Similarity Score Estimation and Gaps Trimming of Multiple Sequence Alignment for Phylogenetic Tree Analysis

Kasikrit Damkliang[1], **Pichaya Tandayya**[2], **Unitsa Sangket**[3], and **Ekawat Pasomsub**[4]

ABSTRACT

Phylogenetic tree analysis is a process for finding the highest possible revolution tree history of an interested organism. The important step of the process is multiple sequences alignment (MSA) which is operated using any MSA tool that produces a result in blocks of the Phylip format. Bioinformaticians have to manually determine and trim gaps of the MSA blocks using relevant tools of a software package in the off-line mode. The data blocks need to be manually cut-and-pasted between these tools. This working steps tend to be error-prone and time consuming. In addition, improper algorithm selection for tree inferring without applying an MSA similarity score tends to generate the phylogenetic tree with low accuracy and also take much more time. In this work, we present an automatic approach for the phylogenetic tree analysis applying our enhancement for the similarity score estimation and gaps trimming of the MSA blocks. We propose *in-silico* algorithms for automating the concerned similarity score estimation and gaps trimming, and deploy them as web services. We demonstrate the web services utilized by composing them into an integrated stateful WSDL workflow. Our case study datasets are a complete coding sequences (CDS) and sets of complete genome of Dengue Viruses - 2, fetched from the NCBI RefSeq nucleotide database. Our proposed algorithms have correctly returned results, verified and satisfied by our bioinformaticians. Our distributions, user manuals and endpoints of the web services, and the open source programs are available at `http://bioservices.sci.psu.ac.th`.

Keywords: Multiple Sequence Alignment (MSA), Similarity Score, Gaps Trimming, Tree Inferring, Web Service

1. INTRODUCTION

Phylogenetic tree analysis is a process for finding the highest possible evolution tree history of an interested organism [1-3]. Nowadays, large complete Coding Sequences (CDS) and large collections of genomics regions of an organism have been being discovered and revealed frequently. Currently, it is the beginning era of phylogenomics which researchers are expanding the evolution study at a genomic scale [4-6]. Software, analysis processes, and related tools have also been continuously developed and upgraded simultaneously. In bioinformatics, although there are many tools, websites, and web services, a lot of bioinformaticians still have to work manually as there still remain many obstacles and there are not enough interfaces due to too many different formats and implementations involved. Cutting and pasting between websites, tools, and services usually induce errors and mistakes, especially in the phylogenetic tree analysis.

There are many conventional software packages and web services utilized for the phylogenetic tree analysis such as PHYLIPNEW applications [7-8], MEGA [9], European Bioinformatics Institute (EMBL-EBI) services [10-11], Swiss Institute of Bioinformatics (SIB) services [12], and cooperation related-tools, for example, an R interface for PHYLIP [13], analytic tools for the phylogenetic trees based on powered grid resources [14] and the integrated platform of response web tools for expert and non-expert users [15].

The phylogenetic tree analysis usually is decomposed into three main steps consisting of multiple sequence alignment (MSA), tree inferring, and tree visualization. The tree inferring is a significant process for estimating the phylogenetic tree. Its input is the MSA result in the Phylip [16] format. In this paper, we focus on the popular tree inferring algorithms bundled with the PHYLIPNEW [7] applications, EMBOSS [8] conversions of the program in Joe Felsenstein's PHYLIP package which consists of Parsimony (PARS) [17], Distance Matrix - Neighbor Joining (DM-NJ) [18], and Maximum Likelihood (ML) [19]. These inferring algorithms support both nucleotide and amino acid sequences. By the way, for general analysis of tree inferring steps, bioinformaticians usually remove some gaps of sequences of the

Manuscript received on January 13, 2017 ; revised on July 24, 2017.

Final manuscript received on August 21, 2017.

[1,2] The authors are with the Department of Computer Engineering, Faculty of Engineering, Prince of Songkla University, Hat Yai, Songkhla, Thailand 90112, E-mail: kasikrit.d@psu.ac.th, pichaya@coe.psu.ac.th

[3] The author is with the Center for Genomics and Bioinformatics Research, Faculty of Science, Prince of Songkla University, Hat Yai, Songkhla, Thailand 90112, E-mail: unitsa.s@psu.ac.th

[4] The author is with the Department of Pathology, Faculty of Medicine, Ramathibodi Hospital, Mahidol University, Bangkok, Thailand 10400, E-mail: ekawat.pas@mahidol.ac.th

MSA result for more accuracy, before entering them into bootstrapping step. It is an important step because the MSA result significantly influences the outputs of the phylogenetic tree. Nevertheless, the bioinformaticians have to manually determine and trim the MSA result. It appears to be a tedious task. However, there is a tool for automated removal of spurious sequences or poorly aligned regions before performing MSA [20]. However, our approach considers a different method.

