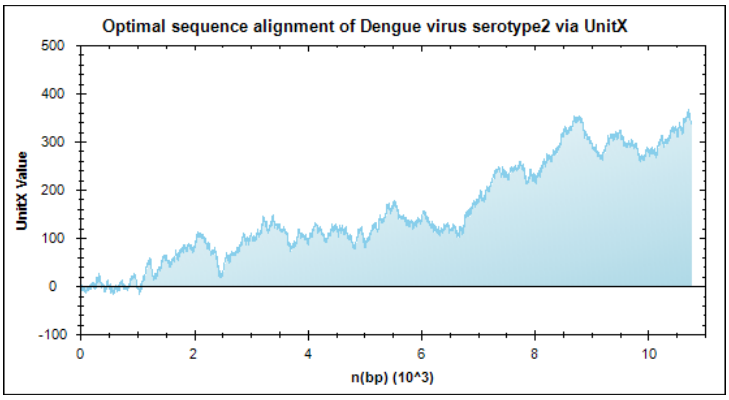
**Step1:** Random one-tenth of each serotype to align using ClustalW.

**Step2:** Find the most frequently value of each nucleotide in each position.

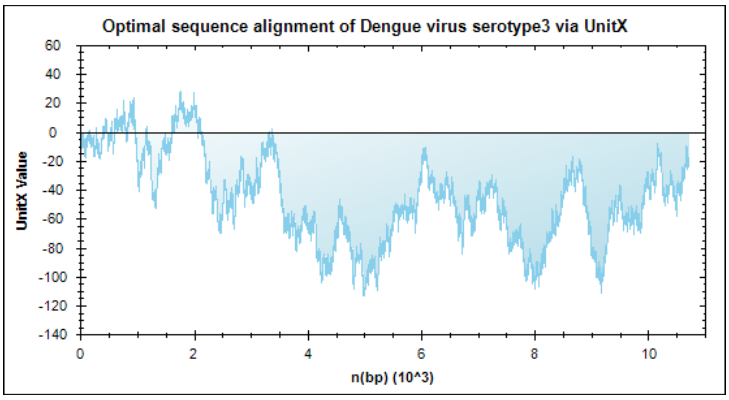
**Step3:** Convert the most frequently value of each serotype to UnitX representative.



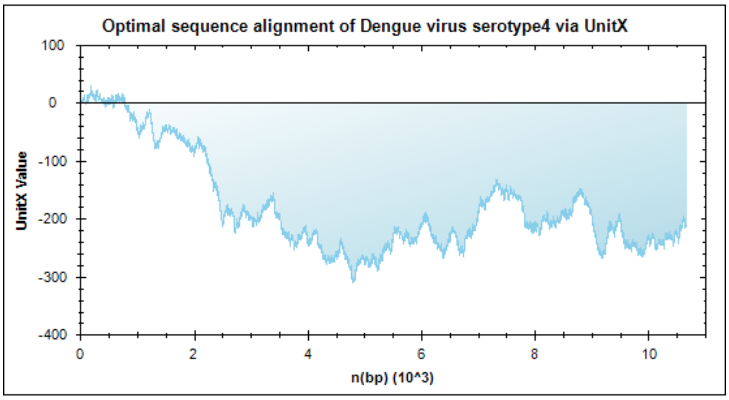
**(a)** Optimal sequence alignment of Dengue virus serotype 1 via UnitX



**(b)** Optimal sequence alignment of Dengue virus serotype 2 via UnitX



**(c)** Optimal sequence alignment of Dengue virus serotype 3 via UnitX



**(d)** Optimal sequence alignment of Dengue virus serotype 4 via UnitX

**Figure 3.9** Optimal sequence alignment of each Dengue virus serotype via UnitX

Figs. 3.9 illustrate the optimal sequence alignment of each Dengue virus serotype via UnitX. To verify the optimal sequence alignment created by multiple sequence alignment technique, we employ these sequences as database sequence of standalone BLAST+ application. After that BLAST search is performed to assign serotype to the unknown Dengue virus sequence. In this experiment, all 2,184 sample sequences compose of four serotypes each serotype contains 952, 737, 405 and 90 sequences, respectively. We employ 10% of each serotype to create the optimal nucleotide represented each serotype. According to Table 3.2, we can conclude that this technique is able to classify all four Dengue virus serotypes with a high accuracy.

**Table 3.2** Summary of classification results by multiple sequence alignment technique

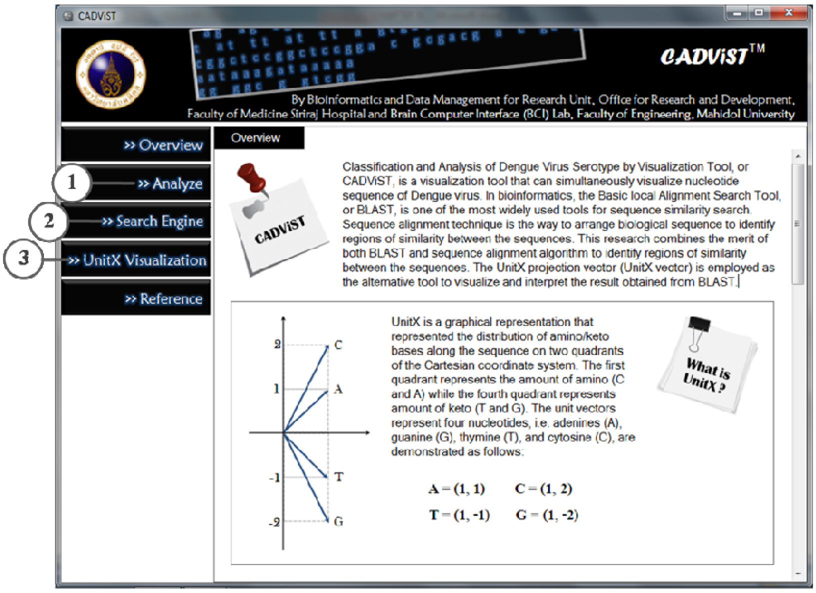
| **Round** | **Number of sample sequences** | **Classification accuracy** |
|---|---|---|
| 1 | 856 | 100 % |
| 2 | 663 | 100 % |
| 3 | 364 | 100 % |
| 4 | 81 | 100 % |
| | | **100 %** |

# CHAPTER IV
# RESULTS

In the previous chapter, we introduced the proposed software, CADVIST. We know that CADVIST consists of three main functions: (1) Dengue virus search engine, (2) UnitX visualization and analysis and (3) UnitX visualization referred to the user defined references. This chapter aim to describe the result of these functions. Firstly, we introduce the main interface of CADVIST. Then, we illustrate the result obtained from each function.

CADVIST interface are implemented in C# programming languages with .Net Framework. It combines standalone BLAST+ application so that it can automatically perform BLAST search. The nucleotide sequence database includes Dengue virus sequences available in GenBank database. Accordingly, CADVIST is good for local user on a standalone computer.

Fig. 4.1 shows the front page of CADVIST. The main menu of CADVIST consists of three buttons: (1) “Analyze”, (2) “Search Engine” and (3) “UnitX Visualization”. The user can access the desired function by pressing a corresponding button on the left hand side.


CADVIST
CADVIST™
By BioInformatics and Data Management for Research Unit, Office for Research and Development,
Faculty of Medicine Siriraj Hospital and Brain Computer Interface (BCI) Lab, Faculty of Engineering, Mahidol University
1
2
3
>> Overview
>> Analyze
>> Search Engine
>> UnitX Visualization
>> Reference
Overview
CADVIST
What is UnitX ?
2
1
-1
-2
C
A
T
G

Classification and Analysis of Dengue Virus Serotype by Visualization Tool, or CADViST, is a visualization tool that can simultaneously visualize nucleotide sequence of Dengue virus. In bioinformatics, the Basic local Alignment Search Tool, or BLAST, is one of the most widely used tools for sequence similarity search. Sequence alignment technique is the way to arrange biological sequence to identify regions of similarity between the sequences. This research combines the merit of both BLAST and sequence alignment algorithm to identify regions of similarity between the sequences. The UnitX projection vector (UnitX vector) is employed as the alternative tool to visualize and interpret the result obtained from BLAST.

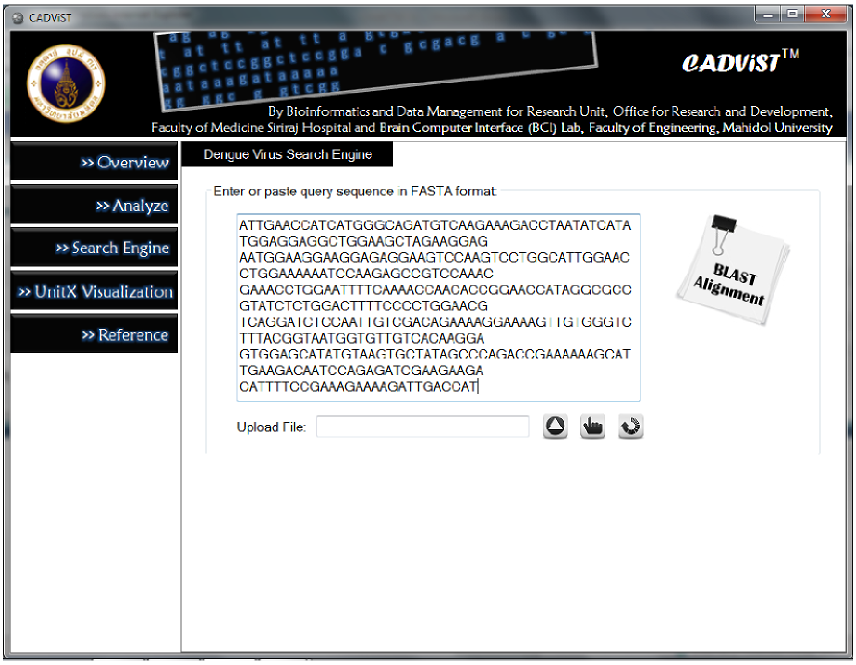
UnitX is a graphical representation that represented the distribution of amino/keto bases along the sequence on two quadrants of the Cartesian coordinate system. The first quadrant represents the amount of amino (C and A) while the fourth quadrant represents amount of keto (T and G). The unit vectors represent four nucleotides, i.e. adenines (A), guanine (G), thymine (T), and cytosine (C), are demonstrated as follows:

A = (1, 1) C = (1, 2)
T = (1, -1) G = (1, -2)

**Figure 4.1** ***CADViST*** main interface

## 4.1 Result of Dengue virus search engine

As an example, we employ ***CADViST*** to process the partial sequence of GQ199895 Dengue virus 2 isolate DENV-2/NI/BID-V2683/1999. The input is copied and put into the blank space (see Fig 4.2). According to ***CADViST*** process, the result has two parts. The first part gives the information about query sequence. The second part shows the details of tenth closest match. According to Fig. 4.3, the first part reports one line description and UnitX representative of partial sequence from GQ199895 Dengue virus 2 isolate DENV-2/NI/BID-V2683/1999, complete genome, 2037 – 5067 base position (3031 letters) and the second part shows the details of the first (GQ199895) and second (FJ850065) highest scores matched sequences *etc*.

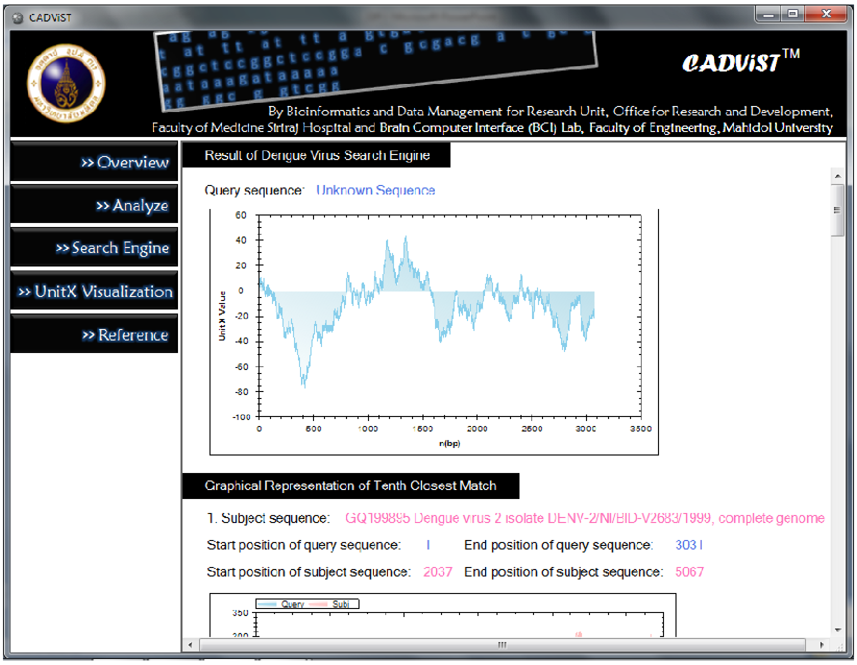

CADVIST
CADVIST™
By Bioinformatics and Data Management for Research Unit, Office for Research and Development,
Faculty of Medicine Siriraj Hospital and Brain Computer Interface (BCI) Lab, Faculty of Engineering, Mahidol University
>> Overview
>> Analyze
>> Search Engine
>> UnitX Visualization
>> Reference
Dengue Virus Search Engine
BLAST Alignment

Enter or paste query sequence in FASTA format

```
ATTGAACCATCATGGGCAGATGTCAAGAAAGACCTAATATCATA
TGGAGGAGGCTGGAAGCTAGAAGGAG
AATGGAAGGAAGGAGAGGAAGTCCAAGTCCTGGCATTGGAAC
CTGGAAAAAATCCAAGAGCCGTCCAAAC
GAAACCTGGAATTTTCAAAACCAACACCGGAACCATAGGCGCC
GTATCTCTGGACTTTTCCCCTGGAACG
TCAGGATCTCCAATTGTCGACAGAAAAGGAAAAGTTGTGGGTC
TTTACGGTAATGGTGTTGTCACAAGGA
GTGGAGCATATGTAAGTGCTATAGCCCAGACCGAAAAAGCAT
TGAAGACAATCCAGAGATCGAAGAAGA
CATTTTCCGAAAGAAAAGATTGACCAT
```

Upload File:

**Figure 4.2** “Dengue virus search engine” interface


CADVIST
CADVIST™
By Bioinformatics and Data Management for Research Unit, Office for Research and Development,
Faculty of Medicine Siriraj Hospital and Brain Computer Interface (BCI) Lab, Faculty of Engineering, Mahidol University
>> Overview
>> Analyze
>> Search Engine
>> UnitX Visualization
>> Reference
Result of Dengue Virus Search Engine
60
40
20
0
-20
-40
-60
-80
-100
UnitX Value
0
500
1000
1500
2000
2500
3000
3500
n(bp)
Graphical Representation of Tenth Closest Match
Query
Subj
350
300

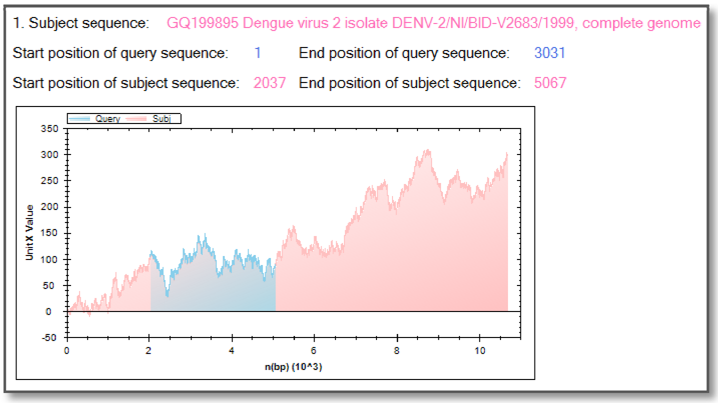
Query sequence: Unknown Sequence

1. Subject sequence: GQ199895 Dengue virus 2 isolate DENV-2/NI/BID-V2683/1999, complete genome

Start position of query sequence: 1 End position of query sequence: 3031

Start position of subject sequence: 2037 End position of subject sequence: 5067

**Figure 4.3** Example result of “Dengue virus search engine” interface

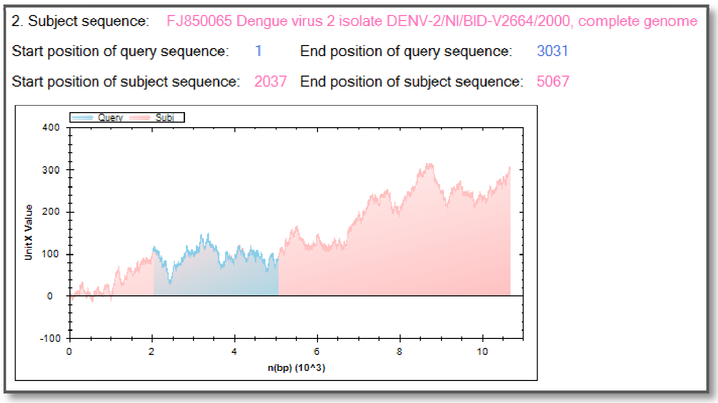


1. Subject sequence: GQ199895 Dengue virus 2 isolate DENV-2/NI/BID-V2683/1999, complete genome

Start position of query sequence: 1 End position of query sequence: 3031

Start position of subject sequence: 2037 End position of subject sequence: 5067

**(a)**



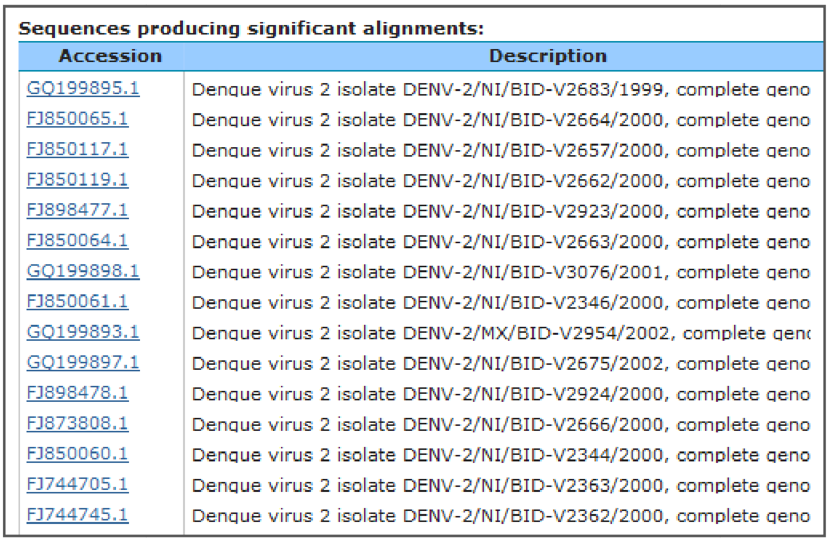
2. Subject sequence: FJ850065 Dengue virus 2 isolate DENV-2/NI/BID-V2664/2000, complete genome

Start position of query sequence: 1 End position of query sequence: 3031

Start position of subject sequence: 2037 End position of subject sequence: 5067

**(b)**

**Figure 4.4** (a) The first (GQ199895) and (b) second (FJ850065) highest scores matched sequences



**Sequences producing significant alignments:**

| Accession | Description |
|---|---|
| GQ199895.1 | Dengue virus 2 isolate DENV-2/NI/BID-V2683/1999, complete geno |
| FJ850065.1 | Dengue virus 2 isolate DENV-2/NI/BID-V2664/2000, complete geno |
| FJ850117.1 | Dengue virus 2 isolate DENV-2/NI/BID-V2657/2000, complete geno |
| FJ850119.1 | Dengue virus 2 isolate DENV-2/NI/BID-V2662/2000, complete geno |
| FJ898477.1 | Dengue virus 2 isolate DENV-2/NI/BID-V2923/2000, complete geno |
| FJ850064.1 | Dengue virus 2 isolate DENV-2/NI/BID-V2663/2000, complete geno |
| GQ199898.1 | Dengue virus 2 isolate DENV-2/NI/BID-V3076/2001, complete geno |
| FJ850061.1 | Dengue virus 2 isolate DENV-2/NI/BID-V2346/2000, complete geno |
| GQ199893.1 | Dengue virus 2 isolate DENV-2/MX/BID-V2954/2002, complete gen |
| GQ199897.1 | Dengue virus 2 isolate DENV-2/NI/BID-V2675/2002, complete geno |
| FJ898478.1 | Dengue virus 2 isolate DENV-2/NI/BID-V2924/2000, complete geno |
| FJ873808.1 | Dengue virus 2 isolate DENV-2/NI/BID-V2666/2000, complete geno |
| FJ850060.1 | Dengue virus 2 isolate DENV-2/NI/BID-V2344/2000, complete geno |
| FJ744705.1 | Dengue virus 2 isolate DENV-2/NI/BID-V2363/2000, complete geno |
| FJ744745.1 | Dengue virus 2 isolate DENV-2/NI/BID-V2362/2000, complete geno |

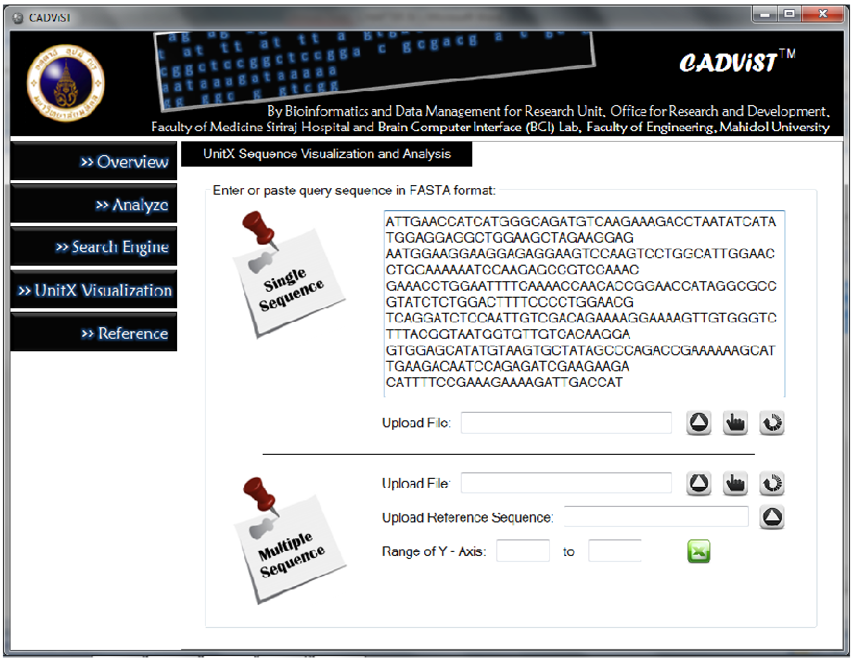
**Figure 4.5** One-line description from NCBI BLAST webpage

According to the one line descriptions of each database sequence that found to match the query sequence obtained from NCBI Homepage, the first and second highest scores matched sequences is GQ199895 and FJ850065, respectively (see Fig 4.5). The first and second highest score from CADViST analysis (see Figs 4.4) shows the same as the first and second highest matched sequence of NCBI analysis.

By employing the essential information from BLAST result, CADViST provides an extensive picture of the results for further analysis. This also provides very useful information for sequence contamination checking.