Bioinformaticians usually consider the MSA similarity score when they select which tree inferring algorithm is suitable. The MSA similarity score can be generated by different algorithms depending on the knowledge and experiences of the bioinformaticians. Improperly algorithm selection without applying the MSA similarity score tends to generate a phylogenetic tree with low accuracy and also take much more time.

Currently, there are no programs or services which automatically provide for both similarity score estimation and gaps trimming for these automatic steps. In this work, we present an *in-silico* approach for the phylogenetic tree analysis using our enhancement for the similarity score estimation and gaps trimming of the MSA result in the interleaved Phylip format. This paper proposes to automate these two processes using conventional algorithms and further deploy them as web services. We also demonstrate our services utilized in workflows running on the Taverna Workbench, a scientific workflow management system (SWFMS), desktop version [21-22]. Our case study datasets are a set of complete CDS and sets of complete genome of Dengue Viruses - 2, fetched from the NCBI RefSeq nucleotide database [23].

In the next section, we describe our case study's data and give an overview of the MSA and tree inferring algorithms. In Section 3, we describe our new practical approach for the phylogenetic tree analysis consisting of our proposed algorithms for the similarity score estimation and gaps trimming. Then, the implemention, deployment and demonstratation of our *in-silico* methods using web services and workflow mechanisms are described in Section 4. In Section 5, we report the performance of workflow runnings and their results. Discussion and conclusion are presented in Sections 6 and 7 respectively.

2. BACKGROUNDS AND RELATED ISSUES

2.1 Taverna

Taverna is a scientific workflow management system (SWFMS) which has been introduced by myGrid project for a decade [21-22]. Taverna has initially emerged in the field bioinformatics. Nowadays, Taverna workbench has been widely deployed in a variety of research fields including biodiversity [24], chemistry, astronomy [25], data and text mining, digitization, document and image analysis, etc. Many Taverna distributions are open source and support for a variety of running environments including desktop client application, the Workbench, the Command Line Tool for a quick execution of workflows from a terminal, the Server for remote execution of workflows, the Player (a web interface plugin for submitting remote execution of workflows), and Taverna Online providing researchers to create Taverna workflows from a web browser.

A workflow is an representative of instruction steps which execute and produce required results using various types of services including WSDL Web Services, local scripts, BioMart data warehouses, RESTful Web Services, Grid Services, Cloud Services, Rscripts and distributed command-line scripts. In this paper, we demonstrate of our *in-silico* methods using web services and workflow mechanisms. Our workflows run on Taverna Workbench. We compose our workflows and saved them into files. Components of the workflows are web services which are both distributed public-access and our own local services.

There also are other SWFMSes in other significant fields such as Kepler for physics [26], Swift for climate science, Vistrails is for earth science, and VIEW for medical science [27]. In addition, the contemporary trend usually is to merge SWFMSes into Cloud platforms and enable users to access services via their portals [28-29].

2.2 Case Study Dataset

This work uses two datasets of Dengue Viruses - 2, fetched from the NCBI RefSeq nucleotide database [23]. The first dataset is GenBank Accession Numbers KF744397 - KF744408 which were found in Philippines for 12 control groups and GenBank Accession Numbers JN697058 was found in Malaysia for an out-group. Therefore, our first case study contains a 13-CDS dataset and each CDS contains about 10,000 nucleotide bases.

Another dataset is also the Dengue Viruses - 2 which were found in Thailand with different year periods from the NCBI RefSeq as shown in Table 1. There are four groups extracted from the same gene for *polyprotein* encoding. Our out-group of this dataset is also the Genbank Accession number JN697058 which were found in Malaysia. Each CDS contains about 10,000 nucleotide bases. We also utilize complete genome of this dataset for testing our proposed algorithms.

2.3 Multiple Sequence Alignment

MSA is a preprocessed data step for bootstrapping phylogenetic analysis. It supports both nucleotide and amino acid sequences. There are relevant tools and web services for performing the MSA, for example, MEGA, an offline package for many tools for phylogenetic analysis. However, the user has to manually

Table 1: *The second case study dataset of the Dengue Viruses-2 were found in Thailand and identified between 1964 - 2005*

Group No.	Time Period	No. of Seqs.	Accession Numbers	Out-group Seq
1	<1990	8	GQ868591, NC_001474, GU289914, DQ181805, DQ181806, DQ181804, DQ181803, DQ181802	JN697058
2	1990 - 1995	5	DQ181801, GQ868542, EU726767, EU687246, GQ868543	JN697058
3	1996 - 1999	8	DQ181800, GQ868545, GQ868544, FJ906958, FJ906957, U87411, DQ181798, DQ181799	JN697058
4	2000 - 2005	44	DQ181797, DQ181798, FJ810409, FJ810410, FJ810411, FJ810412, FJ744725, FJ744724, FJ744723, FJ744722, FJ744721, FJ744720, FJ744719, FJ744718, FJ744717, FJ744716, FJ744715, FJ744714, FJ744713, FJ744712, FJ744711, FJ744710, FJ687447, FJ687446, FJ687445, FJ687444, FJ687443, FJ687442, FJ687441, FJ687440, FJ687439, FJ687438, FJ687437, FJ687436, FJ687435, FJ687434, FJ687433, FJ687432, FJ687431, FJ687430, FJ687429, FJ687428, FJ898452, GU131886	JN697058

copy and paste intermediate results between tools in the package, including the prefix and suffix gaps trimming of the MSA result. There are many MSA tools publicly provided by web service providers such as the EMBL-EBI and SIB. For example, *Emma* is an MSA service provided by the SIB which wraps the ClustalW program [30]. However, it is a pretty outdated as its implementation was done quite long time ago and it cannot handle the ever gaining data such as the large CDS and complete Genome of our case study datasets.

The EMBL-EBI also is a reliable web services provider including *Clustal Omega* [16], *Kalign* [31], *MAFFT* [32] and *MUSCLE* [33]. The *Clustal Omega* is a new MSA tool that uses seeded guide trees and HMM (Hidden Markov model) profile-profile techniques to generate alignments. It is suitable for medium and large alignments. *Clustal Omega* supports a set of protein, DNA, and RNA sequences as an input dataset and also produces an output of the MSA result in the interleaved Phylip blocks format (MSA blocks), our required format for downstream processes. For example, first, we take a step of bootstrapping replicates of the MSA output, and then enter them into a tree interring prediction algorithm consequently. In addition, the EMBL-EBI provides useful information for accessing these services, especially how to compose their services into a workflow using the Taverna Workbench. In case of amino acid sequences alignment, our proposed workflow provides sequence translation using the SIB web service before entering the result into the *Clustal Omega*.

Another service provider is the Phylomon web server [15] which serves as an integrated platform for three main objectives including alignment and file format conversion, phylogenetic reconstruction, and evolutionary tests. They proposed TrimAl version 1.3 for automatically removing poorly aligned regions before applying any method to improve the alignment's quality.

This work proposes to implicitly utilize *Clustal Omega*, a stateful web service of SOAP interface for both alignment and Percent Identity Matrix (PIM) values estimation process [16] [34]. Our concerns are to estimate the MSA similarity score and improve the alignment's quality after the MSA process. Our automatic approach for the phylogenetic tree analysis will be presenting in the next section.

2.4 Tree inferring algorithms

The tree inferring is a significant process for estimating the phylogenetic tree. Its input is the MSA blocks [16] generated by the *Clustal Omega* web service. Our work focuses on the popular tree inferring algorithms bundled with the PHYLIPNEW applications [7], EMBOSS conversions [8] of the program in Joe Felsenstein's PHYLIP package which consists of PARS, DM-NJ, and ML. These inferring algorithms support both nucleotide and amino acid sequences.

3. IN-SILICO APPROACH FOR THE PHYLOGENETIC TREE ANALYSIS

In this section, we present our automatic approach for the phylogenetic tree analyzing based on the MSA similarity score and prefix and suffix gaps trimming algorithms.