## 4.2 Result of UnitX visualization and analysis

This function is capable of predicting serotype of Dengue virus in both partial and full lengths of query sequence. As an example, we employ CADViST to process the partial sequence of GQ199895 Dengue virus 2 isolate DENV-2/NI/BID-V2683/1999. From the description, we know that query sequence is Dengue virus serotype 2. Once the input is copied and put into the blank space (see Fig 4.6), CADViST calls standalone BLAST to generate BLAST output. Then, it returns the similarity score among the query sequence and optimal library represented each serotype. By employing the essential information from BLAST output, CADViST provides (1) description of the optimal library found to match the query, (2) the start and end positions regard to the optimal library represented the result of the best match serotype and (3) the graphical visualization via UnitX displayed with the corresponding optimal library. As expected, the optimal library found to match the query is the optimal library of Dengue virus serotype 2 (see Fig 4.7). Furthermore, this function can simultaneously process multiple queries. Therefore, the benefit of this function is that the user can simultaneously compare multiple sequences.



By Bioinformatics and Data Management for Research Unit, Office for Research and Development, Faculty of Medicine Siriraj Hospital and Brain Computer Interface (BCI) Lab, Faculty of Engineering, Mahidol University

UnitX Sequence Visualization and Analysis

Enter or paste query sequence in FASTA format:

```
ATTGAACCATCATGGGCAGATGTCAAGAAAGACCTAATATCATA
TGGAGGAGGCTGGAAGCTAGAAGGAG
AATGGAAGGAAGGAGAGGAAGTCCAAGTCCTGGCATTGGAAC
CTGCAAAAAATCCAAGAGCCGTCCAAAC
GAAACCTGGAATTTTCAAAACCAACACCGGAACCATAGGCGCC
GTATCTCTGGACTTTTCCCCTGGAACG
TCAGGATCTCCAATTGTCGACAGAAAAGGAAAAGTTGTGGGTC
TTTACGGTAATGGTGTTGTCACAAGGA
GTGGAGCATATGTAAGTGCTATAGCCCAGACCGAAAAAAGCAT
TGAAGACAATCCAGAGATCGAAGAAGA
CATTTTCCGAAAGAAAAGATTGACCAT
```

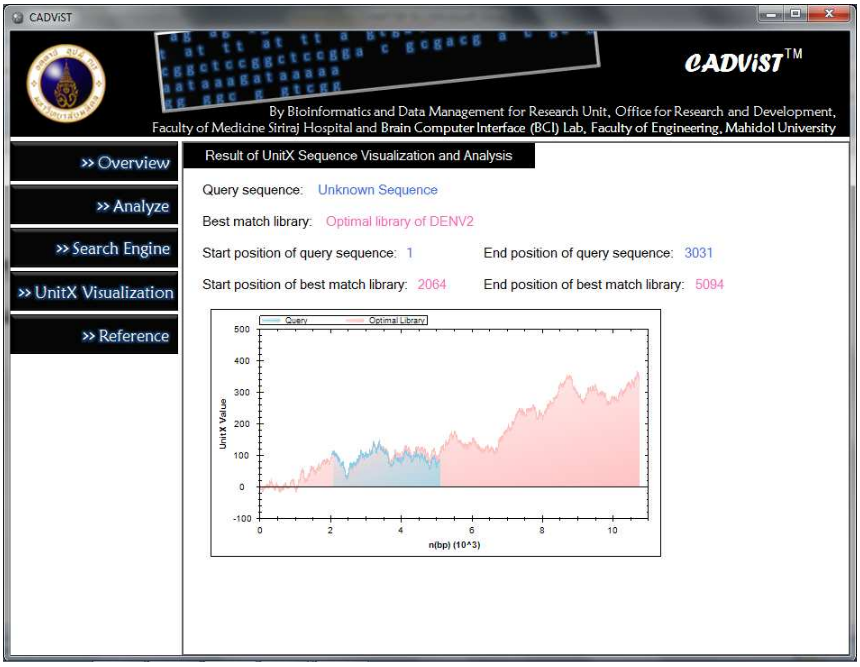
Upload File:

Upload File:

Upload Reference Sequence:

Range of Y - Axis: to

**Figure 4.6** “UnitX visualization and analysis” interface



By Bioinformatics and Data Management for Research Unit, Office for Research and Development, Faculty of Medicine Siriraj Hospital and Brain Computer Interface (BCI) Lab, Faculty of Engineering, Mahidol University

Result of UnitX Sequence Visualization and Analysis

Query sequence: Unknown Sequence

Best match library: Optimal library of DENV2

Start position of query sequence: 1 End position of query sequence: 3031

Start position of best match library: 2064 End position of best match library: 5094

**Figure 4.7** Example result of “UnitX sequence visualization and analysis” interface

## 4.3 Results of UnitX visualization referred to the user defined references

This function is used to display the tabular output from BLAST+ application that users can define their own reference sequence. The user must provide three inputs to the system: (1) the tabular output from BLAST+ application (2) - (3) query and database in Fasta file format. To explain the results of this function, all 100 query samples constructed by combining twenty sequences with different lengths.

In this section, we first explain the way to create Tabular output from BLAST+ application. This example assumes that sample query sequence contained in a file called “Query_Seq.fasta” and BLAST database using the name “All_Seqs.fasta”. The results are the output as the file called “BLAST_Result.txt”. The step to create Tabular output file can be summarized as follow:

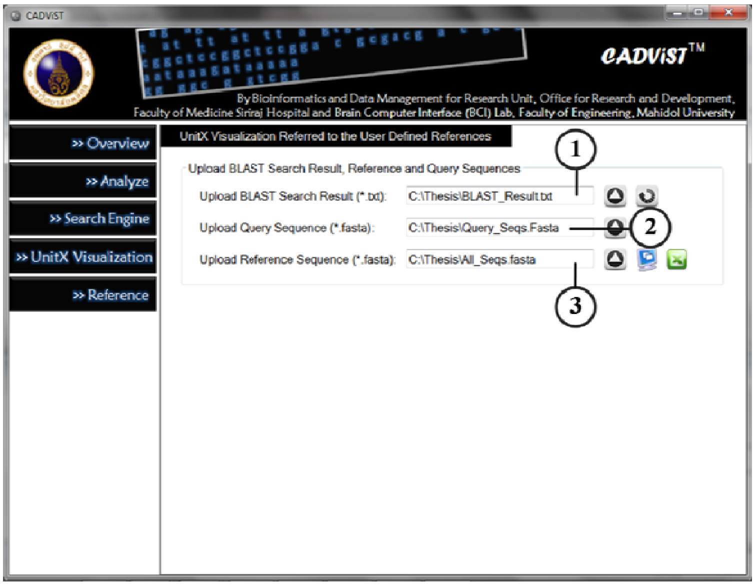
**Step1:** Use the following command to create BLAST search database,

```
formatdb -p F -o T -i All_Seqs.fasta
```

**Step2:** Perform BLAST search by the command below. When the BLAST search is completed, the result is reported in Tabular format.

```
blastn.exe -db All_Seqs.fasta -query Query_Seqs.fasta -outfmt 7
 -num_descriptions 1 -num_alignments 1 -out BLAST_Result.txt
```

Once three inputs are put into the blank space (see Fig. 4.8), **CADViST** returns the result from Tabular output file by sorting both serotype and Subject ID (see Fig. 4.9).


CADViST
CADViST™
By Bioinformatics and Data Management for Research Unit, Office for Research and Development, Faculty of Medicine Siriraj Hospital and Brain Computer Interface (BCI) Lab, Faculty of Engineering, Mahidol University
» Overview
» Analyze
» Search Engine
» UnitX Visualization
» Reference
1
2
3

UnitX Visualization Referred to the User Defined References

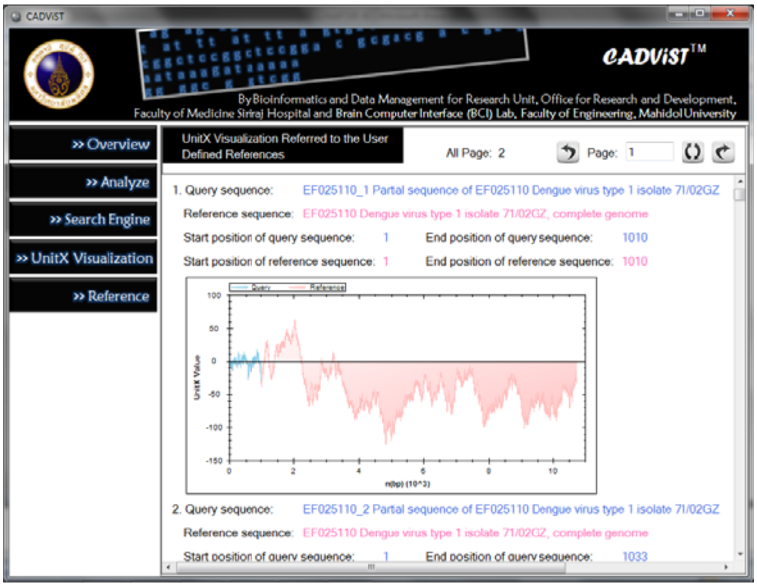
Upload BLAST Search Result, Reference and Query Sequences

Upload BLAST Search Result (*.txt): C:\Thesis\BLAST_Result.txt

Upload Query Sequence (*.fasta): C:\Thesis\Query_Seqs.Fasta

Upload Reference Sequence (*.fasta): C:\Thesis\All_Seqs.fasta

**Figure 4.8** “UnitX visualization referred to the user defined references” interface


CADViST
CADViST™
By Bioinformatics and Data Management for Research Unit, Office for Research and Development, Faculty of Medicine Siriraj Hospital and Brain Computer Interface (BCI) Lab, Faculty of Engineering, Mahidol University
» Overview
» Analyze
» Search Engine
» UnitX Visualization
» Reference
Query
Reference
100
50
0
-50
-100
-150
UnitX Value
0
2
4
6
8
10
n(bp) (10^3)

UnitX Visualization Referred to the User Defined References

All Page: 2 Page: 1

1. Query sequence: EF025110_1 Partial sequence of EF025110 Dengue virus type 1 isolate 71/02GZ

Reference sequence: EF025110 Dengue virus type 1 isolate 71/02GZ, complete genome

Start position of query sequence: 1 End position of query sequence: 1010

Start position of reference sequence: 1 End position of reference sequence: 1010

2. Query sequence: EF025110_2 Partial sequence of EF025110 Dengue virus type 1 isolate 71/02GZ

Reference sequence: EF025110 Dengue virus type 1 isolate 71/02GZ, complete genome

Start position of query sequence: 1 End position of query sequence: 1033

**Figure 4.9** Example result of “UnitX visualization referred to the user defined references” interface

Table 4.1 illustrates the essential information to explain the result of this function. This table contains eight columns. Column one to five is the information of Tabular output from a file named “BLAST_Result.txt” and the last column is serotype of database sequence found to match query sequence from a file named “All_Seqs.fasta”. The first column contains the ordinal number of that sequence in the query sequence file from a file named “Query_Seq.fasta”. The second column contains the name of query sequence. The third column contains the name of database sequence found to match the query sequence. Columns four and five contain the start and end position of query sequence. Columns six and seven contain the start and end position of database sequence.

**Table 4.1** Example data explained the result of "UnitX visualization referred to the user defined references" function

| Index | Query ID | Subject ID | Q.Start | Q.End | S.Start | S.End | Serotype |
|---|---|---|---|---|---|---|---|
| 1 | FJ226067_1 | FJ226067 | 1 | 4039 | 1 | 4039 | 4 |
| 2 | GQ868585_1 | GQ868585 | 1 | 3505 | 7102 | 10606 | 4 |
| 3 | GQ868594_1 | GQ868594 | 1 | 3071 | 4049 | 7119 | 4 |
| 4 | FJ882598_1 | FJ882598 | 1 | 10606 | 1 | 10606 | 4 |
| 5 | EU854299_1 | EU854299 | 1 | 10606 | 1 | 10606 | 4 |
| 6 | DQ863638_1 | DQ863638 | 1 | 10659 | 1 | 10659 | 3 |
| 7 | GQ868578_1 | GQ868578 | 1 | 10659 | 1 | 10659 | 3 |
| 8 | GQ252674_1 | GQ252674 | 1 | 6594 | 4089 | 10682 | 3 |
| 9 | FN429918_1 | FN429918 | 1 | 4075 | 3065 | 7139 | 3 |
| 10 | FJ390371_1 | FJ390371 | 1 | 502 | 10147 | 10648 | 3 |
| 11 | GQ868604_1 | GQ868604 | 1 | 1010 | 1 | 1010 | 2 |
| 12 | FJ226066_1 | FJ226066 | 1 | 1010 | 5054 | 6063 | 2 |
| 13 | FN429891_1 | FN429891 | 1 | 106 | 8095 | 8200 | 2 |
| 14 | AB479042_1 | AB479042 | 1 | 2066 | 5053 | 7118 | 2 |
| 15 | EU482639_1 | EU482639 | 1 | 9600 | 1080 | 10679 | 2 |
| 16 | FJ469908_1 | FJ469908 | 1 | 3068 | 4085 | 7152 | 1 |
| 17 | EF025110_1 | EF025110 | 1 | 1010 | 1 | 1010 | 1 |
| 18 | GQ868539_1 | GQ868539 | 1 | 552 | 10112 | 10663 | 1 |
| 19 | FJ410205_1 | FJ410205 | 1 | 5616 | 5061 | 10676 | 1 |
| 20 | EU482481_1 | EU482481 | 1 | 591 | 10100 | 10690 | 1 |

| Index | Query ID | Subject ID | Q.Start | Q.End | S.Start | S.End | Serotype |
|---|---|---|---|---|---|---|---|
| 21 | FJ226067_2 | FJ226067 | 1 | 1043 | 8102 | 9144 | 4 |
| 22 | GQ868585_2 | GQ868585 | 1 | 1010 | 1 | 1010 | 4 |
| 23 | GQ868594_2 | GQ868594 | 1 | 503 | 10105 | 10607 | 4 |
| 24 | FJ882598_2 | FJ882598 | 1 | 3047 | 1 | 3047 | 4 |
| 25 | EU854299_2 | EU854299 | 1 | 1017 | 1018 | 2034 | 4 |
| 26 | DQ863638_2 | DQ863638 | 1 | 2019 | 2024 | 4042 | 3 |
| 27 | GQ868578_2 | GQ868578 | 1 | 3028 | 2026 | 5053 | 3 |
| 28 | GQ252674_2 | GQ252674 | 1 | 2019 | 3033 | 5051 | 3 |
| 29 | FN429918_2 | FN429918 | 1 | 2025 | 4105 | 6129 | 3 |
| 30 | FJ390371_2 | FJ390371 | 1 | 4586 | 6063 | 10648 | 3 |
| 31 | GQ868604_2 | GQ868604 | 1 | 3029 | 1015 | 4043 | 2 |
| 32 | FJ226066_2 | FJ226066 | 1 | 3029 | 1019 | 4047 | 2 |
| 33 | FN429891_2 | FN429891 | 1 | 1020 | 5057 | 6076 | 2 |
| 34 | AB479042_2 | AB479042 | 1 | 3056 | 3045 | 6100 | 2 |
| 35 | EU482639_2 | EU482639 | 1 | 1038 | 5096 | 6133 | 2 |
| 36 | FJ469908_2 | FJ469908 | 1 | 5585 | 5058 | 10642 | 1 |
| 37 | EF025110_2 | EF025110 | 1 | 1033 | 6072 | 7104 | 1 |
| 38 | GQ868539_2 | GQ868539 | 1 | 3540 | 7124 | 10663 | 1 |
| 39 | FJ410205_2 | FJ410205 | 1 | 2542 | 8135 | 10676 | 1 |
| 40 | EU482481_2 | EU482481 | 1 | 4605 | 6086 | 10690 | 1 |

| Index | Query ID | Subject ID | Q.Start | Q.End | S.Start | S.End | Serotype |
|---|---|---|---|---|---|---|---|
| 41 | FJ226067_3 | FJ226067 | 1 | 2020 | 1 | 2020 | 4 |
| 42 | GQ868585_3 | GQ868585 | 1 | 2031 | 4103 | 6133 | 4 |
| 43 | GQ868594_3 | GQ868594 | 1 | 3478 | 7130 | 10607 | 4 |
| 44 | FJ882598_3 | FJ882598 | 1 | 1010 | 1 | 1010 | 4 |
| 45 | EU854299_3 | EU854299 | 1 | 1027 | 5090 | 6116 | 4 |
| 46 | DQ863638_3 | DQ863638 | 1 | 2020 | 2026 | 4045 | 3 |
| 47 | GQ868578_3 | GQ868578 | 1 | 10585 | 75 | 10659 | 3 |
| 48 | GQ252674_3 | GQ252674 | 1 | 6638 | 4045 | 10682 | 3 |
| 49 | FN429918_3 | FN429918 | 1 | 2023 | 5124 | 7146 | 3 |
| 50 | FJ390371_3 | FJ390371 | 1 | 2565 | 8084 | 10648 | 3 |
| 51 | GQ868604_3 | GQ868604 | 1 | 3029 | 4042 | 7070 | 2 |
| 52 | FJ226066_3 | FJ226066 | 1 | 2032 | 5052 | 7083 | 2 |
| 53 | FN429891_3 | FN429891 | 1 | 6660 | 4064 | 10723 | 2 |
| 54 | AB479042_3 | AB479042 | 1 | 10579 | 69 | 10647 | 2 |
| 55 | EU482639_3 | EU482639 | 1 | 1067 | 2076 | 3142 | 2 |
| 56 | FJ469908_3 | FJ469908 | 1 | 103 | 6102 | 6204 | 1 |
| 57 | EF025110_3 | EF025110 | 1 | 1014 | 1014 | 2027 | 1 |
| 58 | GQ868539_3 | GQ868539 | 1 | 9632 | 1032 | 10663 | 1 |
| 59 | FJ410205_3 | FJ410205 | 1 | 3029 | 1 | 3029 | 1 |
| 60 | EU482481_3 | EU482481 | 1 | 5566 | 5125 | 10690 | 1 |

| Index | Query ID | Subject ID | Q.Start | Q.End | S.Start | S.End | Serotype |
|---|---|---|---|---|---|---|---|
| 61 | FJ226067_4 | FJ226067 | 1 | 6555 | 4052 | 10606 | 4 |
| 62 | GQ868585_4 | GQ868585 | 1 | 1012 | 1013 | 2024 | 4 |
| 63 | GQ868594_4 | GQ868594 | 1 | 2020 | 1 | 2020 | 4 |
| 64 | FJ882598_4 | FJ882598 | 1 | 1010 | 1011 | 2020 | 4 |
| 65 | EU854299_4 | EU854299 | 1 | 2019 | 2023 | 4041 | 4 |
| 66 | DQ863638_4 | DQ863638 | 1 | 3045 | 2061 | 5105 | 3 |
| 67 | GQ868578_4 | GQ868578 | 1 | 1015 | 1018 | 2032 | 3 |
| 68 | GQ252674_4 | GQ252674 | 1 | 1009 | 4041 | 5049 | 3 |
| 69 | FN429918_4 | FN429918 | 1 | 10706 | 1 | 10706 | 3 |
| 70 | FJ390371_4 | FJ390371 | 1 | 9587 | 1062 | 10648 | 3 |
| 71 | GQ868604_4 | GQ868604 | 1 | 10679 | 1 | 10679 | 2 |
| 72 | FJ226066_4 | FJ226066 | 1 | 2019 | 4129 | 6147 | 2 |
| 73 | FN429891_4 | FN429891 | 1 | 10723 | 1 | 10723 | 2 |
| 74 | AB479042_4 | AB479042 | 1 | 10647 | 1 | 10647 | 2 |
| 75 | EU482639_4 | EU482639 | 1 | 1031 | 4126 | 5156 | 2 |
| 76 | FJ469908_4 | FJ469908 | 1 | 10642 | 1 | 10642 | 1 |
| 77 | EF025110_4 | EF025110 | 1 | 1010 | 1 | 1010 | 1 |
| 78 | GQ868539_4 | GQ868539 | 1 | 10663 | 1 | 10663 | 1 |
| 79 | FJ410205_4 | FJ410205 | 1 | 10676 | 1 | 10676 | 1 |
| 80 | EU482481_4 | EU482481 | 1 | 1046 | 4113 | 5158 | 1 |

| Index | Query ID | Subject ID | Q.Start | Q.End | S.Start | S.End | Serotype |
|---|---|---|---|---|---|---|---|
| 81 | FJ226067_5 | FJ226067 | 1 | 3041 | 5059 | 8099 | 4 |
| 82 | GQ868585_5 | GQ868585 | 1 | 3041 | 5059 | 8099 | 4 |
| 83 | GQ868594_5 | GQ868594 | 1 | 2526 | 8082 | 10607 | 4 |
| 84 | FJ882598_5 | FJ882598 | 1 | 2019 | 2023 | 4041 | 4 |
| 85 | EU854299_5 | EU854299 | 1 | 2113 | 1 | 2113 | 4 |
| 86 | DQ863638_5 | DQ863638 | 1 | 5640 | 5068 | 10707 | 3 |
| 87 | GQ868578_5 | GQ868578 | 1 | 5108 | 1 | 5108 | 3 |
| 88 | GQ252674_5 | GQ252674 | 1 | 7616 | 3067 | 10682 | 3 |
| 89 | FN429918_5 | FN429918 | 1 | 8678 | 2029 | 10706 | 3 |
| 90 | FJ390371_5 | FJ390371 | 1 | 4123 | 1 | 4123 | 3 |
| 91 | GQ868604_5 | GQ868604 | 1 | 5560 | 5120 | 10679 | 2 |
| 92 | FJ226066_5 | FJ226066 | 1 | 531 | 10154 | 10684 | 2 |
| 93 | FN429891_5 | FN429891 | 1 | 1630 | 9094 | 10723 | 2 |
| 94 | AB479042_5 | AB479042 | 1 | 2025 | 5053 | 7077 | 2 |
| 95 | EU482639_5 | EU482639 | 1 | 1091 | 1 | 1091 | 2 |
| 96 | FJ469908_5 | FJ469908 | 1 | 2562 | 8081 | 10642 | 1 |
| 97 | EF025110_5 | EF025110 | 1 | 633 | 10103 | 10735 | 1 |
| 98 | GQ868539_5 | GQ868539 | 1 | 2019 | 4043 | 6061 | 1 |
| 99 | FJ410205_5 | FJ410205 | 1 | 1010 | 5051 | 6060 | 1 |
| 100 | EU482481_5 | EU482481 | 1 | 3619 | 7072 | 10690 | 1 |

Table 4.2 illustrates the result of this function. Each row in Table 4.2 is grouped by both serotype (Column 8) and the name of database sequence found to match query sequence (Column 3), respectively. As an example, the fields in the first fifty rows are grouped by serotypes 1 and 2, respectively. Each serotype is grouped by the name of database sequence. The first twenty-five rows is a group of Dengue virus serotype 1. Each row is grouped by the name of database sequence: EF025110, EU482481, FJ410205, FJ469908 and GQ868539, respectively.

**Table 4.2** Result of query sequence sorted by both serotype and database sequence

| Index | Query ID | Subject ID | Q.Start | Q.End | S.Start | S.End | Serotype |
|---|---|---|---|---|---|---|---|
| 17 | EF025110_1 | EF025110 | 1 | 1010 | 1 | 1010 | 1 |
| 37 | EF025110_2 | EF025110 | 1 | 1033 | 6072 | 7104 | 1 |
| 57 | EF025110_3 | EF025110 | 1 | 1014 | 1014 | 2027 | 1 |
| 77 | EF025110_4 | EF025110 | 1 | 1010 | 1 | 1010 | 1 |
| 97 | EF025110_5 | EF025110 | 1 | 633 | 10103 | 10735 | 1 |
| 20 | EU482481_1 | EU482481 | 1 | 591 | 10100 | 10690 | 1 |
| 40 | EU482481_2 | EU482481 | 1 | 4605 | 6086 | 10690 | 1 |
| 60 | EU482481_3 | EU482481 | 1 | 5566 | 5125 | 10690 | 1 |
| 80 | EU482481_4 | EU482481 | 1 | 1046 | 4113 | 5158 | 1 |
| 100 | EU482481_5 | EU482481 | 1 | 3619 | 7072 | 10690 | 1 |
| 19 | FJ410205_1 | FJ410205 | 1 | 5616 | 5061 | 10676 | 1 |
| 39 | FJ410205_2 | FJ410205 | 1 | 2542 | 8135 | 10676 | 1 |
| 59 | FJ410205_3 | FJ410205 | 1 | 3029 | 1 | 3029 | 1 |
| 79 | FJ410205_4 | FJ410205 | 1 | 10676 | 1 | 10676 | 1 |
| 99 | FJ410205_5 | FJ410205 | 1 | 1010 | 5051 | 6060 | 1 |
| 16 | FJ469908_1 | FJ469908 | 1 | 3068 | 4085 | 7152 | 1 |
| 36 | FJ469908_2 | FJ469908 | 1 | 5585 | 5058 | 10642 | 1 |
| 56 | FJ469908_3 | FJ469908 | 1 | 103 | 6102 | 6204 | 1 |
| 76 | FJ469908_4 | FJ469908 | 1 | 10642 | 1 | 10642 | 1 |
| 96 | FJ469908_5 | FJ469908 | 1 | 2562 | 8081 | 10642 | 1 |
| 18 | GQ868539_1 | GQ868539 | 1 | 552 | 10112 | 10663 | 1 |
| 38 | GQ868539_2 | GQ868539 | 1 | 3540 | 7124 | 10663 | 1 |
| 58 | GQ868539_3 | GQ868539 | 1 | 9632 | 1032 | 10663 | 1 |
| 78 | GQ868539_4 | GQ868539 | 1 | 10663 | 1 | 10663 | 1 |
| 98 | GQ868539_5 | GQ868539 | 1 | 2019 | 4043 | 6061 | 1 |

| Index | Query ID | Subject ID | Q.Start | Q.End | S.Start | S.End | Serotype |
|---|---|---|---|---|---|---|---|
| 14 | AB479042_1 | AB479042 | 1 | 2066 | 5053 | 7118 | 2 |
| 34 | AB479042_2 | AB479042 | 1 | 3056 | 3045 | 6100 | 2 |
| 54 | AB479042_3 | AB479042 | 1 | 10579 | 69 | 10647 | 2 |
| 74 | AB479042_4 | AB479042 | 1 | 10647 | 1 | 10647 | 2 |
| 94 | AB479042_5 | AB479042 | 1 | 2025 | 5053 | 7077 | 2 |
| 15 | EU482639_1 | EU482639 | 1 | 9600 | 1080 | 10679 | 2 |
| 35 | EU482639_2 | EU482639 | 1 | 1038 | 5096 | 6133 | 2 |
| 55 | EU482639_3 | EU482639 | 1 | 1067 | 2076 | 3142 | 2 |
| 75 | EU482639_4 | EU482639 | 1 | 1031 | 4126 | 5156 | 2 |
| 95 | EU482639_5 | EU482639 | 1 | 1091 | 1 | 1091 | 2 |
| 12 | FJ226066_1 | FJ226066 | 1 | 1010 | 5054 | 6063 | 2 |
| 32 | FJ226066_2 | FJ226066 | 1 | 3029 | 1019 | 4047 | 2 |
| 52 | FJ226066_3 | FJ226066 | 1 | 2032 | 5052 | 7083 | 2 |
| 72 | FJ226066_4 | FJ226066 | 1 | 2019 | 4129 | 6147 | 2 |
| 92 | FJ226066_5 | FJ226066 | 1 | 531 | 10154 | 10684 | 2 |
| 13 | FN429891_1 | FN429891 | 1 | 106 | 8095 | 8200 | 2 |
| 33 | FN429891_2 | FN429891 | 1 | 1020 | 5057 | 6076 | 2 |
| 53 | FN429891_3 | FN429891 | 1 | 6660 | 4064 | 10723 | 2 |
| 73 | FN429891_4 | FN429891 | 1 | 10723 | 1 | 10723 | 2 |
| 93 | FN429891_5 | FN429891 | 1 | 1630 | 9094 | 10723 | 2 |
| 11 | GQ868604_1 | GQ868604 | 1 | 1010 | 1 | 1010 | 2 |
| 31 | GQ868604_2 | GQ868604 | 1 | 3029 | 1015 | 4043 | 2 |
| 51 | GQ868604_3 | GQ868604 | 1 | 3029 | 4042 | 7070 | 2 |
| 71 | GQ868604_4 | GQ868604 | 1 | 10679 | 1 | 10679 | 2 |
| 91 | GQ868604_5 | GQ868604 | 1 | 5560 | 5120 | 10679 | 2 |

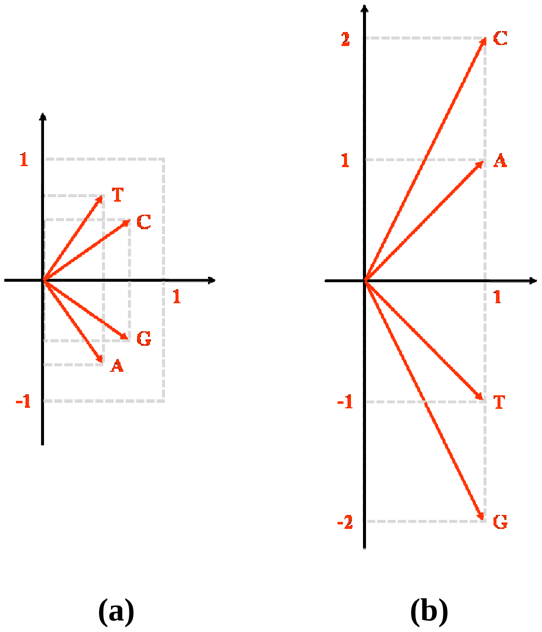
| Index | Query ID | Subject ID | Q.Start | Q.End | S.Start | S.End | Serotype |
|---|---|---|---|---|---|---|---|
| 6 | DQ863638_1 | DQ863638 | 1 | 3240 | 4081 | 7320 | 3 |
| 26 | DQ863638_2 | DQ863638 | 1 | 2019 | 2024 | 4042 | 3 |
| 46 | DQ863638_3 | DQ863638 | 1 | 2020 | 2026 | 4045 | 3 |
| 66 | DQ863638_4 | DQ863638 | 1 | 3045 | 2061 | 5105 | 3 |
| 86 | DQ863638_5 | DQ863638 | 1 | 5640 | 5068 | 10707 | 3 |
| 10 | FJ390371_1 | FJ390371 | 1 | 502 | 10147 | 10648 | 3 |
| 30 | FJ390371_2 | FJ390371 | 1 | 4586 | 6063 | 10648 | 3 |
| 50 | FJ390371_3 | FJ390371 | 1 | 2565 | 8084 | 10648 | 3 |
| 70 | FJ390371_4 | FJ390371 | 1 | 9587 | 1062 | 10648 | 3 |
| 90 | FJ390371_5 | FJ390371 | 1 | 4123 | 1 | 4123 | 3 |
| 9 | FN429918_1 | FN429918 | 1 | 4075 | 3065 | 7139 | 3 |
| 29 | FN429918_2 | FN429918 | 1 | 2025 | 4105 | 6129 | 3 |
| 49 | FN429918_3 | FN429918 | 1 | 2023 | 5124 | 7146 | 3 |
| 69 | FN429918_4 | FN429918 | 1 | 10706 | 1 | 10706 | 3 |
| 89 | FN429918_5 | FN429918 | 1 | 8678 | 2029 | 10706 | 3 |
| 8 | GQ252674_1 | GQ252674 | 1 | 6594 | 4089 | 10682 | 3 |
| 28 | GQ252674_2 | GQ252674 | 1 | 2019 | 3033 | 5051 | 3 |
| 48 | GQ252674_3 | GQ252674 | 1 | 6638 | 4045 | 10682 | 3 |
| 68 | GQ252674_4 | GQ252674 | 1 | 1009 | 4041 | 5049 | 3 |
| 88 | GQ252674_5 | GQ252674 | 1 | 7616 | 3067 | 10682 | 3 |
| 7 | GQ868578_1 | GQ868578 | 1 | 10659 | 1 | 10659 | 3 |
| 27 | GQ868578_2 | GQ868578 | 1 | 3028 | 2026 | 5053 | 3 |
| 47 | GQ868578_3 | GQ868578 | 1 | 10585 | 75 | 10659 | 3 |
| 67 | GQ868578_4 | GQ868578 | 1 | 1015 | 1018 | 2032 | 3 |
| 87 | GQ868578_5 | GQ868578 | 1 | 5108 | 1 | 5108 | 3 |

| Index | Query ID | Subject ID | Q.Start | Q.End | S.Start | S.End | Serotype |
|---|---|---|---|---|---|---|---|
| 5 | EU854299_1 | EU854299 | 1 | 10606 | 1 | 10606 | 4 |
| 25 | EU854299_2 | EU854299 | 1 | 1017 | 1018 | 2034 | 4 |
| 45 | EU854299_3 | EU854299 | 1 | 1027 | 5090 | 6116 | 4 |
| 65 | EU854299_4 | EU854299 | 1 | 2019 | 2023 | 4041 | 4 |
| 85 | EU854299_5 | EU854299 | 1 | 2113 | 1 | 2113 | 4 |
| 1 | FJ226067_1 | FJ226067 | 1 | 4039 | 1 | 4039 | 4 |
| 21 | FJ226067_2 | FJ226067 | 1 | 1043 | 8102 | 9144 | 4 |
| 41 | FJ226067_3 | FJ226067 | 1 | 2020 | 1 | 2020 | 4 |
| 61 | FJ226067_4 | FJ226067 | 1 | 6555 | 4052 | 10606 | 4 |
| 81 | FJ226067_5 | FJ226067 | 1 | 10539 | 68 | 10606 | 4 |
| 4 | FJ882598_1 | FJ882598 | 1 | 10606 | 1 | 10606 | 4 |
| 24 | FJ882598_2 | FJ882598 | 1 | 3047 | 1 | 3047 | 4 |
| 44 | FJ882598_3 | FJ882598 | 1 | 1010 | 1 | 1010 | 4 |
| 64 | FJ882598_4 | FJ882598 | 1 | 1010 | 1011 | 2020 | 4 |
| 84 | FJ882598_5 | FJ882598 | 1 | 2019 | 2023 | 4041 | 4 |
| 2 | GQ868585_1 | GQ868585 | 1 | 3505 | 7102 | 10606 | 4 |
| 22 | GQ868585_2 | GQ868585 | 1 | 1010 | 1 | 1010 | 4 |
| 42 | GQ868585_3 | GQ868585 | 1 | 2031 | 4103 | 6133 | 4 |
| 62 | GQ868585_4 | GQ868585 | 1 | 1012 | 1013 | 2024 | 4 |
| 82 | GQ868585_5 | GQ868585 | 1 | 3041 | 5059 | 8099 | 4 |
| 3 | GQ868594_1 | GQ868594 | 1 | 3071 | 4049 | 7119 | 4 |
| 23 | GQ868594_2 | GQ868594 | 1 | 503 | 10105 | 10607 | 4 |
| 43 | GQ868594_3 | GQ868594 | 1 | 3478 | 7130 | 10607 | 4 |
| 63 | GQ868594_4 | GQ868594 | 1 | 2020 | 1 | 2020 | 4 |
| 83 | GQ868594_5 | GQ868594 | 1 | 2526 | 8082 | 10607 | 4 |

# CHAPTER V
# DISCUSSION

## 5.1 Comparison of UnitX representative with Yau's method

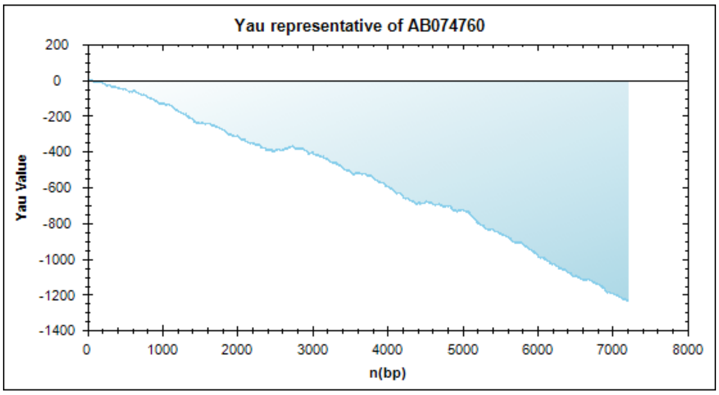
Fig. 5.1 (a) is the unit vector uesd to represent four-letter alphabet (nucleotides A, Y, C and G) proposed by Yau et al [14]. The idea of Yau's method is to convert nucleotide sequence to a sequence graph. By comparing with existing visualization tool in the literature, Yau's method provides an extensive picture of this data and has low computational complexity. By these merit, we propose a more suitable representative (see Fig. 5.1 (b)) to represent nucleotide sequence of Dengue virus, called UnitX.



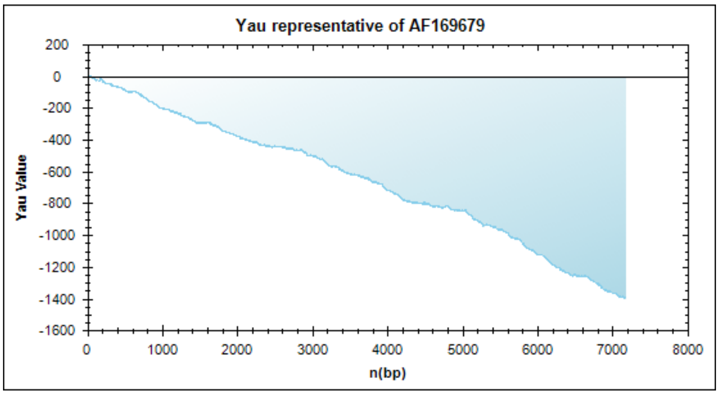
(a) (b)

**Figure 5.1** (a) The unit vector designed by Yau (b) The unit vector of UnitX

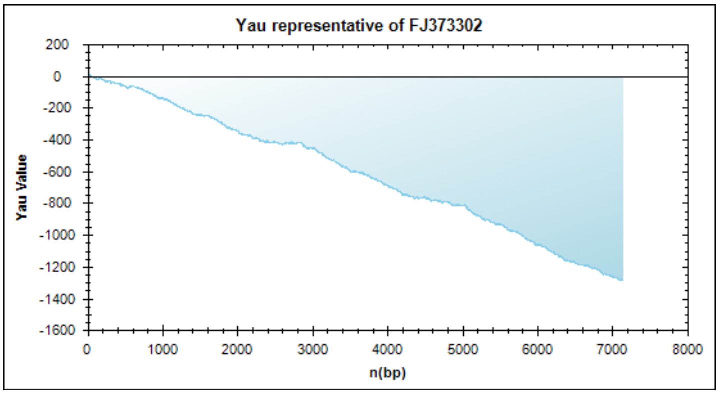
The following (see Figs. 5.2 and 5.3) is representative of all four serotype of Dengue virus. Figs. 5.2 (a)-(d) show Yau's representative of four Dengue virus serotypes. The x and y axes of the graph corresponds to the cumulative amount of pyrimidine and purine bases (a measurement value).


Yau representative of AB074760
Yau Value
200
0
-200
-400
-600
-800
-1000
-1200
-1400
0
1000
2000
3000
4000
5000
6000
7000
8000
n(bp)

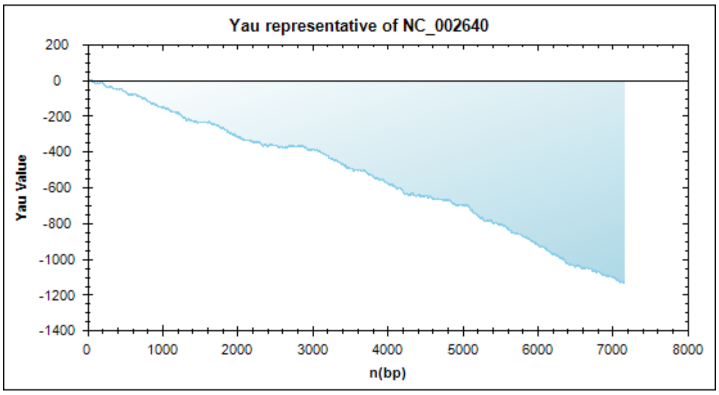
**(a)**


Yau representative of AF169679
Yau Value
200
0
-200
-400
-600
-800
-1000
-1200
-1400
-1600
0
1000
2000
3000
4000
5000
6000
7000
8000
n(bp)

**(b)**


Yau representative of FJ373302
Yau Value
200
0
-200
-400
-600
-800
-1000
-1200
-1400
-1600
0
1000
2000
3000
4000
5000
6000
7000
8000
n(bp)

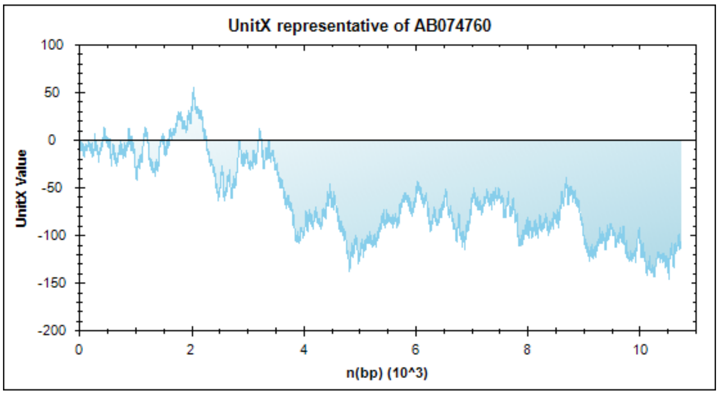
**(c)**



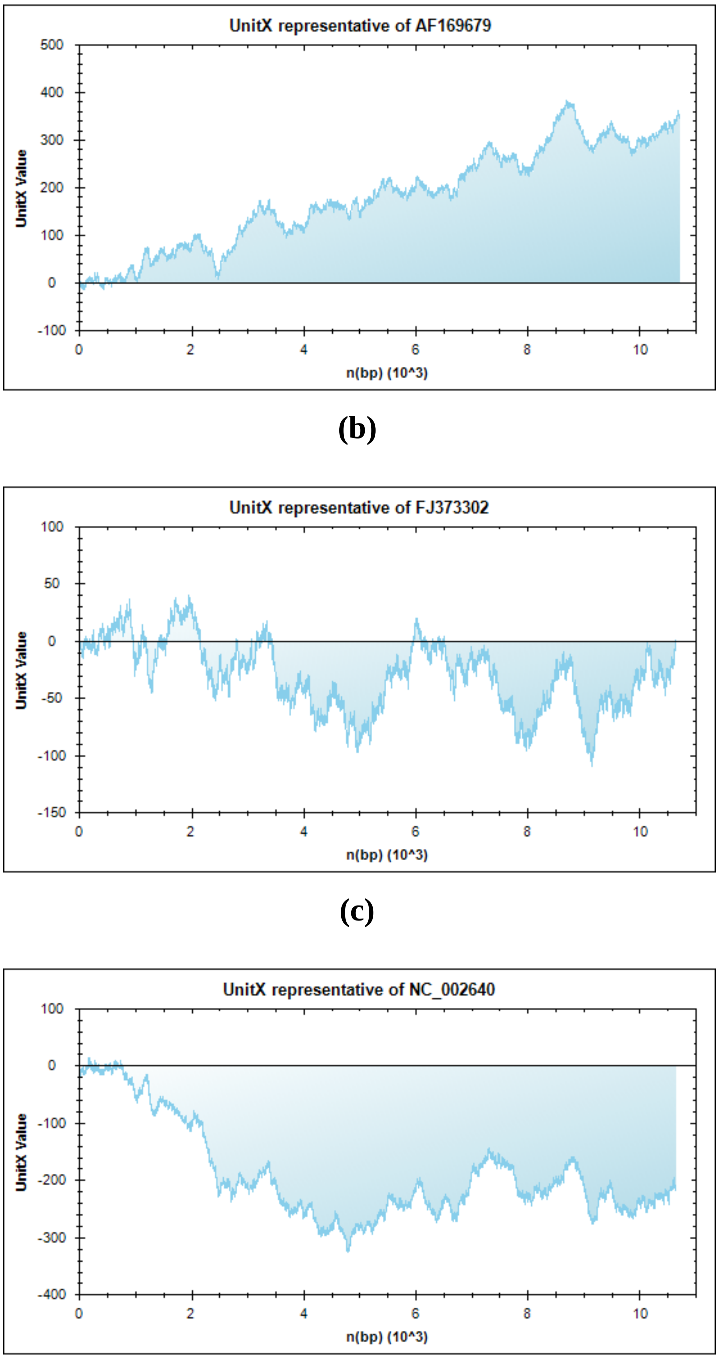
**(d)**

**Figure 5.2** Yau representative of four Dengue virus serotypes:
(a) DENV-1: AB074760, (b) DENV-2: AF169679,
(c) DENV-3: FJ373302 and (d) DENV-4: NC_002640

Figs. 5.3 (a)-(d) show UnitX representative of four Dengue virus serotypes. The x-axis of the graph corresponds to the base pair position along nucleotide sequence and the y-axis corresponds to the cumulative amount of amino and keto bases (a measurement value).



**(a)**



(b)

(c)

(d)

**Figure 5.3** UnitX representative of four Dengue virus serotypes: (a) DENV-1: AB074760, (b) DENV-2: AF169679, (c) DENV-3: FJ373302 and (d) DENV-4: NC_002640

According to Figs 5.2 and 5.3, it is clear that Yau' s method is difficult to visually distinguish all serotypes of Dengue virus (DENV-1 to DENV-4). Furthermore, the data plotted on the x-axis are not the integer numbers (i.e. values on the x-axis are not directly equal to the number of bases) so that the result is difficult to interpret compared to the convention base-by-base basic (e.g. BLAST). By using UnitX representative, we can visually distinguish all four Dengue virus serotypes. As a result, UnitX is a well designed representative for visualization nucleotide sequence of Dengue virus.