Input: Coding Sequences of Dengue Viruses - 2 in the Fasta format; $codingSeqsInput$
A number of replicates for the bootstrapping algorithm; $replicatesNumber$
An out-group number of coding sequences; $outgroupNumber$

Output: A consensus phylogenetic tree; $concensusTree(treeFile, treeSpecification)$
A generated consensus tree file from an inferring algorithm; $treeFile$
A generated consensus tree specification from an inferring algorithm; $treeSpecification$

procedure PHYLOGENETICTREEANALYSIS ($codingSeqsInput, replicatesNumber, outgroupNumber$)
  if $codingSeqsInput == aminoAcidSeqs$ **then**
    $codingSeqs \leftarrow translateSeqs(codingSeqsInput)$
  else
    $codingSeqs \leftarrow codingSeqsInput$
  else if
  $msaResult \leftarrow performMsa(codingSeqs)$
  $pimMsaData \leftarrow estimatePimIndex(codingSeqs)$ ▷ performed by using the *CLUSTAL-OMEGA* web service
  $msaSimScore \leftarrow estimateMsaSimScore(pimMsaData)$ ▷ performed by using our own implemented web service
  $gapsTrimmedMSA \leftarrow gapsTrimming(msaResult)$ ▷ performed by using our own implemented web service
  $prettySeqAligned \leftarrow drawSeqAlignment(gapsTrimmedMSA)$
  $msaAlignedDisplay \leftarrow displayMSA(gapsTrimmedMSA)$
  $bootstrappedSeqs \leftarrow bootstrapingSeqs(gapsTrimmedMSA, replicatesNumber)$
  if $msaSimScore >= 60$ **then**
    $possibleOutTrees \leftarrow performPARS(bootstrappedSeqs)$ ▷ perform the PARS algorithm
  else if $msaSimScore >= 10$ AND $msaSimScore < 60$ **then**
    $distanceMatrixOut \leftarrow performDM(bootstrappedSeqs)$
    $possibleOutTrees \leftarrow performNJ(distanceMatrixOut)$ ▷ perform the DM and NJ algorithms
  else
    $possibleOutTrees \leftarrow performML(bootstrappedSeqs)$ ▷ perform the ML algorithm
  else if
  $consensusTree \leftarrow performMajorityRule(possibleOutTrees)$
return $consensusTree$
end procedure

Fig.1: *Analysis of the phylogenetic tree using public-access web services and our own local web services*

3.1 Conventional Algorithm for in-silico Similarity Score Estimation of MSA

In this work, we proposed *in-silico* steps for the MSA similarity score in order to select a tree inferring algorithm as shown in Figure 1. The score is estimated using conventional formula [35] in Equation (1) where s is the similarity score, m is the number of CDS, n is the number of each member in the CDS, and x_{ij} is the PIM value generated by the *Clustal Omega* service. The example of PIM values of the Dengue Viruses dataset is shown in Figure 2 and its similarity score is 94.40.

1: KF744397	100.00	100.00	100.00	100.00	92.17	100.00	100.00	100.00	100.00	100.00	100.00	100.00	67.66
2: KF744398	100.00	100.00	100.00	100.00	92.15	100.00	100.00	100.00	100.00	100.00	100.00	100.00	67.73
3: KF744399	100.00	100.00	100.00	100.00	92.17	100.00	100.00	100.00	100.00	100.00	100.00	100.00	67.66
4: KF744400	100.00	100.00	100.00	100.00	92.17	100.00	100.00	100.00	100.00	100.00	100.00	100.00	67.66
5: KF744401	92.17	92.15	92.17	92.17	100.00	92.17	92.17	92.17	92.17	92.17	92.17	92.17	67.94
6: KF744402	100.00	100.00	100.00	100.00	92.17	100.00	100.00	100.00	100.00	100.00	100.00	100.00	67.66
7: KF744403	100.00	100.00	100.00	100.00	92.17	100.00	100.00	100.00	100.00	100.00	100.00	100.00	67.66
8: KF744404	100.00	100.00	100.00	100.00	92.17	100.00	100.00	100.00	100.00	100.00	100.00	100.00	67.66
9: KF744405	100.00	100.00	100.00	100.00	92.17	100.00	100.00	100.00	100.00	100.00	100.00	100.00	67.66
10: KF744406	100.00	100.00	100.00	100.00	92.17	100.00	100.00	100.00	100.00	100.00	100.00	100.00	67.66
11: KF744407	100.00	100.00	100.00	100.00	92.17	100.00	100.00	100.00	100.00	100.00	100.00	100.00	67.66
12: KF744408	100.00	100.00	100.00	100.00	92.17	100.00	100.00	100.00	100.00	100.00	100.00	100.00	67.66
13: JN697058	67.66	67.73	67.66	67.66	67.94	67.66	67.66	67.66	67.66	67.66	67.66	67.66	100.00