## 5.2 Comparison of the optimal library created by UnitX method and multiple sequence alignment technique

In this section, we aim to discuss the optimal library created by the proposed UnitX representation and the multiple sequence alignment technique. In this experiment, we use 2,184 sample sequences of all four Dengue virus serotypes each serotype contains 952, 737, 405 and 90 sequences, respectively. According to Tables 3.1 and 3.2, the ability to classify all four serotypes of Dengue virus of UnitX method is 98.86 % and multiple sequence alignment technique is 100 %. In this thesis, the optimal library represented each serotype is created by multiple sequence alignment technique. There are two major reasons to use multiple sequence alignment technique:

**(1)** The optimal library created by this technique result in high accuracy and reliability because of using ten percent of each serotype for training.

**(2)** This technique is extension of pairwise alignment to incorporate more than two sequences at a time.

## 5.3 Discussion on *CADVIST*

In bioinformatics, there are numerous tools for sequence analysis. The idea of this thesis is to take the advantage of the unique ability of visual perception to detect meaningful patterns that may otherwise remain hidden. The goal of this thesis is database searching and identification of homology. For database search tool, both time and accuracy are necessary to get the best quality results. Basic Local Alignment

Search Tool (BLAST) is one of the most widely used tools for sequence similarity search due to its speed and reasonable accuracy of search performance. BLAST can be used via web interface or as a standalone tool. By its merit, this thesis employs standalone BLAST+ application to perform BLAST search, the essential information from BLAST results as an input for further analysis. In this thesis, we employ this information to propose the software, called CADViST, to visualize and analyze nucleotide sequence of Dengue virus. CADViST consists of three main functions: (1) Dengue virus search engine, (2) UnitX visualization and analysis and (3) UnitX visualization referred to the user defined references. The merit of each function is described in more detail below.

### 5.3.1 Discussion on Dengue virus search engine

Dengue virus search engine is used to perform local similarity search engine. This function employs the results in textual format from BLAST+ application to provide a simplified method of sequence similarity search between query sequence and sequence in database. After that UnitX is employed to represent the result of BLAST in graphic form. Using standalone BLAST+ application, this function can perform database search without the cost or training. In other word, this function of CADViST is a visualization tool to aid the interpretation of BLAST search results.

### 5.3.2 Discussion on UnitX visualization and analysis

UnitX visualization and analysis is used to automatically assign the serotype to the unknown Dengue virus sequence. The idea of this function is construction the optimal sequence alignment represented each Dengue virus serotype. BLAST is a simplified method of sequence similarity search between query sequence and sequence in database. In standalone mode, BLAST is able to create your own BLAST database. By its merit, we employ standalone BLAST+ application to find the best match between the unknown Dengue virus sequence and the optimal sequence alignment represented each Dengue virus serotype. The database sequence of this function is the optimal sequence alignment represented each Dengue virus serotype. After that, this function employs UnitX to represent the result of BLAST+ application in graphic form so that this function provides a simple way to visualize and analyze the unknown Dengue virus sequence. In other word, this function of CADVIST employ standalone BLAST+ application as a tool to predict the serotype of the unknown Dengue virus sequence. Using the optimal sequence alignment as database sequences, this function provides the result in high accuracy and reliability.

### 5.3.3 Discussion on UnitX visualization referred to the user defined references

UnitX visualization referred to the user defined references used to display the results in textual format from BLAST+ application. This function requires three components: (1) the result in "Tabular" format from BLAST+ application, (2) - (3) query and database sequences. The merit of this function is users can select their own sequence database so that user can compare samples to the desired reference sequence. This function employs UnitX to create the BLAST results in a graphical form. By employing UnitX, this function is useful for users to get a quick overview of search results. According to web interface for BLAST at NCBI, graphical overview of BLAST results (see Fig. 2.29) are rarely complete descriptors because of a limited space to display information and tends to pack results into a small visible area. In other word, this function of CADVIST provides a clear picture to aid the interpretation of BLAST search results.

# CHAPTER VI
# CONCLUSIONS AND FUTURE DIRECTIONS

## 6.1 Conclusions

This thesis presents the prototype software, called CADVIST, to visualize and analyze nucleotide sequence of Dengue virus. CADVIST extract the essential information from BLAST output to visualize and analyze via graphic display. CADVIST consists of three main functions: (1) Dengue virus search engine, (2) UnitX visualization and analysis and (3) UnitX visualization referred to the user defined references. First, we have presented Dengue virus search engine tool to locate partial matches on sequence data. Second, we have presented visualization and analysis tool for nucleotide sequence of Dengue virus. Finally, we have proposed visualization tool to display the Tabular output from BLAST+ application that the user can define their reference database. The result of this function is sorted by both serotype and database sequence found to match query sequence. All of these function display nucleotide sequence of Dengue virus with the proposed representative, called UntiX. CADVIST is readily useful for all research areas that require BLAST functions. One advantage is to provide an extensive picture of the results by viewing in a computer screen, regardless of how long these sequences are. Besides, many options in CADVIST can also benefit the bioinformatics experts, e.g. save, print, show the raw numeric values on the graph, and report in Microsoft Office Excel 2007 file format. With the .net framework of C#, CADVIST can be easily modified to include more open source or in house developed mathematical modeling, while maintaining the user friendly GUI.

## 6.2 Future directions

CADVIST is prototype software that suitable for visualization and analysis nucleotide sequence of Dengue virus. In order to improve CADVIST, the major future directions can be listed as follows:

(1) In this thesis, we designed and implemented CADVIST using C# programming language. C# is Modern Object-Oriented Programming language that makes it easier to create multithreaded code. In term of efficiency, it would be interesting to implement CADVIST using multithreaded programming.

(2) CADVIST can be employed to enhance epidemiology as Fig. 6.1. The present version of CADVIST is developed for a standalone computer. We plan to develop CADVIST as web interface for joining and changing epidemic data between countries. Besides that CADVIST can also be used as the part of Dengue virus real-time monitoring to analyze and inform the tendency of Dengue virus in each area. Genetic data is illustrated through CADVIST so that sending serum sample is not necessary.

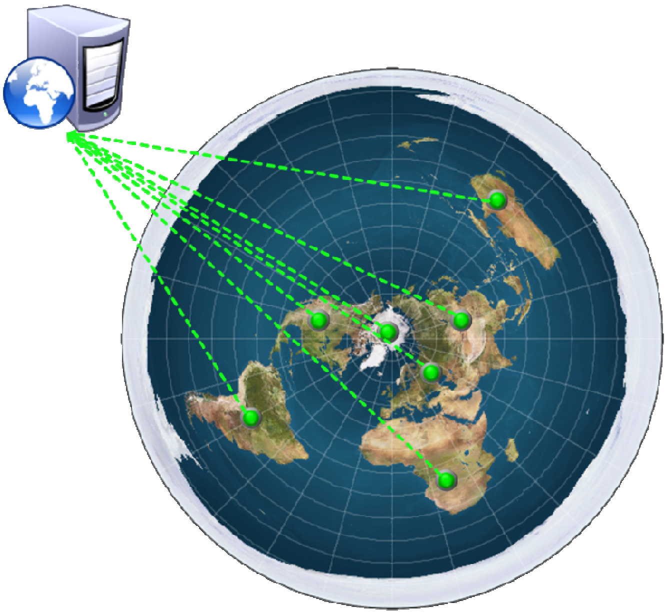


**Figure 6.1** The future idea of CADVIST in web-based interface

# REFERENCES

1 Swarna Bai Arniker and Hon Keung Kwan (2009). "Graphical Representation of DNA sequences." EIT 2009: 311-314.

2 Sukathida Ubol and Chantapong Wasi (2006). "Dengue Heorrhagic Fever, Virology, Immunopathogenesis, Diagnosis, Treatment, Vaccine, Prevention and Control." The Virology Association (Thailand), Bangkok, Thailand.

3 Alex Tang "Clinical spectrum of Dengue infection." This information is available from: http://www.kairos2.com/79_Dengue%20and%20Dengue%20shock %20syndrome.pdf.

4 Michael B. Nathan "Dengue in the Western Pacific Region." This information is available from: http://www.wpro.who.int/health_topics/dengue/.

5 Sansanee Noisakran and Guey Chuen Perng (2008). "MINIREVIEW: Alternate Hypothesis on the Pathogenesis of Dengue Hemorrhagic Fever (DHF)/Dengue Shock Syndrome (DSS) in Dengue Virus Infection." The Society for Experimental Biology and Medicine: 401-408.

6 Eugene Hamori and John Ruskin (1982). "H curves, a novel method of representation of nucleotide series especially suited for long DNA sequences." The Journal of biological chemistry 258(2): 1318-1327.

7 Ren Zhang and Chun-Ting Zhang (1994). "Z curves, an intutive tool for visualizing and analyzing the DNA sequences." Journal of biomolecular structure & dynamics 11(4): 767-782.

8 Chun-Ting Zhang, Ju Wang and Ren Zhang (2001). "A novel method to calculate the G+C content of genomic DNA sequences." Journal of biomolecular structure & dynamics 19(2): 333-341.

9 The Center of Bioinformatics, Tianjin University, China (TUBIC) "Introduction to the Z curve." This information is available from: http://tubic.tju.edu.cn/zcurve/.

10 James J Cai, David K Smith, Xuhua Xia and Kwok-yung Yuen (2005). "MBEToolbox: a Matlab toolbox for sequence data analysis in molecular biology and evolution." BMC Bioinformatics 6(64).

11 Liu Xikui and Li Yan (2009). "Graphical Representation of DNA Sequence with Degeneracy Tends to Zero." The 3rd International Conference on Bioinformatics and Biomedical Engineering (iCBBE 2009).

12 Michael A. Gates (1986). "A simple way to look at DNA." Journal of theoretical biology 119(3): 319-328.

13 A. Nandy (1996). "Two-dimensional graphical representation of DNA sequences and intron-exon discrimination in intron-rich sequences."  12(1): 55-62.

14 Stephen S. -T. Yau, Jiasong Wang, Amir Niknejad, Chaoxiao Lu, Ning Jin and Yee-Kin Ho (2003). "DNA sequence representation without degeneracy" Nucleic Acids Research 31(12): 3078-3080.

15 Altschul Stephen F., Warren Gish, Webb Miller, Eugene W. Myers and David J. Lipman (1990). "Basic local alignment search tool." Journal of Molecular Biology 215: 403-410.

16 Law Ngai-Fong, Cheng Kin-On and Siu Wan-Chi (2006). "On relationship of Z-curve and Fourier approaches for DNA coding sequence classification." Bioinformation 1(7): 242-246.

17 Dimitris Anastassiou (2001). "Genomic Signal Processing." IEEE Signal Processing Magazine 18(4): 8-20.

18 Daniel Huson (2009). "BLAST and BLAT." Lecture note: Bioinformatics I:
This article is available from: http://www-ab.informatik.uni-tuebingen.de/
teaching/ws09/bioinformatics-i/04-blast-blat-fasta.pdf.

19 Julie D.Thompson, Desmond G.Higgins and Toby J.Gibson (1994). "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice." Nucleic Acids Research 22(22): 4673-4680.

20 Tom Madden (2003). "The BLAST sequence analysis tool." The NCBI handbook:
This article is available from http://www.mr-ideahamster.com/
population_genetics/bioinformatics/blast-ncbi.pdf.

21 Qi Yutao and Lin Feng (2003). "CyberparaBLAST: the Parallelized BLAST Web Server." Second International Conference on Cyberworlds (CW'03): 474.

22 Paul Fawcett and Jeff Elhai (2009). "BNFO 601: Integrated Bioinformatics What's wrong with BlastN?" This article is available from: http://www.vcu.edu/ csbc/bnfo601/Scenarios/Blast/WhatsWrongWithBlastN.pdf.

23 NCBI - National Center for Biotechnology Information (2010). "What is "low-complexity" sequence?" This information is available from: http://www.ncbi.nlm.nih.gov/blast/Blast.cgi?CMD=Web&PAGE_TYPE= BlastDocs&DOC_TYPE=FAQ.

24 Crossroads (2006). "A Survey of BLAST Software." This information is available from: http://www.acm.org/crossroads/wikifiles/13-11-R/13-11-10.pdf.

25 Joanne Fox, Michael Smith Laboratories and UBC (2007). "Lecture/Lab: BLAST" http://bioteach.ubc.ca/FILES/asmcue2009/BLAST-background-notes.pdf.

26 NCBI - National Center for Biotechnology Information (2010). "The Statistics of Sequence Similarity Scores." This article is available from: http://www.ncbi.nlm.nih.gov/BLAST/tutorial/Altschul-1.html#head2.

27 The U.S. Department of Energy (DOE) Human Genome Project Information Web site "Understanding BLAST results." This information is available from: http://www.ornl.gov/sci/techresources/Human_Genome/posters/chromosome/blast.shtml.

28 This information is available from: http://blast.ncbi.nlm.nih.gov/Blast.cgi.

29 Jesse Wolfgang (2004). "Lecture note - Trees, Stars, and Multiple Biological Sequence Alignment." This information is available from: http://www.cse.lehigh.edu/~lopresti/Courses/2003-2004/CSE2397-2497/.

30 Naruya Saitou and Masatoshi Nei (1987). "The Neighbor-joining Method: A New Method for Reconstructing Phylogenetic Trees." Molecular Biology and Evolution 4(4): 406-425.

31 Michael Brudno (2007). "Lecture Notes - Phylogenetic Trees." This information is available from: http://www.cs.toronto.edu/~brudno/bcb410/phylotrees.pdf.

32 Cédric Notredame "Recent Progress in Multiple Sequence Alignments: A Survey." This information is available from: http://genome.nci.nih.gov/talks/msa/.

33 Eric Rouchka "Lecture notes - Dynamic Programming." This information is available from: http://sce.uhcl.edu/boetticher/ML_DataMining/ DynamicProgramming.PDF.

34 Boonyarat Viriyasaksathian, Yodchanan Wongsawat and Prapat Suriyapol (2009.). "UnitX: Dengue Virus Sequence Graphical Representation for Serotypes Classification." International Symposium on Biomedical Engineering (ISBME), Bangkok, Thailand

35 Boonyarat Viriyasaksathian, Yodchanan Wongsawat and Prapat Suriyapol (2010). "CADViST: Visualization Tool for BLAST Alignment of Dengue Virus Sequences." The 4th International Conference on Bioinformatics and Biomedical Engineering (iCBBE 2010).

# APPENDICES

```
>GQ199895 Dengue virus 2 isolate DENV-2/NI/BID-V2683/1999, complete genome
ACAAAGACAGATTCTTTGAGGGAGCTAAGCTCAACGTAGTTCTAACAGTTTTTAATTA
GAGAGCAGATCTCTGATGAATAACCAACGGAAAAAGGCGAGAAGTACGCCTTTCAAT
ATGCTGAAACGCGAGAGAAACCGCGTGTCAACTGTGCAACAGCTGACAAAGAGATTC
TCACTTGGAATGCTGCAAGGACGCGGACCATTAAAACTGTTCATGGCCCTTGTGGCGT
TCCTTCGTTTCCTAACAATCCCACCAACAGCAGGGATACTAAAAAGATGGGGAACGAT
CAAAAAATCAAAAGCTATCAATGTTTTGAGAGGGTTCAGGAAAGAGATTGGAAGGAT
GCTGAACATCTTGAACAGGAGACGCAGGACAGCAGGCGTGATTGTTATGTTGATTCCA
ACAGCGATGGCGTTCCATTTAACCACACGCAATGGAGAACCACACATGATCGTTGGTA
GGCAGGAGAAAGGGAAAAGTCTTCTGTTCAAAACAGAGGATGGTGTTAACATGTGTA
CCCTCATGGCCATAGACCTTGGTGAATTGTGTGAAGATACAATCACGTACAAGTGTCC
TCTCCTCAGACAAAATGAACCAGAAGACATAGATTGTTGGTGCAACTCTACGTCCACA
TGGGTAACTTATGGGACATGTACCACCACAGGAGAACACAGAAGAGAAAAAAGATCA
GTGGCGCTCGTTCCACATGTTGGTATGGGACTGGAGACACGAACTGAAACATGGATGT
CATCAGAAGGGGCCTGGAAACATGTTCAGAGAATTGAAACCTGGATCTTGAGACATC
CAGGCTTTACCATAATGGCAGCAATCCTGGCATACACCATAGGAACGACACATTTCCA
AAGGGCCTTGATTTTCATCTTACTGACAGCTGTCGCTCCTTCAATGACAATGCGCTGCA
TAGGAATATCAAATAGAGACTTCGTAGAAGGGGTTTCAGGAGGAAGCTGGGTTGACA
TAGTCTTAGAACATGGAAGTTGTGTGACGACGATGGCAAAAAACAAACCAACATTGG
ATTTTGAACTGATAAAAACAGAAGCCAAACAACCTGCCACTCTAAGGAAGTACTGTAT
AGAAGCAAAGCTGACCAACACAACAACAGAATCGCGTTGCCCAACACAAGGGGAACC
CAGTCTAAATGAAGAGCAGGACAAAAGGTTCATCTGCAAACACTCCATGGTAGACAG
AGGATGGGGAAATGGATGTGGATTATTTGGAAAGGGAGGCATTGTGACCTGTGCTAT
GTTTACATGCAAAAAGAACATGGAAGGAAAAGTCGTGCAGCCAGAAAATTTGGAATA
CACCATCGTGATAACACCTCACTCAGGAGAAGAGCACGCTGTAGGTAATGACACAGG
AAAGCATGGCAAGGAAATCAAAATAACACCACAGAGTTCCATCACAGAAGCAGAACT
GACAGGCTATGGCACTGTCACGATGGAGTGCTCTCCGAGAACGGGCCTCGACTTCAAT
GAGATGGTGCTGCTGCAGATGGAAGACAAAGCTTGGCTGGTGCACAGGCAATGGTTC
CTAGACCTGCCGTTACCATGGCTACCCGGAGCGGACACACAAGGATCAAATTGGATA
CAGAAAGAGACATTGGTCACTTTCAAAAATCCCCACGCGAAGAAACAGGATGTCGTT
GTTTTAGGGTCTCAAGAAGGGGCCATGCACACGGCACTCACAGGGGCCACAGAAATC
CAGATGTCATCAGGAAACTTACTGTTCACAGGACATCTCAAGTGCAGGCTGAGAATGG
ACAAACTACAGCTCAAAGGAATGTCATACTCTATGTGTACAGGAAAGTTTAAAATTGT
GAAGGAAATAGCAGAAACACAACATGGAACAATAGTTATCAGAGTACAATATGAAGG
GGACGGTTCTCCATGTAAGATCCCTTTTGAGATAACAGATTTGGAAAAAAGACACGTC
TTAGGTCGCTTGATTACAGTTAACCCAATCGTAACAGAAAAAGATAGCCCAGTCAACA
TAGAAGCAGAACCTCCATTCGGAGACAGCTACATCATCATAGGAGTAGAGCCGGGAC
AATTGAAACTCAATTGGTTTAAGAAGGGAAGTTCCATCGGCCAAATGTTTGAGACAAC
AATGAGAGGAGCAAAGAGAATGGCCATTTTAGGTGACACAGCCTGGGACTTTGGATC
CCTGGGAGGAGTGTTTACATCTATAGGAAAGGCTCTCCACCAAGTTTTCGGAGCAATC
TATGGGGCTGCTTTTAGTGGGGTCTCATGGACTATGAAAATCCTCATAGGAGTCATCA
TCACATGGATAGGAATGAATTCACGTAGCACCTCACTGTCTGTGTCGCTAGTATTGGT
GGGCGTCGTGACACTGTACCTGGGAGCTATGGTGCAAGCTGATAGTGGTTGCGTTGTG
AGCTGGAAAAATAAAGAACTGAAATGTGGCAGCGGGATCTTCATCACAGATAACGTA
CACACATGGACAGAACAATATAAGTTCCAACCAGAATCCCCTTCAAAACTAGCTTCAG
CTATCCAAAAAGCTCATGAAGAGGGCATTTGTGGAATCCGCTCAGTAACAAGATTGG
AGAATCTGATGTGGAAACAAATAACACCAGAATTGAATCATATTCTATCAGAAAATG
AGGTAAAGTTGACCATTATGACAGGAGACATTAAAGGAATCATGCAGGCAGGAAAAC
GATCCTTGCGGCCCCAGCCCACTGAGCTGAAGTACTCATGGAAAACATGGGGAAAGG
CGAAAATGCTCTCCACAGAGTCTCACAATCAGACCTTTCTTATTGATGGCCCTGAAAC
AGCAGAATGCCCCAACACAAACAGAGCTTGGAACTCGCTGGAAGTTGAAGACTATGG
TTTTGGAGTTTTCACCACCAATATATGGCTGAAATTGAGAGAAAAACAGGATGTATTT
TGTGACTCAAAACTCATGTCAGCGGCCATTAAAGACAACAGAGCCGTTCATGCCGATA
TGGGTTATTGGATAGAAAGTGCACTCAATGACACATGGAAGATGGAGAAAGCCTCCT
TCATTGAAGTTAAAAGCTGCCACTGGCCAAAGTCACACACCCTTTGGAGCAATGGAGT
```

```
ATTAGAAAGTGAGATGATAATCCCAAAAAATTTTGCCGGGCCAGTGTCACAACACAA
CTACAGACCAGGCTACCATACACAAACAGCAGGACCTTGGCATCTAGGTAAGCTTGA
GATGGACTTTGATCTCTGCGAAGGAACTACAGTGGTGGTGACTGAGGACTGTGGAAAT
AGAGGACCCTCTTTAAGAACGACCACTGCCTCTGGAAAACTCATAACAGAATGGTGCT
GCCGATCCTGCACACTACCACCTCTAAGATACAGAGGTGAGGATGGATGCTGGTACG
GGATGGAAATCAGACCATTGAAAGAGAAAGAGGAGAATTTGGTTAACTCCTTGGTCA
CAGCCGGACATGGGCAGATTGACAACTTTTCACTAGGAGTCTTGGGAATGGCACTGTT
CCTGGAAGAAATGCTCAGGACCCGAATAGGAACGAAACATGCAATACTGCTAGTTGC
AGTATCTTTTGTGACATTGATTACTGGGAACATGTCTTTTAGAGACCTGGGAAGAGTG
ATGGTTATGGTGGGCGCTACCATGACGGATGACATAGGTATGGGAGTGACTTATCTTG
CCCTACTAGCAGCTTTTAAAGTTAGACCAACTTTTGCAGCTGGACTACTCTTGAGAAA
ACTGACCTCCAAGGAATTGATGATGGCCACCATAGGAATCGCACTCCTTTCCCAAAGC
ACCATACCAGAGACCATTCTTGAACTGACTGATGCGTTAGCTTTGGGCATGATGGTCC
TCAAAATAGTGAGAAATATGGAAAAATACCAATTGGCAGTGACTATCATGGCTATTTC
GTGTGTCCCAAATGCAGTGATACTGCAAAACGCATGGAAGGTGAGTTGCACAATATTG
GCAGCGGTGTCCGTTTCACCACTGCTCTTAACATCCTCACAGCAGAAAGCGGATTGGA
TACCACTGGCATTGACGATAAAAGGTCTCAATCCAACAGCCATTTTTCTAACAACTCT
TTCGAGAACCAGCAAGAAAAGGAGCTGGCCGCTAAATGAAGCTATCATGGCAGTCGG
GATGGTGAGCATTTTAGCCAGTTCTCTCCTAAAGAATGATATTCCCATGACAGGTCCA
TTAGTGGCTGGAGGGCTCCTTACCGTATGTTACGTGCTCACTGGACGATCGGCCGATT
TGGAACTGGAGAGAGCTGCCGATGTAAAATGGGAAGATCAGGCAGAAATATCAGGAA
GCAGCCCAATCCTGTCAATAACAATATCAGAAGATGGCAGCATGTCGATAAAAAATG
AAGAGGAAGAACAAACACTGACCATACTCATCAGAACGGGATTGTTGGTGATCTCAG
GAGTCTTTCCAGTATCGATACCAATTACGGCAGCAGCATGGTACCTGTGGGAAGTGAA
GAAACAACGGGCTGGAGTATTGTGGGACGTCCCTTCACCCCCACCAGTGGAAAAAGC
CGAACTGGAGGATGGAGCCTACAGAATCAAGCAAAGAGGGATTCTTGGATATTCTCA
GATTGGAGCCGGAGTTTACAAAGAAGGAACATTCCATACAATGTGGCACGTCACACG
TGGTGCTGTTCTGATGCATAGAGGGAAGAGGATTGAACCATCATGGGCAGATGTCAA
GAAAGACCTAATATCATATGGAGGAGGCTGGAAGCTAGAAGGAGAATGGAAGGAAG
GAGAGGAAGTCCAAGTCCTGGCATTGGAACCTGGAAAAAATCCAAGAGCCGTCCAAA
CGAAACCTGGAATTTTCAAAACCAACACCGGAACCATAGGCGCCGTATCTCTGGACTT
TTCCCCTGGAACGTCAGGATCTCCAATTGTCGACAGAAAAGGAAAAGTTGTGGGTCTT
TACGGTAATGGTGTTGTCACAAGGAGTGGAGCATATGTAAGTGCTATAGCCCAGACCG
AAAAAAGCATTGAAGACAATCCAGAGATCGAAGAAGACATTTTCCGAAAGAAAAGAT
TGACCATCATGGACCTCCATCCAGGAGCAGGAAAGACAAAAAGATACCTTCCAGCCA
TAGTTAGAGAAGCCATAAAACGTGGCTTGAGAACATTGATCCTGGCTCCCACTAGAGT
AGTGGCAGCTGAAATGGAGGAAGCTCTTAGAGGACTTCCAATAAGATACCAAACCCC
AGCCATCAAAACCGAGCATACCGGGCGGGAGATCGTGGACCTAATGTGTCATGCCAC
ATTTACTATGAGGCTGCTATCACCAGTCAGAGTGCCAAATTACAACCTGATCATCATG
GACGAAGCCCACTTCACAGACCCAGCAAGTATAGCAGCTAGAGGATACATTTCAACT
CGAGTAGAGATGGGTGAAGCAGCCGGGATTTTTATGACAGCCACTCCTCCGGGAAGT
AGAGACCCATTTCCACAGAGCAATGCACCAATCATGGATGAGGAAAGAGAAATCCCT
GAGCGTTCGTGGAATTCAGGACACGAATGGGTCACGGATTTTAAGGGAAAGACTGTTT
GGTTTGTTCCAAGTATAAAAGCAGGAAATGATATAGCAGCTTGTCTTAGGAAAAATGG
AAAGAAAGTGATACAACTCAGTAGGAAGACTTTTGACTCTGAGTATGTTAAGACTAG
AGCCAATGATTGGGACTTTGTGGTCACAACTGACATTTCAGAAATGGGTGCCAACTTC
AAGGCTGAGAGGGTTATAGACCCCAGACGTTGCATGAAACCAGTTATACTAACAGAT
GGCGAAGAGCGGGTGATCTTGGCAGGACCTATGCCAGTGACCCACTCTAGTGCAGCG
CAAAGAAGAGGGAGAATAGGAAGAAATCCAAAAAATGAAAATGACCAGTACATATA
CATGGGGGAACCTCTTGAAAATGATGAAGACTGTGCACATTGGAAAGAAGCTAAAAT
GCTCCTAGATAACATCAACACACCCGAAGGAATCATTCCTAGTATGTTCGAACCAGAG
CGTGAAAAAGTGGATGCCATTGATGGTGAATACCGTTTGAGAGGAGAAGCAAGGAAA
ACCTTTGTGGACCTAATGAGAAGAGGGGACTTACCAGTCTGGTTGGCCTACAAAGTGG
CAGCTGAAGGCATCAACTACGCAGACAGAAAGTGGTGTTTTGATGGAATTAAGAACA
ACCAAATACTGGAAGAAAATATGGAAGTGGAAATCTGGACAAAAGAAGGGGAAAGG
```

```
AAAAAATTAAAACCCAGATGGTTGGATGCTAGGATCTATTCTGACCCACTGGCACTAA
AAGAATTCAAGGAATTTGCAGCTGGAAGAAAATCTTTGACCCTGAACCTAATCACAG
AAATGGGTAGGCTTCCAACTTTCATGACTCAGAAAGCAAGAAACGCACTGGACAACT
TGGCTGTGCTGCATACGGCTGAGGCAGGTGGAAGGGCGTACAATCATGCTCTCAGTGA
ACTGCCGGAGACCCTGGAGACACTGCTTCTACTGACACTCCTGGCAACAGTCACAGGA
GGAATCTTCTTATTCTTAATGAGCGGAAAAGGTATAGGGAAGATGACCCTGGGAATGT
GTTGCATAATCACGGCTAGTATCCTCCTATGGTATGCACAGATACAACCACACTGGAT
AGCAGCTTCAATAATACTGGAGTTTTTCTCATAGTTTTGCTCATTCCAGAACCAGAAA
AACAGAGAACACCCCAAGACAACCAATTGACCTACGTTGTCATAGCCATCCTCACAGT
GGTGGCCGCAACCATGGCAAACGAGATGGGTTTCCTGGAAAAAACCAAGAAAGACCT
CGGATTGGGAAGCATTACAACCCAGGAATCTGAGAGCAATATCCTGGACATAGATCT
ACGCCCTGCATCAGCATGGACGCTGTATGCCGTGGCTACAACATTTGTCACACCAATG
TTGAGACATAGCATTGAAAATTCCTCAGTGAATGTCTCCCTAACAGCCATTGCTAACC
AAGCTACAGTGCTAATGGGTCTTGGGAAAGGATGGCCATTGTCAAAGATGGACATTG
GAGTTCCCCTCCTTGCCATTGGATGCTATTCACAAGTCAACCCTATAACTCTCACAGCA
GCTCTTCTTTTATTGGTAGCACATTATGCCATTATAGGGCCAGGACTTCAAGCAAAAG
CAACCAGAGAAGCTCAGAAAAGAGCGGCAGCAGGCATCATGAAAAACCCAACAGTC
GATGGAATAACAGTGATTGACCTAGAACCAATACCCTATGATCCAAAATTTGAAAAG
CAGTTAGGACAAGTAATGCTCCTAATCCTCTGCGTGACTCAAGTATTAATGATGAGGA
CTACATGGGCTTTGTGTGAGGCTCTAACCCTAGCGACCGGGCCCATCTCCACACTGTG
GGAAGGAAATCCAGGGAGGTTTTGGAACACTACCATTGCAGTGTCAATGGCTAACAT
CTTTAGGGGGAGCTACTTGGCCGGAGCTGGACTTCTCTTTTCCATCATGAAAAACACA
ACAAACACAAGAAGAGGAACTGGCAACATAGGAGAGACACTTGGAGAAAAATGGAA
AAGTCGATTAAACGCACTGGGAAAAAGTGAATTTCAAATCTACAAGAAAAGTGGAAT
CCAGGAAGTGGATAGAACCCTAGCAAAAGAAGGTATCAAAAGAGGAGAAACGGACC
ACCATGCTGTGTCGCGAGGTTCAGCAAAACTGAGATGGTTCGTCGAGAGAAATATGGT
CACACCGGAAGGGAAGGTGGTGGATCTCGGTTGCGGCAGAGGGGGCTGGTCATACTA
TTGTGGGGGACTAAAGAATGTAAGAGAAGTCAAAGGCCTAACAAAAGGAGGACCAG
GACATGAAGAACCCATCCCCATGTCAACATATGGGTGGAATCTAGTGCGTCTGCAAAG
TGGAGTTGACGTTTTCTTCACCCCGCCAGAAAAGTGTGATACATTGTTGTGTGACATA
GGGGAGTCGTCACCAAATCCCACGATAGAAGCAGGACGAACACTCAGAGTCCTCAAC
TTAGTGGAAAATTGGTTGAACAATAACACCCAATTTTGCATAAAGGTCCTCAACCCAT
ATATGCCTTCAGTCATAGAAAAAATGGAAGCATTACAAAGGAAATATGGAGGAGCCT
TAGTGAGGAATCCACTCTCACGAAACTCCACGCATGAAATGTACTGGGTATCTAATGC
CACCGGGAACATAGTGTCATCAGTGAACATGATTTCAAGGATGTTGATTAACAGATTC
ACAATGAAACACAAGAAAGCCACTTACGAGCCAGATGTTGACCTAGGAAGTGGAACC
CGCAACATTGGAATTGAAAGTGAGATACCAAATCTAGACATAATAGGAAAGAGAATA
GAGAAAATAAAACAAGAGCATGAAACATCATGGCACTATGATCAAGACCACCCATAC
AAAACGTGGGCTTACCATGGCAGCTATGAAACAAAACAAACTGGATCAGCATCATCT
ATGGTGAACGGAGTGGTCAGATTGCTGACAAAACCTTGGGATGTCGTCCCTATGGTGA
CACAGATGGCAATGACAGACACGACTCCATTTGGACAACAGCGCGTTTTCAAAGAGA
AAGTAGACACGAGAACCCAAGAACCGAAGGAAGGCACAAAGAAACTGATGAAAATT
ACGGCAGAGTGGCTTTGGAAAGAACTAGGAAAGAAAAAGACACCTAGGATGTGTACC
AGAGAAGAATTCACAAGAAAAGTGAGAAGCAATGCAGCCTTGGGGGCCATATTCACT
GATGAGAACAAATGGAAATCGGCACGTGAGGCTGTTGAAGATGGTAGGTTTTGGGAG
CTGGTTGACAGGGAAAGAAATCTCCATCTTGAAGGAAAGTGTGAAACATGTGTGTAC
AACATGATGGGAAAAAGAGAGAAGAAACTAGGGGAGTTCGGCAAGGCAAAAGGTAG
CAGAGCCATATGGTACATGTGGCTTGGAGCACGCTTCTTAGAGTTTGAAGCCCTAGGA
TTCCTGAATGAAGATCACTGGTTCTCCAGAGGGAACTCCCTGAGTGGAGTGGAAGGA
GAAGGGCTGCACAGGCTAGGCTACATTTTAAGAGACGTGGGCAAGAAGGAAGGGGG
AGCAATGTACGCCGATGATACAGCAGGATGGGACACAAGAATCACACTAGAAGACTT
AAAAAATGAAGAAATGGTAACAAATCACATGAAAGGAGAACACAAGAAACTAGCCG
AGGCTATATTCAAATTAACGTACCAAAACAAGGTGGTGCGTGTGCAAAGACCAACAC
CAAGAGGCACAGTAATGGATATCATATCGAGAAGAGACCAAAGAGGCAGTGGGCAA
GTCGGCACCTATGGCCTTAATACTTTCACCAATATGGAAGCCCAATTAATTAGACAGA
```

```
TGGAGGGAGAAGGAATCTTCAAAAGCATTCAGCACCTGACAGCCACAGAAGAAATCG
CTGTACAGAACTGGTTAGCAAGAGTGGGGCGTGAAAGGCTATCAAGAATGGCAATCA
GTGGAGATGATTGTGTTGTAAAACCTATAGATGACAGATTTGCAAGTGCTTTAACAGC
TCTAAATGACATGGGAAAAGTTAGGAAAGATATACAACAATGGGAACCTTCAAGAGG
ATGGAACGATTGGACACAAGTGCCTTTCTGTTCACACCATTTTCATGAGTTAGTCATG
AAAGATGGTCGCGTGCTCGTAGTCCCATGCAGAAACCAAGATGAACTGATTGGTAGA
GCCCGAATTTCCCAGGGAGCTGGGTGGTCTTTGAAGGAGACGGCCTGTTTGGGGAAGT
CTTACGCCCAAATGTGGACCCTGATGTACTTCCACAGACGTGACCTCAGACTGGCGGC
AAATGCCATTTGCTCGGCAGTCCCGCCACATTGGGTTCCAACAAGTCGAACAACCTGG
TCCATACACGCTAAGCATGAATGGATGACGACGGAAGACATGCTGGCAGTCTGGAAC
AGGGTGTGGATCCAAGAAAACCCGTGGATGGAAGATAAAACTCCAGTGGAATCATGG
GAAGAAGTCCCATACTTGGGAAAAAGAGAAGACCAATGGTGCGGCTCATTGATTGGG
CTAACAAGCAGGGCTACCTGGGCAAAGAACATCCAAACAGCAATAAATCAAGTCAGA
TCCCTCATAGGCAATGAGGAATACACAGACTACATGCCATCCATGAAGAGATTCAGA
AGGGAAGAGGAAGAGGCAGGTGTCCTGTGGTAGAAGGCAAAACTAACATGAAACAA
GGCTAAAAGTCAGGTCGGATTAAGCCATAGTACGGAAAAAACTATGCTACCTGTGAG
CCCCGTCCAAGGACGTTAAAAGAAGTCAGGCCATCACAAAATGCCACAGCTTGAGTA
AACTGTGCAGCCTGTAGCTCCACCTGAGGAGGTGTAAAAAACCTGGGAGGCCACAAA
CCATGGAAGCTGTACGCATGGCGTAGTGGACTAGCGGTTAGAGGAGACCCCTCCCTTA
CAAATCGCAGCAAACAACGGGGGCCCAAGGTGAGATGAAGCTGTAATCTCACTGGAA
GGACTAGAGGTTAGAGGAGACCCCCCCAAAACAAAAAACAGCATATTGACGCTGGGA
AAGACCAGAGATCCTGCTGTCTCCTCAGCATCATTCCAGGCACAGAACGCCAGAAAAT
GGAAT
```

```
>FJ850065 Dengue virus 2 isolate DENV-2/NI/BID-V2664/2000, complete genome
ACAAAGACAGATTCTTTGAGGGAGCTAAGCTCAACGTAGTTCTAACAGTTTTTGATTA
GAGAGCAGATCTCTGATGAATAACCAACGGAAAAAGGCGAGAAGTACGCCTTTCAAT
ATGCTGAAACGCGAGAGAAACCGCGTGTCAACTGTGCAACAGCTGACAAAGAGATTC
TCACTTGGAATGCTGCAAGGACGCGGACCATTAAAACTGTTCATGGCCCTTGTGGCGT
TCCTTCGTTTCCTAACAATCCCACCAACAGCAGGGATACTAAAAAGATGGGGAACGAT
CAAAAAATCAAAAGCTATCAATGTTTTGAGAGGGTTCAGGAAAGAGATTGGAAGGAT
GCTGAACATCTTGAACAGGAGACGCAGGACAGCAGGCGTGATTGTTATGTTGATTCCA
ACAGCGATGGCGTTCCATTTAACCACACGCAATGGAGAACCACACATGATCGTTGGTA
GGCAGGAGAAAGGGAAAAGTCTTCTGTTCAAAACAGAGGATGGTGTTAACATGTGTA
CCCTCATGGCCATAGACCTTGGTGAATTGTGTGAAGATACAATCACGTACAAGTGTCC
TCTCCTCAGACAAAATGAACCAGAAGACATAGATTGTTGGTGCAACTCTACGTCCACA
TGGGTAACTTATGGGACATGTACCACCACAGGAGAACACAGAAGAGAAAAAAGATCA
GTGGCGCTCGTTCCACATGTGGGTATGGGACTGGAGACACGAACTGAAACATGGATG
TCATCAGAAGGGGCCTGGAAACATGTTCAGAGAATTGAAACCTGGATCTTGAGACAT
CCAGGCTTTACCATAATGGCAGCAATCCTGGCATACACCATAGGAACGACACATTTCC
AAAGGGCCTTGATTTTCATCTTACTGACAGCTGTCGCTCCTTCAATGACAATGCGCTGC
ATAGGAATTTCAAATAGAGACTTCGTAGAAGGGGTTTCAGGAGGAAGCTGGGTTGAC
ATAGTCTTAGAACATGGAAGTTGTGTGACGACGATGGCAAAAAACAAACCAACATTG
GATTTTGAACTGATAAAAACAGAAGCCAAACAACCTGCCACTCTAAGGAAGTACTGT
ACAGAAGCAAAGCTGACCAACACAACAACAGAATCGCGTTGCCCAACACAAGGGGA
ACCCAGTCTAAATGAAGAGCAGGACAAAAGGTTCATCTGCAAACACTCCATGGTAGA
CAGAGGATGGGGAAATGGATGTGGATTATTTGGAAAGGGAGGCATTGTGACCTGTGC
TATGTTTACATGCAAAAAGAACATGGAAGGAAAAGTCGTGCAGCCAGAAAATTTGGA
ATACACCATCGTGATAACACCTCACTCAGGAGAAGAGCACGCTGTAGGTAATGACAC
AGGAAAGCATGGCAAGGAAATCAAAATAACACCACAGAGTTCCATCACAGAAGCAG
AACTGACAGGCTATGGCACTGTCACGATGGAGTGCTCTCCGAGAACGGGCCTCGACTT
CAACGAGATGGTGCTGCTGCAGATGGAAGACAAAGCTTGGCTGGTGCACAGGCAATG
GTTCCTAGACCTGCCGTTACCATGGCTACCCGGAGCGGACACACAAGGATCAAATTGG
ATACAGAAAGAGACATTGGTCACTTTCAAAAATCCCCACGCGAAGAAACAGGATGTC
GTTGTTTTAGGGTCTCAAGAAGGGGCCATGCACACGGCACTCACAGGGGCCACAGAA
ATCCAGATGTCATCAGGAAACTTACTGTTCACAGGACATCTCAAGTGCAGGCTGAGAA
TGGACAAACTACAGCTCAAAGGAATGTCATACTCTATGTGTACAGGAAAGTTTAAAAT
TGTGAAGGAAATAGCAGAAACACAACATGGAACAATAGTTATCAGAGTACAATATGA
AGGGGACGGTTCTCCATGTAAGATCCCTTTTGAGATAACAGATTTGGAAAAAAGACAC
GTCTTAGGTCGCTTGATTACAGTTAACCCAATCGTAACAGAAAAAGATAGCCCAGTCA
ACATAGAAGCAGAACCTCCATTCGGAGACAGCTACATCATCATAGGAGTAGAGCCGG
GACAATTGAAACTCAATTGGTTTAAGAAGGGAAGTTCCATCGGCCAAATGTTTGAGAC
AACAATGAGAGGAGCAAAGAGAATGGCCATTTTAGGTGACACAGCCTGGGACTTTGG
ATCCCTGGGAGGAGTGTTTACATCTATAGGAAAGGCTCTCCACCAAGTTTTCGGAGCA
ATCTATGGGGCTGCTTTTAGTGGGGTCTCATGGACTATGAAAATCCTCATAGGAGTCA
TCATCACATGGATAGGAATGAATTCACGTAGCACCTCACTGTCTGTGTCGCTAGTATT
GGTGGGCGTCGTGACACTGTACCTGGGAGCTATGGTGCAAGCTGATAGTGGTTGCGTT
GTGAGCTGGAAAAATAAAGAACTGAAATGTGGCAGCGGGATCTTCATCACAGATAAC
GTACACACATGGACAGAACAATATAAGTTCCAACCAGAATCCCCTTCAAAACTAGCTT
CAGCTATCCAAAAAGCTCATGAAGAGGGCATTTGTGGAATCCGCTCAGTAACAAGATT
GGAGAATCTGATGTGGAAACAAATAACACCAGAATTGAATCATATTCTATCAGAAAA
TGAGGTAAAGTTGACCATTATGACAGGAGACATTAAAGGAATCATGCAGGCAGGAAA
ACGATCCTTGCGGCCCCAGCCCACTGAGCTGAAGTACTCATGGAAAACATGGGGAAA
GGCGAAAATGCTCTCCACAGAGTCTCACAATCAGACCTTTCTTATTGATGGCCCTGAA
ACAGCAGAATGCCCCAACACAAACAGAGCTTGGAACTCGCTGGAAGTTGAAGACTAT
GGTTTTGGAGTTTTCACCACCAATATATGGCTGAAATTGAGAGAAAAACAGGATGTAT
TTTGTGACTCAAAACTCATGTCAGCGGCCATTAAAGACAACAGAGCCGTTCATGCCGA
TATGGGTTATTGGATAGAAAGTGCACTCAATGACACATGGAAGATGGAGAAAGCCTC
CTTCATTGAAGTTAAAAGCTGCCACTGGCCAAAGTCACACACCCTTTGGAGCAATGGA
```

```
GTATTAGAAAGTGAGATGATAATCCCAAAAAATTTTGCCGGGCCAGTGTCACAACAC
AACTACAGACCAGGCTACCATACACAAACAGCAGGACCTTGGCATCTAGGTAAGCTT
GAGATGGACTTTGATCTCTGCGAAGGAACTACAGTGGTGGTGACTGAGGACTGTGGA
AATAGAGGACCCTCTTTAAGAACGACCACTGCCTCTGGAAAACTCATAACAGAATGGT
GCTGCCGATCCTGCACACTACCACCTCTAAGATACAGAGGTGAGGATGGATGCTGGTA
CGGGATGGAAATCAGACCATTGAAAGGAAAGAGGAGAATTTGGTTAACTCCTTGGTC
ACAGCCGGACATGGGCAGATTGACAACTTTTCACTAGGAGTCTTGGGAATGGCACTGT
TCCTGGAAGAAATGCTCAGGACCCGAATAGGAACGAAACATGCAATACTGCTAGTTG
CAGTATCTTTTGTGACATTGATTACTGGGAACATGTCTTTTAGAGACCTGGGAAGAGT
GATGGTTATGGTGGGCGCTACCATGACGGATGACATAGGTATGGGAGTGACTTATCTT
GCCCTACTAGCAGCTTTTAAAGTTAGACCAACTTTTGCAGCTGGACTACTCTTGAGAA
AACTGACCTCCAAGGAATTGATGATGGCCACCATAGGAATCGCACTCCTTTCCCAAAG
CACCATACCAGAGACCATTCTTGAACTGACTGATGCGTTAGCCTTGGGCATGATGGTC
CTCAAAATAGTGAGAAATATGGAAAAATACCAATTGGCAGTGACTATCATGGCTATTT
CGTGTGTCCCAAATGCAGTGATACTGCAAAACGCATGGAAGGTGAGTTGCACAATATT
GGCAGCGGTGTCCGTTTCACCACTGCTCTTAACATCCTCACAGCAGAAAGCGGATTGG
ATACCACTGGCATTGACGATAAAAGGTCTCAATCCAACAGCCATTTTTCTAACAACTC
TTTCGAGAACCAGCAAGAAAAGGAGCTGGCCGCTAAATGAAGCTATCATGGCAGTCG
GGATGGTGAGCATTTTAGCCAGTTCTCTCCTAAAGAATGATATTCCCATGACAGGTCC
ATTAGTGGCTGGAGGGCTCCTTACCGTATGTTACGTGCTCACTGGACGATCGGCCGAT
TTGGAACTGGAGAGAGCTGCCGATGTAAAATGGGAAGATCAGGCAGAAATATCAGGA
AGCAGCCCAATCCTGTCAATAACAATATCAGAAGATGGCAGCATGTCGATAAAAAAT
GAAGAGGAAGAACAAACACTGACCATACTCATCAGAACGGGTTTGTTGGTGATCTCA
GGAGTCTTTCCAGTATCGATACCAATTACGGCAGCAGCATGGTACCTGTGGGAAGTGA
AGAAACAACGGGCTGGAGTATTGTGGGACGTCCCTTCACCCCCACCAGTGGAAAAAG
CCGAACTGGAGGATGGAGCCTACAGAATCAAGCAAAGAGGGATTCTTGGATATTCTC
AGATTGGAGCCGGAGTTTACAAAGAAGGAACATTCCATACAATGTGGCACGTCACAC
GTGGTGCTGTTCTGATGCATAGAGGGAAGAGGATTGAACCATCATGGGCAGATGTCA
AGAAAGACCTAATATCATATGGAGGAGGCTGGAAGCTAGAAGGAGAATGGAAGGAA
GGAGAGGAAGTCCAAGTCCTGGCATTGGAACCTGGAAAAAATCCAAGAGCCGTCCAA
ACGAAACCTGGAATTTTCAAAACCAACACCGGAACCATAGGCGCCGTATCTCTGGACT
TTTCCCCTGGAACGTCAGGATCTCCAATTGTCGACAGAAAAGGAAAAGTTGTGGGTCT
TTACGGTAATGGTGTTGTCACAAGGAGTGGAGCATATGTAAGTGCTATAGCCCAGACC
GAAAAAAGCATTGAAGACAATCCAGAGATCGAAGAAGACATTTTCCGAAAGAAAAG
ATTGACCATCATGGACCTCCATCCAGGAGCAGGAAAGACAAAAAGATACCTTCCAGC
CATAGTTAGAGAAGCCATTAAACGTGGCTTGAGAACATTGATCCTGGCTCCCACTAGA
GTAGTGGCAGCTGAAATGGAGGAAGCTCTTAGAGGACTTCCAATAAGATACCAAACC
CCAGCCATCAAAACCGAGCATACCGGGCGGGAGATCGTGGACCTAATGTGTCATGCC
ACATTTACTATGAGGCTGCTATCACCAGTCAGAGTGCCAAATTACAACCTGATCATCA
TGGACGAAGCCCACTTCACAGACCCAGCAAGTATAGCAGCTAGAGGATACATTTCAA
CTCGAGTAGAGATGGGTGAAGCAGCCGGGATTTTCATGACAGCCACTCCTCCGGGAA
GTAGAGACCCATTTCCACAGAGCAATGCACCAATCATGGATGAGGAAAGAGAAATCC
CTGAGCGTTCGTGGAATTCAGGACACGAATGGGTCACGGATTTTAAGGGAAAGACTG
TTTGGTTTGTTCCAAGTATAAAAGCAGGAAATGATATAGCAGCTTGTCTTAGGAAAAA
TGGAAAGAAAGTGATACAACTCAGTAGGAAGACTTTTGACTCTGAGTATGTTAAGACT
AGAGCCAATGATTGGGACTTTGTGGTCACAACTGACATTTCAGAAATGGGTGCCAACT
TCAAGGCTGAGAGGGTTATAGACCCCAGACGTTGCATGAAACCAGTTATACTAACAG
ATGGCGAAGAGCGGGTGATCTTGGCAGGACCTATGCCAGTGACCCACTCTAGTGCAG
CGCAAAGAAGAGGGAGAATAGGAAGAAATCCAAAAAATGAAAATGACCAGTACATA
TACATGGGGGAACCTCTTGAAAATGATGAAGACTGTGCACATTGGAAAGAAGCTAAA
ATGCTCCTAGATAACATCAACACACCCGAAGGAATCATTCCTAGTATGTTCGAACCAG
AGCGTGAAAAAGTGGATGCCATTGATGGTGAATACCGTTTGAGAGGAGAAGCAAGGA
AAACCTTTGTGGACCTAATGAGAAGAGGGGACTTACCAGTCTGGTTGGCCTACAAAGT
GGCAGCTGAAGGCATCAACTACGCAGACAGAAAGTGGTGTTTTGATGGAATTAAGAA
CAACCAAATACTGGAAGAAAATATGGAAGTGGAAATCTGGACAAAAGAAGGGGAAA
```

```
GGAAAAAATTAAAACCCAGATGGTTGGATGCTAGGATCTATTCTGACCCACTGGCACT
AAAAGAATTCAAGGAATTTGCAGCTGGAAGAAAATCTTTGACCCTGAACCTAATCAC
AGAAATGGGTAGGCTTCCAACTTTCATGACTCAGAAAGCAAGAAACGCACTGGACAA
CTTGGCTGTGCTGCATACGGCTGAGGCAGGTGGAAGGGCGTACAATCATGCTCTCAGT
GAACTGCCGGAGACCCTGGAGACACTGCTTCTACTGACACTCCTGGCAACAGTCACAG
GAGGAATCTTCTTATTCTTAATGAGCGGAAAAGGTATAGGGAAGATGACCCTGGGAA
TGTGTTGCATAATCACGGCTAGTATCCTCCTATGGTATGCACAGATACAACCACATTG
GATAGCAGCTTCAATAATACTGGAGTTTTTCTCATAGTTTTGCTCATTCCAGAACCAG
AAAAACAGAGAACACCCCAAGACAACCAATTGACCTACGTTGTCATAGCCATCCTCA
CAGTAGTGGCCGCAACCATGGCAAACGAGATGGGTTTCCTGGAAAAAACCAAGAAAG
ACCTCGGATTGGGAAGCATTACAACCCAGGAATCTGAGAGCAATATCCTGGACATAG
ATCTACGCCCTGCATCAGCATGGACGCTGTATGCCGTGGCTACAACATTTGTCACACC
AATGTTGAGACATAGCATTGAAAATTCCTCAGTGAATGTCTCCCTAACAGCCATTGCT
AACCAAGCTACAGTGCTAATGGGTCTTGGGAAAGGATGGCCATTGTCAAAGATGGAC
ATTGGAGTTCCCCTCCTTGCCATTGGATGCTATTCACAAGTCAACCCTATAACTCTCAC
AGCAGCTCTTCTTTTATTGGTAGCACATTATGCCATTATAGGGCCAGGACTTCAAGCA
AAAGCAACCAGAGAAGCTCAGAAAAGAGCGGCAGCAGGCATCATGAAAAACCCAAC
AGTCGATGGAATAACAGTGATTGACCTAGAACCAATACCCTATGATCCAAAATTTGAA
AAGCAGTTAGGACAAGTAATGCTCCTAATCCTCTGCGTGACTCAAGTATTAATGATGA
GGACTACATGGGCTTTGTGTGAGGCTCTAACCCTAGCGACCGGGCCCATCTCCACACT
GTGGGAAGGAAATCCAGGGAGGTTTTGGAACACTACCATTGCAGTGTCAATGGCTAA
CATCTTTAGGGGGAGCTACTTGGCCGGAGCTGGACTTCTCTTTTCCATCATGAAAAAC
ACAACAAACACAAGAAGAGGAACTGGCAACATAGGAGAGACACTTGGAGAAAAATG
GAAAAGTCGATTAAACGCACTGGGAAAAAGTGAATTTCAAATCTACAAGAAAAGTGG
AATCCAGGAAGTGGATAGAACCCTAGCAAAAGAAGGTATCAAAAGAGGAGAAACGG
ACCACCATGCTGTGTCGCGAGGTTCAGCAAAACTGAGATGGTTCGTCGAGAGAAATAT
GGTCACACCGGAAGGGAAGGTGGTGGATCTCGGTTGCGGCAGAGGGGGCTGGTCATA
CTATTGTGGGGGACTAAAGAATGTAAGAGAAGTCAAAGGCCTAACAAAAGGAGGACC
AGGACATGAAGAACCCATCCCCATGTCAACATATGGGTGGAATCTAGTGCGTCTGCAA
AGTGGAGTTGACGTTTTCTTCACCCCGCCAGAAAAGTGTGATACATTGTTGTGTGACA
TAGGGGAGTCGTCACCAAATCCCACGATAGAAGCAGGACGAACACTCAGAGTCCTCA
ACTTAGTGGAAAATTGGTTGAACAATAACACCCAATTTTGCATAAAGGTCCTCAACCC
ATATATGCCTTCAGTCATAGAAAAAATGGAAGCATTACAAAGGAAACATGGAGGAGC
CTTAGTGAGGAATCCACTCTCACGAAACTCCACGCATGAAATGTACTGGGTATCTAAT
GCCACCGGGAACATAGTGTCATCAGTGAACATGATTTCAAGGATGTTGATTAACAGAT
TCACAATGAAACACAAGAAAGCCACTTACGAGCCAGATGTTGACCTAGGAAGTGGAA
CCCGCAACATTGGAATTGAAAGTGAGATACCAAATCTAGACATAATAGGAAAGAGAA
TAGAGAAAATAAAACAAGAGCATGAAACATCATGGCACTATGATCAAGACCACCCAT
ACAAAACGTGGGCTTACCATGGCAGCTATGAAACAAAACAAACTGGATCAGCATCAT
CTATGGTGAACGGAGTGGTCAGATTGCTGACAAAACCTTGGGATGTCGTCCCTATGGT
GACACAGATGGCAATGACAGACACGACTCCATTTGGACAACAGCGCGTTTTCAAAGA
GAAAGTAGACACGAGAACCCAAGAACCGAAGGAAGGCACAAAGAAACTGATGAAAA
TTACGGCAGAGTGGCTTTGGAAAGAACTAGGAAAGAAAAAGACACCTAGGATGTGTA
CCAGAGAAGAATTCACAAGAAAAGTGAGAAGCAATGCAGCCTTGGGGGCCATATTCA
CTGATGAGAACAAATGGAAATCGGCACGTGAGGCTGTTGAAGATGGTAGGTTTTGGG
AGCTGGTTGACAGGGAAAGAAATCTCCATCTTGAAGGAAAGTGTGAAACATGTGTGT
ACAACATGATGGGAAAAAGAGAGAAGAAACTAGGGGAGTTCGGCAAGGCAAAAGGT
AGCAGAGCCATATGGTACATGTGGCTTGGAGCACGCTTCTTAGAGTTTGAAGCCCTAG
GATTCCTGAATGAAGATCACTGGTTCTCCAGAGGGAACTCCCTGAGTGGAGTGGAAG
GAGAAGGGCTGCACAGGCTAGGCTACATTTTAAGAGACGTGGGCAAGAAGGAAGGG
GGAGCAATGTACGCCGATGATACAGCAGGATGGGACACAAGAATCACACTAGAAGAC
TTAAAAAATGAAGAAATGGTAACAAATCACATGAAAGGAGAACACAAGAAACTAGC
CGAGGCTATATTCAAATTAACGTACCAAAACAAGGTGGTGCGTGTGCAAAGACCAAC
ACCAAGAGGCACAGTAATGGATATCATATCGAGAAGAGACCAAAGAGGCAGTGGGC
AAGTCGGCACCTATGGCCTTAATACTTTCACCAATATGGAAGCCCAATTAATTAGACA
```

```
GATGGAGGGAGAAGGAATCTTCAAAAGCATTCAGCACCTGACAGCCACAGAAGAGAT
CGCTGTACAGAACTGGTTAGCAAGAGTGGGGCGTGAAAGGCTATCAAGAATGGCAAT
CAGTGGAGATGATTGTGTTGTAAAACCTATAGATGACAGATTTGCAAGTGCTTTAACA
GCTCTAAATGACATGGGAAAAGTTAGGAAAGATATACAACAATGGGAACCTTCAAGA
GGATGGAACGATTGGACACAAGTGCCTTTCTGTTCACACCATTTTCATGAGTTAGTCA
TGAAAGATGGTCGCGTGCTCGTAGTCCCATGCAGAAACCAAGATGAACTGATTGGTA
GAGCCCGAATCTCCCAGGGAGCTGGGTGGTCTTTGAAGGAGACGGCCTGTTTGGGGA
AGTCTTACGCCCAAATGTGGACCCTGATGTACTTCCACAGACGTGACCTCAGACTGGC
GGCAAATGCCATTTGCTCGGCAGTCCCGCCACATTGGGTTCCAACAAGTCGAACAACC
TGGTCCATACACGCTAAGCATGAATGGATGACGACGGAAGACATGCTGGCAGTCTGG
AACAGGGTGTGGATCCAAGAAAACCCGTGGATGGAAGACAAAACTCCAGTGGAATCA
TGGGAAGAAGTCCCATACTTGGGAAAAAGAGAAGACCAATGGTGCGGCTCATTGATT
GGGCTAACAAGCAGGGCTACCTGGGCAAAGAACATCCAAACAGCAATAAATCAAGTC
AGATCCCTTATAGGCAATGAGGAATACACAGACTACATGCCATCCATGAAGAGATTC
AGAAGGGAAGAGGAAGAGGCAGGTGTCCTGTGGTAGAAGGCAAAACTAACATGAAA
CAAGGCTAAAAGTCAGGTCGGATTAAGCCATAGTACGGAAAAAACTATGCTACCTGT
GAGCCCCGTCCAAGGACGTTAAAAGAAGTCAGGCCATCACAAAATGCCACAGCTTGA
GTAAACTGTGCAGCCTGTAGCTCCACCTGAGGAGGTGTAAAAAACCTGGGAGGCCAC
AAACCATGGAAGCTGTACGCATGGCGTAGTGGACTAGCGGTTAGAGGAGACCCCTCC
CTTACAAATCGCAGCAAACAACGGGGGCCCAAGGTGAGATGAAGCTGTAATCTCACT
GGAAGGACTAGAGGTTAGAGGAGACCCCCCCAAAACAAAAGACAGCATATTGACGCT
GGGAAAGACCAGAGTCCTGCTGTCTCCTCAGCATCATTCCAGGCACAGAACGCCAGA
AAATGGAATG
```

# APPENDIX B

# UnitX: DENGUE VIRUS SEQUENCE GRAPHICAL REPRESENTATION FOR SEROTYPES CLASSIFICATION

Boonyarat Viriyasaksathian, Yodchanan Wongsawat and Prapat Suriyaphol

***Abstract* - Searching for the graphical representations that yield the meaningful interpretation of the long sequences of nucleotides is one of the challenging problems. In this paper, we present the new graphical representation especially for the Dengue virus sequence analysis based on the cumulative amount of amino and keto bases, called UnitX method. Simulation results show that the proposed graphical representation can efficiently distinguish the four serotypes of Dengue virus.**

## I. INTRODUCTION

SEARCHING for the efficient and practical graphical representation for analyzing the Dengue virus sequences is one of the challenging problems in bioinformatics. In the era of information, fast and automatic analysis method is needed in conjunction with the meaningful graphical representation that experts can easily manipulate the enormous data.

There are many existing tools used for visualizing and analyzing the genomic sequence. Each tool is developed based on some specific tasks which can be categorized into four approaches: Base vector, Sequential, Fourier Transform (FT), and Z-curve approaches.

**(1) Base vector approach:** About twenty year ago, Hamori, E. and Ruskin, J. (1983) used a three dimensional H curve to represent a DNA sequence [1]. Afterwards, Gates, M.A. (1985) suggested that graphic representation of DNA sequence in two-dimensional that is simpler than H Curves. Gates' graphical representation represents four nucleotide bases adenines (A), thymine (T), cytosine (C) and guanine (G) by the unit vector on the Cartesian coordinate system with adenines (A) on the negative y-axis, thymine (T) on the positive y-axis, guanine (G) on the positive x-axis, and cytosine (C) on the negative x-axis [2]. About eleven years later, Nandy A. (1996) presented an easy graphical representation for distinct the features of intron and exon segments of eukaryotic sequences [3]. The four nucleotides A, G, C and T were plotted on an ACGT-axis system. The graphical representation was similar to use Gates' method. The indicator for clustering of intron and exon sequences was the slope of this plot. Nevertheless, the graphical represented by Nandy and Gates have high degeneracy, e.g. the sequences AGTC, AGTCA, AGTCAG, *etc.* would be displayed in the same graphical representation [4]. Stephen S. –T. Yau *et al.*, 2003 solve this problem by modified Gates' method. Yau's graphical representation represented a pyrimidine/purine graph on two quadrants of Cartesian coordinate system which the first quadrant was pyrimidines (T and C) and purine (A and G) was the fourth quadrant [4].

**(2) Sequential approach:** Altschul *et al.*, 1990 developed the Basic Local Alignment Search Tool (BLAST) programs. This program is one of the most popular tools for DNA and genomic analysis. This tool can perform a fast similarity search. The program compares the similarity between any two sequences and displays the difference between these sequences by comparing in the base-by-base basis [5].

**(3) Fourier Transform (FT) approach:** Anatassiou D. proposed about the color spectrograms of biomolecular sequences which are a tool of visualization for the biomolecular sequences analysis [7], [8]. Spectrograms will represent the magnitude of the short-time Fourier transform (STFT) that uses the discrete Fourier transform (DFT). Analyzing of the genomic sequence in frequency domain via the Fourier transform (FT) uses the 3-periodicity property for DNA coding sequence. The color spectrogram is defined by using the color: red, green and blue for the biomolecular sequences. However, this method consumes high computational complexity.

**(4) Z-Curve approach:** Zhang C. T. *et al.*, 1994 suggested about a very practical visualization tool called Z-Curve [8]-[12]. James J. *et al* developed this tool in the package called MBEToolbox [13]. According to the assumption on the cumulative components of the genomic sequence, features obtained from Z-Curve can be quickly interpreted, such as the distribution along the sequence of purine/pyrimidine bases, amino/keto bases, strong H-bond/weak H-bond. Since the algorithm of Z-Curve is very simple, it can be applied to all genomic sequences regardless of how long those sequences are. The similar approach with Z-Curve called 3DD-Curve is presented by Zhang Y. and Tan M. (2008). This approach can be viewed as the weighted version of Z-Curve.

Since our goal is to find the suitable graphical representation for Dengue virus. The sequences can be of arbitrary lengths in huge database. Therefore, computational complexity of the graphical representation needs to be low and base locations need to be easy to visualize. By comparing with many graphical representation of genomic sequence in the literature, we propose the novel graphical representation based on the modification of Yau's method [4]. As the projection along x-axis is changed to unity, the proposed method provides more physical meaning of the genomic sequences in the uniformly base-by-base basis while maintains the efficient computational complexity. Furthermore, the vector direction of the four bases A, G, T, C are also modified for the purpose of Dengue virus serotype classification. Moreover, our proposed graphical representation can be quickly interpreted the distribution of amino/keto bases along the sequence.

This paper can be organized as follows. Section II introduces the previous graphical representation method (unit vector representation [4]) of the four nucleotides. The proposed Unit X-projection Vector (UnitX) method is described in section III. In section IV, the simulation results on classification of Dengue virus sequences are shown. Finally, section V concludes the paper.

## II STEPHEN S. –T. YAU'S METHOD

Stephen S. –T. Yau' s graphical representation represented the distribution of pyrimidine/purine base along the sequence on two quadrants of Cartesian coordinate system where the first quadrant was pyrimidine (C and T) and purine (A and G) in the fourth quadrant [4]. The unit vectors that represent four nucleotides adenines (A), guanine (G), thymine (T), and cytosine (C), are defined based on rectangular coordinates as follows:

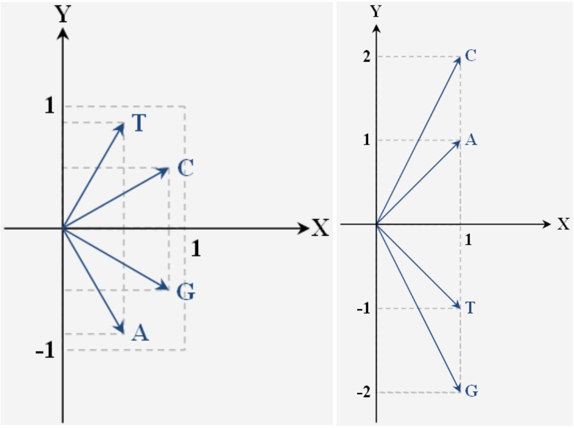
$$\mathrm{A} \rightarrow \left(\frac{1}{2},-\frac{\sqrt{3}}{2}\right) \qquad \mathrm{C} \rightarrow \left(\frac{\sqrt{3}}{2},\frac{1}{2}\right)$$

$$\mathrm{G} \rightarrow \left(\frac{\sqrt{3}}{2},-\frac{1}{2}\right) \qquad \mathrm{T} \rightarrow \left(\frac{1}{2},\frac{\sqrt{3}}{2}\right)$$

By given the numbers of occurring $a$, $c$, $g$, $t$ of bases A, C, G, T, respectively, the coordinate ($x$,$y$) of the projection onto X and Y axes can be illustrated as follows:

$$\sqrt{3}c + a + t + \sqrt{3}g = 2x$$

$$c - \sqrt{3}a + \sqrt{3}t - g = 2$$



**(a)** **(b)**

**Fig. 1** The unit vectors represent four nucleotides A, G, C and T. (a) The unit vectors designed by Stephen S. –T. Yau method [4]. (b) The proposed vectors method (UnitX).

## III. THE PROPOSED UNIT X-PROJECTION VECTOR METHOD (UNITX)

According to Fig. 2, by using Yau's method, it is difficult to distinguish the four serotypes of Dengue virus, DENV-1, DENV-2, DENV-3 and DENV-4. Furthermore, the data plotted on the x-axis in Fig. 2 are not the integer numbers (i.e. values on the x-axis are not directly equal to the number of bases) resulting in difficulty for interpretation compared to the convention base-by-base basis (e.g. BLAST). Therefore, we propose the new visualization tool that is more convenient for visualization, interpretation, and analysis of Dengue virus sequences. Our graphical representation represented the distribution of amino/keto base along the sequence on two quadrants of Cartesian coordinate system which the first quadrant was amino (C and A) and keto (T and G) in the fourth quadrant. The vectors represent four nucleotides, i.e. adenines (A), guanine (G), thymine (T), and cytosine (C), are demonstrated as follows:

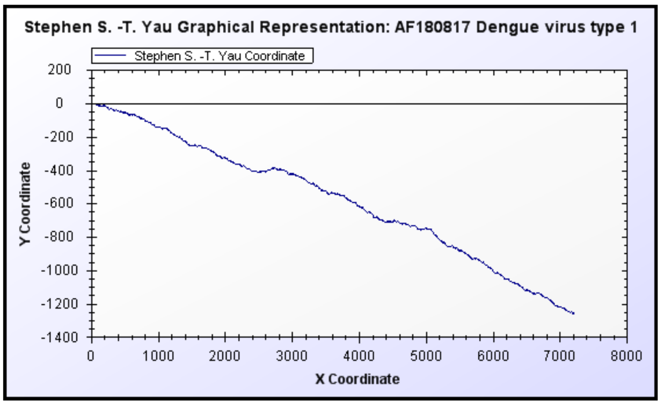
| | | | | | |
|---|---|---|---|---|---|
| A | → | $(1,1)$ | C | → | $(1,2)$ |
| G | → | $(1,-2)$ | T | → | $(1,-1)$ |

By given the numbers of occurring $a$, $c$, $g$, $t$ of bases A, C, G, T, respectively, the coordinate ($x$, $y$) of the projection onto X and Y axes can be illustrated as follows:

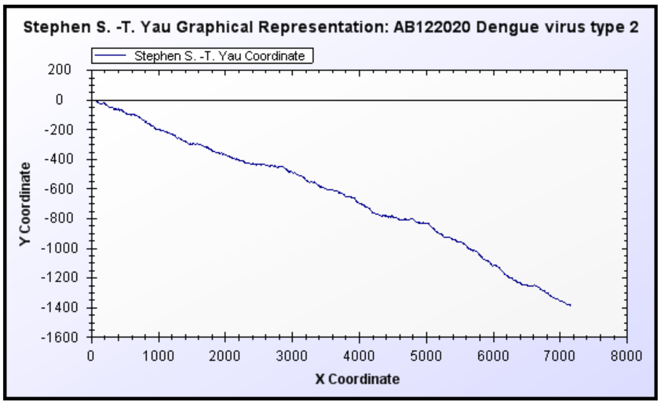
$$c + a + t + g = x$$

$$2c + a - t - 2g = y$$

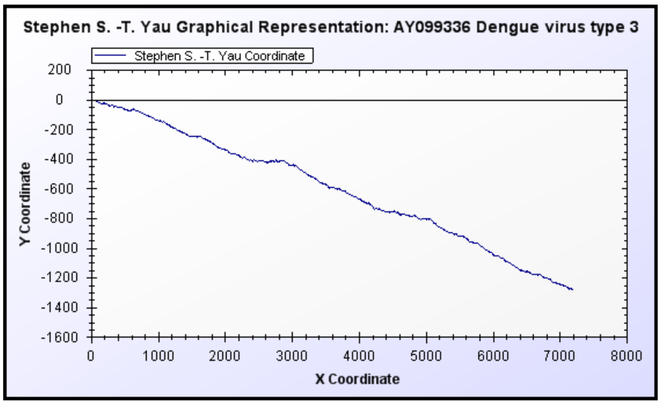
By assigning unity to the size of vectors along x-axis, the graphical representation along x-axis is in the uniform pattern and directly equal to the number of bases. This results in simple visualization for analysis. Figs.3 (a)-(d) show that UnitX yields the representation that can be used to visually distinguish all four serotypes.



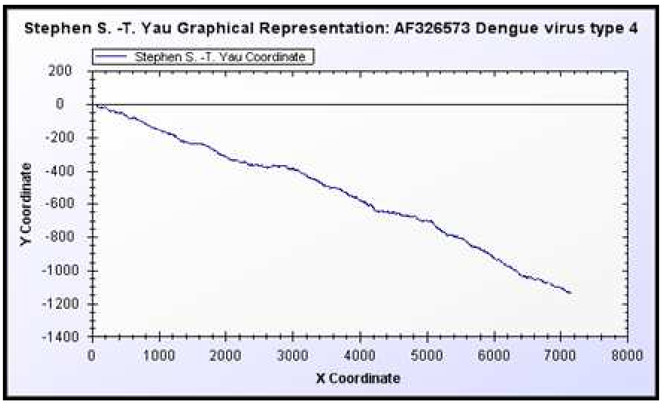
**(a)**



**(b)**

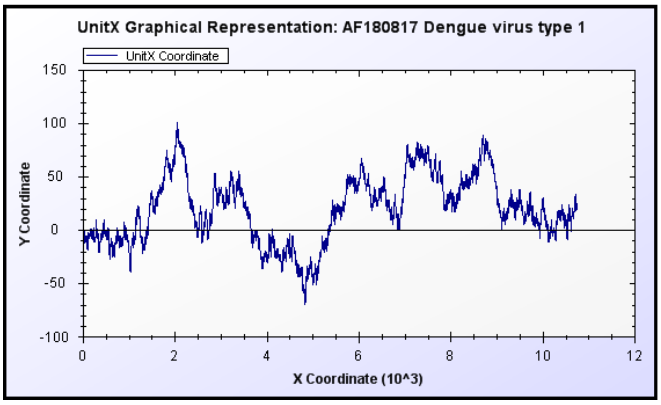


**(c)**

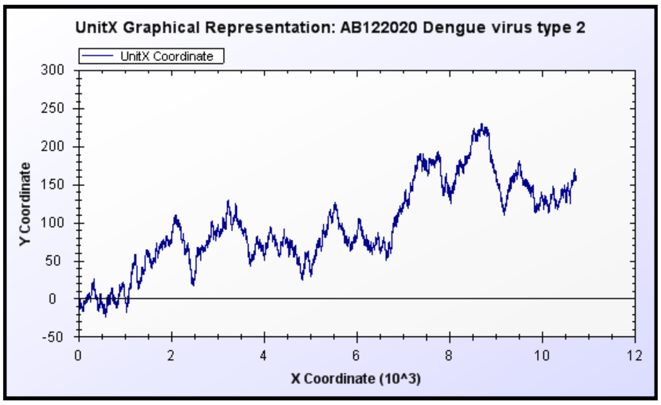


**(d)**

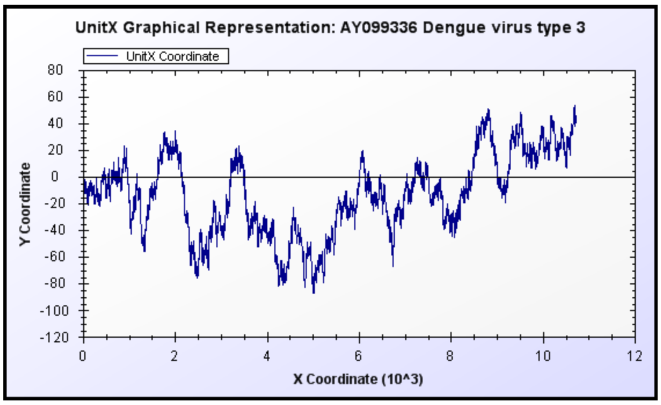
**Fig. 2** Two dimensional graphs of four serotypes of dengue virus complete genome of (a) AF180817 dengue virus type 1, (b) AB122020 dengue virus type 2, (c) AY099336 dengue virus type 3 and (d) AF326573 dengue virus type 4 is generated by Stephen S. –T. Yau's method.



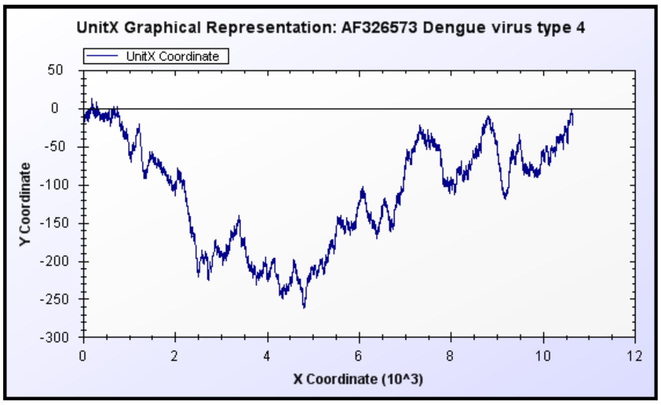
**(a)**



**(b)**



**(c)**



**(d)**

**Fig. 3** Two dimensional graphs of four serotypes of dengue virus complete genome of (a) AF180817 dengue virus type 1, (b) AB122020 dengue virus type 2, (c) AY099336 dengue virus type 3 and (d) AF326573 dengue virus type 4 is generated by our proposed method.

## IV. CLASSIFICATION OF DENGUE VIRUS SEROTYPES VIA UNITX METHOD

Even though, we can visually distinguish the serotypes of some Dengue virus sequences represented by UnitX, automatic method is still needed for the development of the analysis software. The simple idea used in this paper is to find the optimal library (represented by UnitX) to represent each serotype. After that the serotype can be assigned by comparing the correlation coefficient of the selected sequence to the library.

### A. Finding the optimum library

The method used to construct the optimal library of each Dengue virus serotype can be summarized as follows:

**Step1:** Read Dengue virus sequence in FASTA format,

**Step2:** Find $(X_n, Y_n)$ coordinate of each sequence, Let $C_n, A_n, T_n$ and $G_n$ be the cumulative number of bases C, A, T and G, respectively, occurring from the first to the *n*-th base.

$X_n = X_{n-1} + (C_n + A_n + T_n + G_n), n = 1, 2, 3, \dots, N,$

$Y_n = Y_{n-1} + (2C_n + A_n - T_n - 2G_n), n = 1, 2, 3, \dots, N,$

where ($X_i$,$Y_i$ ) are the vector calculated by UnitX Method.

**Step3:** Find $(X_n, Y'_n)$ coordinate of each sequence

$$Y'_n = Y_n - \frac{Y_n}{N},$$

where *N* is the number of base of the whole sequence.

**Step4:** Let $M$ be the number of training sequences for each serotype, the optimal library (the single optimal pattern) of each serotype can be calculated as the centroid $(X'', Y'')$ of $\left(X_n^{(m)}, Y'^{(m)}_n\right)$ for all *m*= 1, 2, 3,..., *M* and *n* = 1, 2, 3,…, *N*

$X'' = \frac{1}{M}\sum_{m=0}^{M} X_n^{(m)},$

$$Y'' = \frac{1}{M}\sum_{m=0}^{M} Y'^{(m)}_n,$$

where $X_n^{(m)}$ and $Y'^{(m)}_n$ denote the coordinate (*x*, *y*) of the *m*-th training sequence for each serotype.

**B. Summarization of the classification method**

This proposed UnitX method can be used to automatically assign the serotype to the unknown Dengue virus sequences. The classification process can be summarized as follows:

**Step1:** Read dengue virus sequence in FASTA format,

**Step2:** Follow Step 2 and Step 3 of section IV *A*,

**Step3:** Compare the output of $Y'$ with the libraries $Y''$ of all four serotypes by measuring their similarities via the correlation coefficients [15]-[17]. The unknown sequence will belong to the serotype that yields the maximum correlation coefficient.

## V. SIMULATION RESULTS

To verify our proposed graphical representation method, we employ UnitX to distinguish different serotypes of Dengue virus. The sample sequences are obtained from NCBI with the keyword of Dengue virus complete genome. All 877 sample sequences compose of four serotypes of Dengue virus sequences (each serotype contains 196, 345, 246 and 90 sample sequences, respectively). In this study, we measure the performance of our software by 4-fold cross validation. The performance of our software is shown in Table1. We can obviously see that UnitX can be efficiently used as features to classify Dengue virus sequence.

**TABLE I** Summary of prediction results

| Round | # of Sample Sequences | Classification Accuracy |
|---|---|---|
| **1** | **657** | **99.09 %** |
| **2** | **658** | **99.54 %** |
| **3** | **658** | **99.39 %** |
| **4** | **658** | **99.39 %** |
| | | **99.35 %** |

## VI CONCLUSION

In this paper, we have presented the new graphical representation of genomic sequence that can provide both the physical meaning and easy-to-understand graphical representation. The proposed method called UnitX has been used to classify the four serotypes of dengue virus. By designing the optimal libraries, UnitX can be perfectly used as the visualization tool for Dengue virus analysis software since it can efficiently assign the serotype for the selected Dengue virus sequence.

## REFERENCES

[1] Hamori E and Ruskin J. “H curves, a novel method of representation of nucleotide series especially suited for long DNA sequences”. The Journal of Biological Chemistry. 1983; 258(2):1318-1327.

[2] Gates M.A. “Simple DNA sequence representations”, Nature, vol. 316, 1985.Nandy A. “Two-dimensional graphical representation of DNA sequences and intron-exon discrimination in intron-rich sequences”. Bioinformatics. 1996; 12(1):55-62.

[3] Yau SS-T, Wang J, Niknejad A, Lu C, Jin N and Ho Y-K. “DNA sequence representation without degeneracy”. Nucleic Acids Research. 2003; 31(12):3078-3080.

[4] Altschul SF, Miller W, Myers E and Lipman DJ. “Basic local alignment search tool”. Journal of Molecular Biology. 1990; 215(3):403–410.

[5] Ye J, McGinnis S and Madden TL. “BLAST: improvements for better sequence analysis”. Nucleic Acids Research. 2006; 34:W6-W9.

[6] Anastassiou D. “Genomic signal processing”. Signal Processing Magazine, IEEE. 2001; 18(4):8-20.

[7] Law N-F, Cheng K-O and Siu W-C. “On relationship of Z-curve and Fourier approaches for DNA coding sequence classification”. Bioinformation. 2006; 1(7):242-246.

[8] Zhang R and Zhang C-T. “Z curves, an intuitive tool for visualizing and analyzing the DNA sequences”. Journal of Biomolecular Structure & Dynamics. 1994; 11:767-782.

[10] Zhang C-T, Wang J and Zhang R. “A novel method to calculate the G+C content of genomic DNA sequences” Journal of Biomolecular Structure & Dynamics. 2001; 19(2):333-341.

[11] Zhang C-T, Zhang R and Ou H-Y. “The Z curve database: a graphic representation of genome sequences” Bioinformatics. 2003; 19(5):593-599.

[12] Yan-ling Y, Ji-hua W, Jia-feng Y and Gui-zhong L, editors. “An Analysis of Non-Coding RNA Using Z-Curve Method” Bioinformatics and Biomedical Engineering, 2008 ICBBE 2008 The 2nd International Conference on; 2008.

[13] Cai JJ, Smith DK, Xia X and Yuen K-y. “MBEToolbox: a Matlab toolbox for sequence data analysis in molecular biology and evolution” BMC Bioinformatics. 2005; 6(44).

[14] Zhang Y and Tan M. “Visualization of DNA sequences based on 3DD-Curves” Journal of Mathematical Chemistry. 2008; 44(1):206-216.

[15] Efrat A, Fan Q and Venkatasubramanian S. “Curve Matching, Time Warping, and Light Fields: New Algorithms for Computing Similarity between Curves” Journal of Mathematical Imaging and Vision. 2007; 27(3):203-216.

[16] Zhu L, Zhou Z and Zhang J. “A Partial Curve Matching Method for Automatic Reassembly of 2D Fragments”. Springerlink. 2006; 345:645-650.

[17] Porrill J and Pollard S. “Curve matching and stereo calibration” Image and Vision Computing. 1991; 9(1):45 – 50.

# CADViST: VISUALIZATION TOOL FOR BLAST ALIGNMENT OF DENGUE VIRUS SEQUENCES

Boonyarat Viriyasaksathian[1], Yodchanan Wongsawat[1] and Prapat Suriyaphol[2]

[1]Department of Biomedical Engineering, Mahidol University, Nakornpathom, Thailand,
[2]Bioinformatics and Data Management for Research Unit, Office for Research and Development, Faculty of Medicine Siriraj Hospital, Mahidol University, Bangkok, Thailand

## ABSTRACT

*Exploration of the search engine that can simultaneously visualize the genomic sequences is one of the challenging problems. In this paper, we propose the software, called CADViST. The UnitX graphical representation (previously proposed by the authors) is employed as the alternative tool to visualize the result obtained from the Basic Local Alignment Search Tool (BLAST). The proposed software can efficiently help the users/experts to easily interpret the results, especially in Dengue virus sequence analysis where different serotypes or subtypes need to be distinguished.*

## I. INTRODUCTION

In bioinformatics, the Basic Local Alignment Search Tool (BLAST) is one of the most widely used tools for sequence similarity search due to its speed and reasonable accuracy of searching performance. However, the BLAST program is still lacked of user friendly graphical representation. Hence, in this paper, we aim to develop a visualization tool that is capable to display the text output resulting from BLAST.

There are many existing tools used for visualizing and analyzing the genomic sequences. Each tool is developed based on some specific tasks which can be categorized into four approaches, i.e. Base vector, Sequential, Fourier Transform (FT) and Z-Curve approaches.

**(1) Base vector approach:** Hamori, E. and Ruskin, J. (1983) represented DNA sequences in a three dimensional curve (H- Curve) [1]. Gates, M.A. (1985) proposed that graphical representation of DNA sequence in two dimensional space was better than H-Curve. Gates' graphical representation shows four nucleotide bases, i.e. adenine (A), thymine (T), cytosine (C), and guanine (G). The unit vector representations of these bases are on the Cartesian coordinate system, i.e. Base A is on the negative *y*-axis, base T is on the positive *y*-axis, base G is on the positive *x*-axis, and base C is on the negative *x*-axis [2]. About eleven years later, Nandy A. (1996) proposed a graphical representation in order to distinct the features of intron and exon segments of eukaryotic sequences [3]. This graphical representation was similar to Gates' method. The A, G, C and T nucleotide was plotted on an ACGT-axis system. The slope of this plot indicated a cluster of intron and exon sequences. However, both Nandy and Gates' methods have high degeneracy such that the sequences such as AGTC, AGTCA, and AGTCAG lead to the same graphical representation [4]. Stephen S. -T. Yau *et al*., 2003 modified Gates' method. The four nucleic acids are classified into pyrimidine/purine graph on two quadrants of the Cartesian coordinate system. The first quadrant represents pyrimidine (T and C), and the forth quadrant represents purine (A and G) [4]. Recently, the authors propose the graphical representation especially for the Dengue virus sequence analysis based on the cumulative amount of amino and keto bases, called UnitX [5].

**(2) Sequential approach:** Altschul *et al*., 1990 developed the Basic Local Alignment Search Tool (BLAST) program. This program is one of the most popular tools for genomic sequence analysis. This tool can perform a fast similarity search. The program compares the similarity between any two sequences and displays the difference between these sequences by comparing in the base-by-base basis [6].

**(3) Fourier Transform (FT) approach:** Anatassiou D. proposed the color spectrograms of biomolecular sequences which is the tool used for visualization of the biomolecular sequence analysis [7], [8]. Spectrograms which can represent the magnitude of the short-time Fourier transform (STFT) is implemented via the discrete Fourier transform (DFT). Analysis of the genomic sequence in frequency domain via the Fourier transform (FT) uses the 3-periodicity property for DNA coding sequence. The color spectrogram is defined by using the color: red, green and blue. Even though this method yields an impressive graphical representation, the computational complexity is fairly high.

**(4) Z-Curve approach:** Zhang C. T. *et al*., 1994 suggested a practical visualization tool called Z-Curve [8]**-**[12]. James J. *et al* developed this tool in the package called MBEToolbox [13]. According to the assumption on the cumulative components of the genomic sequence, features obtained from Z-Curve can be quickly interpreted, such as the distribution along the sequence of purine/pyrimidine bases, amino/keto bases, strong H-bond/weak H-bond. Since the algorithm of Z-Curve is simple, it can be applied to all genomic sequences regardless of how long those sequences are. The similar approach with Z-Curve called 3DD-Curve is presented by Zhang Y. and Tan M. (2008). This approach can be viewed as the weighted version of Z-Curve [14].

The choice of selecting the graphical representation can vary based on the characteristics of genomic sequences of interest. Therefore, in this first version of the proposed software, Dengue virus sequences (nucleotide sequences) are employed to verify the merit of the proposed software. The software is called CADViST which stands for **C**lassification and **A**nalysis of **D**engue **Vi**rus **S**erotype by Visualization **T**ool. By employing UnitX as the visualization tool, the proposed software is suitable to use for interpreting the Dengue virus sequence. However, positioning of partial Dengue sequences on Dengue genome with UnitX representation requires high computational load. BLAST is well known as the efficient searching tool. However, visualizing the results obtained from BLAST needs some improvement. Therefore, in this paper, we propose the software that combines the merit of both BLAST and UnitX. The proposed software can efficiently search the unknown portion of Dengue virus sequences and can simultaneously illustrate graphical representations of the resulting sequences.

This paper can be organized as follows. Section II introduces the proposed visualization tool, called CADViST. The software architecture of CADViST is described in Section III. In Section IV, the simulation results of the proposed software are shown. Finally, Section V concludes the paper.

## II CADVIST: THE PROPOSED VISUALIZATION TOOL

**C**lassification and **A**nalysis of **D**engue **Vi**rus **S**erotype by Visualization **T**ool, or CADViST, is a visualization tool proposed especially for analyzing the Dengue virus sequences. All components and details of CADViST can be described in details as follows:

### *A. Basic Local Alignment Search Tool (BLAST)*

BLAST program is developed by Stephen F. Altchul and his coworkers at the National Center for Biotechnology Information (NCBI). It is widely used for calculating the sequence similarity. BLAST works through the heuristic algorithm to find the best possible results. It finds the homologous sequences by locating short matches between two sequences to make the search fast. Similarity measurement technique of BLAST uses statistical theory to assign a scoring matrix for all possible pairs of residues and produce the Expect value (E-value) for each alignment pair.

The stand-alone BLAST programs are provided as a compressed package. The package, available as BLAST initialed archives for a variety of computer platform, is available on the BLAST ftp site: ftp://ftp.ncbi.nih.gov/ blast/executables/release/. In this paper, we employed stand-alone BLAST version 2.2.22 to generate BLAST output, as input of the proposed software (CADViST).

### *B. UnitX Graphical Representation*

UnitX graphical representation can efficiently reveal the distribution of amino/keto bases along the sequence on two quadrants of the Cartesian coordinate system. The first quadrant represents the amount of amino (C and A) while the fourth quadrant represents amount of keto (T and G). The unit vectors represent four nucleotides, i.e. adenines (A), guanine (G), thymine (T), and cytosine (C), are demonstrated as follows (Fig. 1):

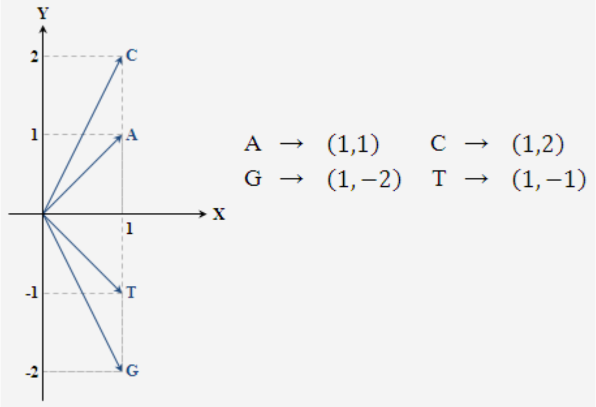


**Figure 1.** The UnitX vectors represent four nucleotides A, G, C and T.

By assigning the numbers of occurring of bases A, C, G, and T in the sequences, the coordinate (x, y) of the projection onto X and Y axes with UnitX representation can be illustrated as follows:

$$C + A + T + G = x$$

$$2C + A - T - 2G = y$$

### *C. Idea of CADViST*

In this paper, we employ BLAST in a “stand-alone” mode to find the similarity score among the query sequence and the Dengue virus nucleotide database. The search results obtained from BLAST are graphically displayed via UnitX representation.

### *D. Creating nucleotide BLAST database*

The main advantage of stand-alone BLAST program is to be able to create your own database. To create a nucleotide BLAST database, we need a source file of sequence in FASTA format. This file will be processed by the formatdb program contained within the stand-alone BLAST package to build index files of the database. After executing formatdb command, three files will be produced from the source FASTA file. For nucleotide databases, the extensions are nhr, nin, and nsq [15]. The formatdb command can be shown as follows:

formatdb -p F -i **DatabaseName.fasta**

The source FASTA file will have the form:

```
>First sequence description
XXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXX…
>Second sequence description
XXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXX…
>Last sequence description
XXXXXXXXXXXXXXXXXXXXXXXX
XXXXXXXXXXXXXX…
```

where Xs are nucleotide codes (A, T, G or C).

In this paper, the database of the proposed software is obtained from NCBI with the keyword of Dengue virus complete genome. All 2,184 nucleotide sequences compose of four serotypes of Dengue virus sequences (each serotype contains 952, 737, 405 and 90 nucleotide sequences, respectively).

**E. Stand-alone executable BLAST**

The stand-alone executable BLAST and NCBI web-based BLAST program provide easy ways for users to perform BLAST search via command line or a website. There are many advantages to run BLAST search program on your own machine, e.g. database can be easily edited. In this paper, we employ stand-alone BLAST program to generate BLAST output. BLAST search can be executed via blastall command as follow:

blastall **-**p blastn **-**d **Databasename.fasta -**i **QuerySequence.fasta -**m 9 **-**F F> **Result.txt**

**F. Graphical Representation via UnitX**

In steady of displaying the search results in alphabets (Figs. 4(b) and 4(c)) like BLAST, CADViST extracts the information from BLAST and represents the results graphically via UnitX representation described section II B. Furthermore, in the case that the users only need to explore the nature of Dengue virus sequences, they can also employ only the graphical feature (UnitX) of CADViST.

## III. SOFTWARE ARCHITECTURE OF CADViST

To develop the user friendly GUI, the proposed CADViST software is written in C# programming. The GUI of CADViST can be shown in Fig.3. The input fields for query sequence can be either (1) the text file in FASTA format or (2) text letter directly copied and put into the blank space in Fig.3. Once the input is inserted, the process inside CADViST can be summarized as follows (Fig. 2):

**Step1:** Call stand-alone BLAST program to generate BLAST output,

**Step2:** Extract sequence accession number and the coordinates of each matched sequence from BLAST output,

**Step3:** Provide matching regions between query and matched sequence identified by BLAST program and send the results to the display unit, i.e. UnitX representation. The results are shown in Figs.4 (d-e).

In addition, other options of CADViST are copy, save, print, show point values in the graph of UnitX vector. The option can be selected by making a right click on the graph.

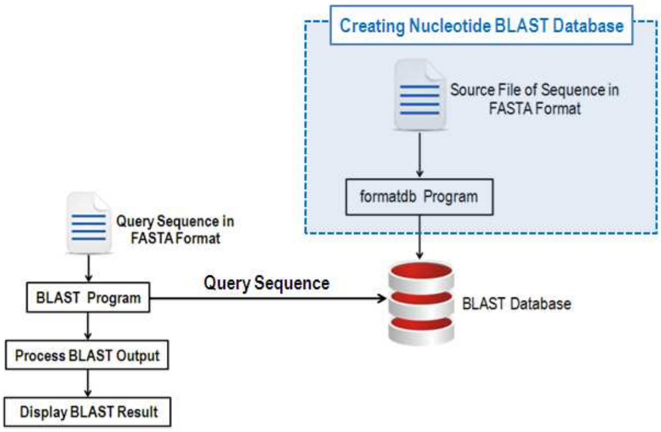
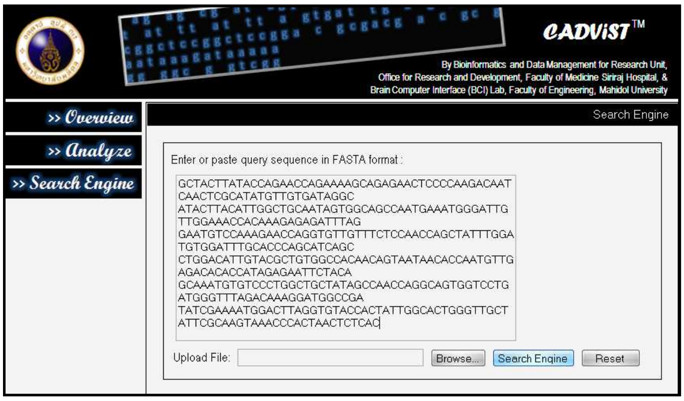

Creating Nucleotide BLAST Database
Source File of Sequence in FASTA Format
formatdb Program
Query Sequence in FASTA Format
BLAST Program
Query Sequence
BLAST Database
Process BLAST Output
Display BLAST Result

**Figure 2.** Flowchart of the proposed software


CADViST™
By Bioinformatics and Data Management for Research Unit, Office for Research and Development, Faculty of Medicine Siriraj Hospital, & Brain Computer Interface (BCI) Lab, Faculty of Engineering, Mahidol University
» Overview
» Analyze
» Search Engine
Search Engine
Upload File:
Browse...
Search Engine
Reset

Enter or paste query sequence in FASTA format :

```
GCTACTTATACCAGAACCAGAAAAGCAGAGAACTCCCCAAGACAAT
CAACTCGCATATGTTGTGATAGGC
ATACTTACATTGGCTGCAATAGTGGCAGCCAATGAAATGGGATTG
TTGGAAACCACAAAGAGAGATTTAG
GAATGTCCAAAGAACCAGGTGTTGTTTCTCCAACCAGCTATTTGGA
TGTGGATTTGCACCCAGCATCAGC
CTGGACATTGTACGCTGTGGCCACAACAGTAATAACACCAATGTTG
AGACACACCATAGAGAATTCTACA
GCAAATGTGTCCCTGGCTGCTATAGCCAACCAGGCAGTGGTCCTG
ATGGGTTTAGACAAAGGATGGCCGA
TATCGAAAATGGACTTAGGTGTACCACTATTGGCACTGGGTTGCT
ATTCGCAAGTAAACCCACTAACTCTCAC
```

**Figure 3.** Screenshots of the proposed software

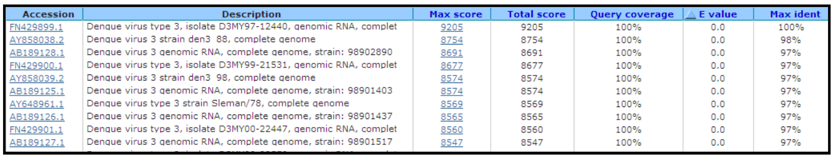
## IV. SIMULATION RESULTS

As an example, we verify the merit of CADViST for finding the similarities among FN429899 Dengue virus serotype3 (2040 – 7143 base position) and our Dengue virus sequence database.

Traditionally, the results obtained from stand-alone BLAST program consist of two major parts, i.e. (1) the one-line descriptions of each database sequence found to match the query sequence (Fig.4 (a)), and (2) the alignment between the input sequence and the matched query sequences (Figs.4 (b)-(c)) [16]. Figs. 4 (b) and (c) illustrate the first and second highest score matched sequences, respectively.

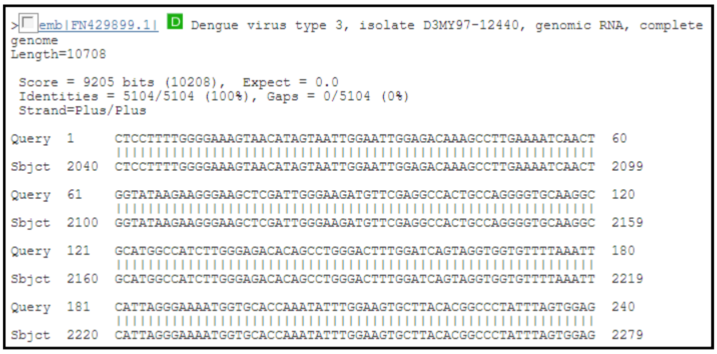
By employing the information obtained from BLAST, Figs.4 (d)-(e) represent the proposed graphical representation via CADViST. The result of the proposed software consists of two main parts where each part displays the graphical representation via UnitX. The first part shows the whole genome of query sequence (Fig.4 (d)). The second part displays the matched regions between the query and input sequence identified by BLAST (Fig.4 (e)). In Fig.4 (e), for convenience, only the first (FN429899) and second (AY858038) highest scores matched sequences are shown. Both sequences are also from the same serotype as our input sequence. As expected, the first highest score matched sequence is the sequence that we copy its portion as our query sequence.

Furthermore, according to Figs.4 (a – c), we can also observe that the output of BLAST still lacks of user friendly graphical representation. Therefore, CADViST can efficiently be one of the alternative way to visualize the resulting sequences obtained from BLAST as shown in Figs.4 (d – e). In Fig.4 (e), we can obviously observe the region of the mismatched base pairs. The result of the proposed software can be displayed via the graph overlaying format together with the UnitX representation of the sequences (Fig.4 (d)).



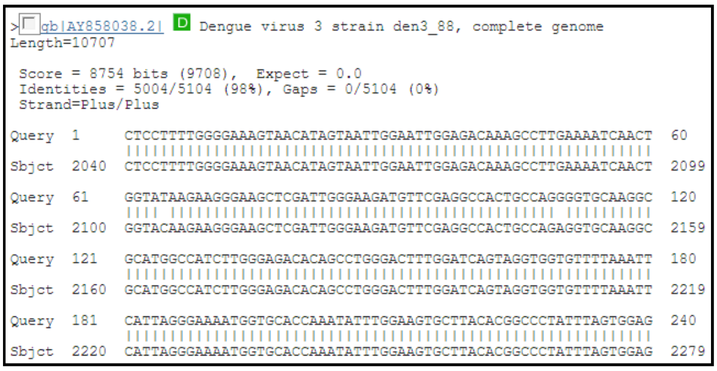
| Accession | Description | Max score | Total score | Query coverage | E value | Max ident |
|---|---|---|---|---|---|---|
| FN429899.1 | Dengue virus type 3, isolate D3MY97-12440, genomic RNA, complet | 9205 | 9205 | 100% | 0.0 | 100% |
| AY858038.2 | Dengue virus 3 strain den3 88, complete genome | 8754 | 8754 | 100% | 0.0 | 98% |
| AB189128.1 | Dengue virus 3 genomic RNA, complete genome, strain: 98902890 | 8691 | 8691 | 100% | 0.0 | 97% |
| FN429900.1 | Dengue virus type 3, isolate D3MY99-21531, genomic RNA, complet | 8677 | 8677 | 100% | 0.0 | 97% |
| AY858039.2 | Dengue virus 3 strain den3 98, complete genome | 8574 | 8574 | 100% | 0.0 | 97% |
| AB189125.1 | Dengue virus 3 genomic RNA, complete genome, strain: 98901403 | 8574 | 8574 | 100% | 0.0 | 97% |
| AY648961.1 | Dengue virus type 3 strain Sleman/78, complete genome | 8569 | 8569 | 100% | 0.0 | 97% |
| AB189126.1 | Dengue virus 3 genomic RNA, complete genome, strain: 98901437 | 8565 | 8565 | 100% | 0.0 | 97% |
| FN429901.1 | Dengue virus type 3, isolate D3MY00-22447, genomic RNA, complet | 8560 | 8560 | 100% | 0.0 | 97% |
| AB189127.1 | Dengue virus 3 genomic RNA, complete genome, strain: 98901517 | 8547 | 8547 | 100% | 0.0 | 97% |

**(a)**



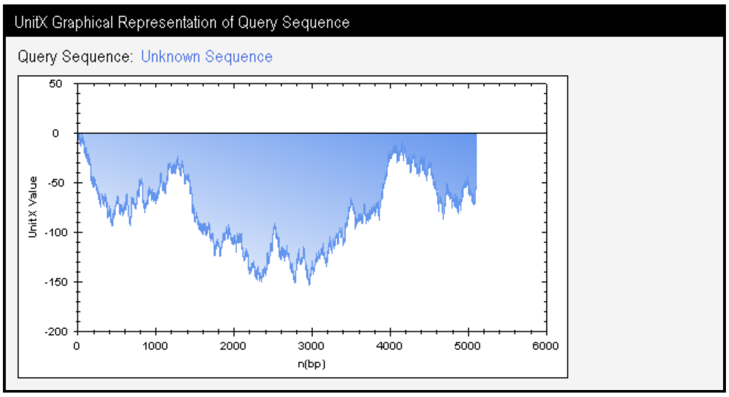
```
>□emb|FN429899.1| D Dengue virus type 3, isolate D3MY97-12440, genomic RNA, complete
genome
Length=10708

 Score = 9205 bits (10208),  Expect = 0.0
 Identities = 5104/5104 (100%), Gaps = 0/5104 (0%)
 Strand=Plus/Plus

Query  1     CTCCTTTTGGGGAAAGTAACATAGTAATTGGAATTGGAGACAAAGCCTTGAAAATCAACT  60
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  2040  CTCCTTTTGGGGAAAGTAACATAGTAATTGGAATTGGAGACAAAGCCTTGAAAATCAACT  2099

Query  61    GGTATAAGAAGGGAAGCTCGATTGGGAAGATGTTCGAGGCCACTGCCAGGGGTGCAAGGC  120
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  2100  GGTATAAGAAGGGAAGCTCGATTGGGAAGATGTTCGAGGCCACTGCCAGGGGTGCAAGGC  2159

Query  121   GCATGGCCATCTTGGGAGACACAGCCTGGGACTTTGGATCAGTAGGTGGTGTTTTAAATT  180
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  2160  GCATGGCCATCTTGGGAGACACAGCCTGGGACTTTGGATCAGTAGGTGGTGTTTTAAATT  2219

Query  181   CATTAGGGAAAATGGTGCACCAAATATTTGGAAGTGCTTACACGGCCCTATTTAGTGGAG  240
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  2220  CATTAGGGAAAATGGTGCACCAAATATTTGGAAGTGCTTACACGGCCCTATTTAGTGGAG  2279
```

**(b)**



```
>□gb|AY858038.2| D Dengue virus 3 strain den3_88, complete genome
Length=10707

 Score = 8754 bits (9708),  Expect = 0.0
 Identities = 5004/5104 (98%), Gaps = 0/5104 (0%)
 Strand=Plus/Plus

Query  1     CTCCTTTTGGGGAAAGTAACATAGTAATTGGAATTGGAGACAAAGCCTTGAAAATCAACT  60
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  2040  CTCCTTTTGGGGAAAGTAACATAGTAATTGGAATTGGAGACAAAGCCTTGAAAATCAACT  2099

Query  61    GGTATAAGAAGGGAAGCTCGATTGGGAAGATGTTCGAGGCCACTGCCAGGGGTGCAAGGC  120
             |||| |||||||||||||||||||||||||||||||||||||||||||| ||||||||||
Sbjct  2100  GGTACAAGAAGGGAAGCTCGATTGGGAAGATGTTCGAGGCCACTGCCAGAGGTGCAAGGC  2159

Query  121   GCATGGCCATCTTGGGAGACACAGCCTGGGACTTTGGATCAGTAGGTGGTGTTTTAAATT  180
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  2160  GCATGGCCATCTTGGGAGACACAGCCTGGGACTTTGGATCAGTAGGTGGTGTTTTAAATT  2219

Query  181   CATTAGGGAAAATGGTGCACCAAATATTTGGAAGTGCTTACACGGCCCTATTTAGTGGAG  240
             ||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
Sbjct  2220  CATTAGGGAAAATGGTGCACCAAATATTTGGAAGTGCTTACACGGCCCTATTTAGTGGAG  2279
```

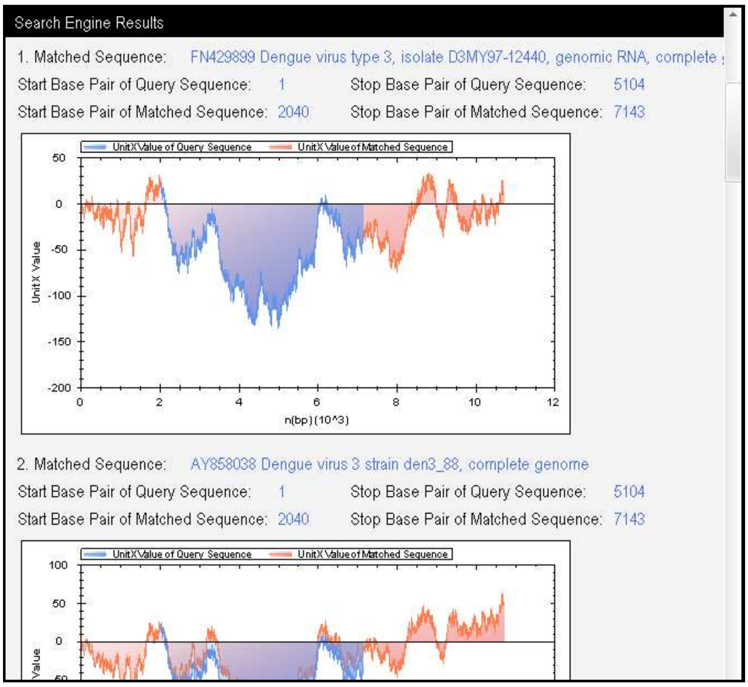
**(c)**



UnitX Graphical Representation of Query Sequence

Query Sequence: Unknown Sequence

**(d)**



Search Engine Results

1. Matched Sequence: FN429899 Dengue virus type 3, isolate D3MY97-12440, genomic RNA, complete

Start Base Pair of Query Sequence: 1 Stop Base Pair of Query Sequence: 5104

Start Base Pair of Matched Sequence: 2040 Stop Base Pair of Matched Sequence: 7143

UnitX Value of Query Sequence
UnitX Value of Matched Sequence
50
0
-50
-100
-150
-200
UnitX Value
0
2
4
6
8
10
12
n(bp) (10^3)

2. Matched Sequence: AY858038 Dengue virus 3 strain den3_88, complete genome

Start Base Pair of Query Sequence: 1 Stop Base Pair of Query Sequence: 5104

Start Base Pair of Matched Sequence: 2040 Stop Base Pair of Matched Sequence: 7143

UnitX Value of Query Sequence
UnitX Value of Matched Sequence
100
50
0
-50
Value

**(e)**

**Figure 4.** (a) One-line description in the BLAST report; (b-c) A pairwise sequence alignment from a BLAST report; (d-e) Graphical display of the CADViST.

## V. CONCLUSION

In this paper, we have developed the software called CADViST. The proposed software can be used to visually analyze the matched regions identified by BLAST between the query sequences and the Dengue virus database. Graphical representation is implemented via UnitX which is suitable especially for analyzing different serotypes of Dengue virus nucleotide sequences. Many options in CADViST can also benefit the bioinformatics experts, e.g. save, print, and show the raw numeric values on the graph. With the .net framework of C#, CADViST can be easily modified to include more open source or in house developed mathematical modeling, while maintaining the user friend GUI.

## REFERENCES

[1] E. Hamori and J. Ruskin, “H curves, a novel method of representation of nucleotide series especially suited for long DNA sequences”. The Journal of Biological Chemistry, vol. 258(2), 1983, pp. 1318-1327.

[2] M.A. Gates, “Simple DNA sequence representations”, Nature, vol. 316, 1985.

[3] A. Nandy, “Two-dimensional graphical representation of DNA sequences and intron-exon discrimination in intron-rich sequences”. Bioinformatics, vol. 12(1), 1996, pp. 55-62.

[4] S.-T. Yau, J. Wang, A. Niknejad, C. Lu, N. Jin and Y-K. Ho, “DNA sequence representation without degeneracy”, Nucleic Acids Research, vol. 31(12), 2003, pp. 3078-3080.

[5] B. Viriyasaksathian, Y. Wongsawat and P. Suriyaphol, “UnitX: Dengue virus sequence graphical representation for serotypes classification”, ISBME 2009, Bangkok, Thailand.

[6] S.F. Altschul, W. Miller, E. Myers and D.J. Lipman, “Basic local alignment search tool”, Journal of Molecular Biology, vol. 215(3), 1990, pp. 403-410.

[7] J. Ye, S. McGinnis and T.L. Madden, “BLAST: improvements for better sequence analysis”, Nucleic Acids Research, vol. 34, 2006, pp. W6-W9.

[8] D. Anastassiou, “Genomic signal processing”, Signal Processing Magazine, IEEE, vol. 18(4), 2001, pp. 8-20.

[9] N-F. Law, K-O. Cheng and W-C. Siu, “On relationship of Z-curve and Fourier approaches for DNA coding sequence classification”, Bioinformation, vol. 1(7), 2006, pp. 242-246.

[10] R. Zhang and C-T. Zhang, “Z curves, an intutive tool for visualizing and analyzing the DNA sequences”, Journal of Biomolecular Structure & Dynamics, vol. 11, 1994, pp.767-782.

[11] C-T. Zhang, J. Wang and R. Zhang, “A novel method to calculate the G+C content of genomic DNA sequences”, Journal of Biomolecular Structure & Dynamics, vol. 19(2), 2001, pp. 333-341.

[12] C-T. Zhang, R. Zhang and H-Y. Ou, “The Z curve database: a graphic representation of genome sequences”, Bioinformatics, vol. 19(5), 2003, pp. 593-599.

[13] J.J. Cai, DK. Smith, X. Xia and K-Y. Yuen, “MBEToolbox: a Matlab toolbox for sequence data analysis in molecular biology and evolution”, BMC Bioinformatics, vol. 6(44), 2005.

[14] Y. Zhang and M. Tan, “Visualization of DNA sequences based on 3DD-Curves”, Journal of Mathematical Chemistry, vol. 44(1), 2008, pp. 206-216.

[15] D. Wheeler, “NCBI News: How to Search Custom Subsets of GenBank Using Standalone BLAST”, National center for biotechnology information, pp. 6, Winter 1999.

[16] T. Madden, “The BLAST sequence analysis tool”, The NCBI handbook, this article is available from http://www.ncbi.nlm.nih.gov/bookshelf/br.-fcgi?book=handbook&part=ch16.

# BIOGRAPHY

| | |
|---|---|
| **NAME** | Boonyarat. Viriyasaksathian |
| **DATE OF BIRTH** | 13 January 1986 |
| **PLACE OF BIRTH** | Songkla, Thailand |
| **INSTITUTIONS ATTENDED** | Prince of Songkla University, 2004-2008:<br>Bachelor of Engineering<br>(Computer Engineering)<br>Mahidol University, 2008-2010:<br>Master of Engineering<br>(Biomedical Engineering) |
| **HOME ADDRESS** | 594 Klong Rean1 Road, Hat Yai District,<br>Hat Yai, Songkla, Thailand<br>E-mail: g51137363@student.mahidol.ac.th |
| **PUBLICATION / PRESENTATION** | "UnitX: Dengue Virus Sequence Graphical Representation for Serotypes Classification", in Proc., International Symposium on Biomedical Engineering (ISBME), Bangkok, Thailand, Dec. 2009<br>"CADViST: Visualization Tool for BLAST Alignment of Dengue Virus Sequences", in Proc., The 4th International Conference on Bioinformatics and Biomedical Engineering (iCBBE 2010), Chengdu, China, June 2010 |