Fig.2: *PIM values of the 13-CDS dataset of Dengue Viruses are generated by Clustal Omega service*

$$s = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}}{mn} \quad (1)$$

Page and Holmes (1998) [1] suggested the PARS algorithm if the MSA similarity score is greater than or equal to 60. The PARS algorithm is a character-based method determining maximum parsimony calculating the distance and building groups of which members are closer to each other. The DM-NJ algorithm should be selected when the score is greater than ten but less than 60. For the rest, the ML

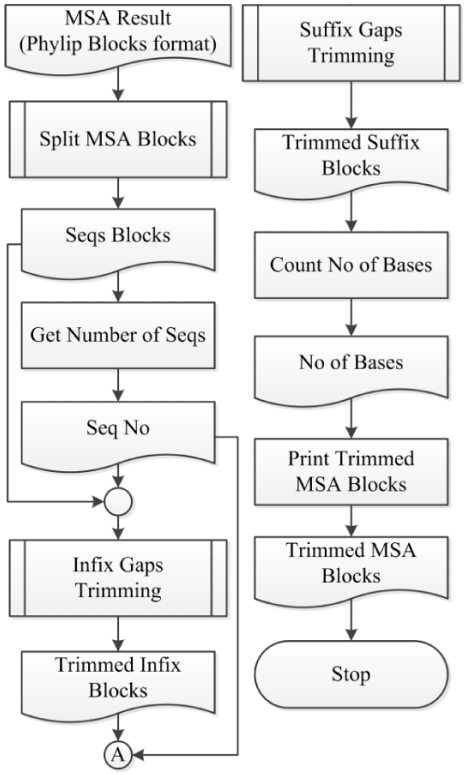


Fig.3: *Algorithm for prefix and suffix gaps trimming of multiple sequence alignment*

algorithm should be selected to perform brute-force tree interring. The ML algorithm, a character-based method which finds the most possible tree, is the most computational intensive, whereas the PARS algorithm is the fastest tree inferring analysis, and the DM-NJ takes more time but is faster than the ML algorithm. Our work concerns how to automatically select the suitable tree inferring algorithm based on the similarity score as well as to enable the user to diverse the decision when the score is close to the boundary.

3.2 Conventional Algorithms for in-silico Gaps Trimming of MSA

In practical steps [1] [9] [36], bioinformaticians usually use a suitable dataset of the phylogenetic tree analysis for avoiding GIGO (garbage in, garbage out). Therefore, the dataset has already been preprocessed and filtered into a CDS. It is suitable to perform MSA to find the similarity among sequences of the CDS. The MSA process usually produces a result containing some gaps in any supported output formats such as Clustal, Pearson/Fasta, Nexus, or Phylip of tools, even though using standalone programs, e.g. MEGA, or web services, e.g. *Clustal Omega*.

The input format of the MSA result for bootstrapping and tree inferring steps used in our work is the interleaved Phylip format. We refer to MSA blocks in this paper. The result generated by the *Clustal Omega* service of the MSA blocks usually contains some gaps resulting from the alignment process. These gaps can occur at any position depending on the similarity of the CDS dataset. If the CDS dataset is highly similar presenting a close relationship of an organism, the gaps usually occur in a few blocks at the beginning (prefix) and the end (suffix) of the MSA blocks, or even no gaps exist. However, it is also normal to have a few gaps scattering all over the MSA blocks.

In this work, we proposed conventional algorithms for *in-silico* prefix and suffix gaps trimming of MSA result as shown in Figure 3. The input is the MSA blocks of the CDS dataset generated by the *Clustal Omega* service. The output is the MSA blocks with prefix and suffix gaps trimmed ready for the next process, the bootstrapping replication. The first step of our proposed gaps trimming is blocks splitting as shown in Figure 4.

The algorithm splits the MSA blocks into individual blocks. For example, Figures 5(a) and 5(b) show the first and second blocks of the 13-CDS datasets respectively. Then, we perform the prefix gaps trimming for each block using the algorithm shown in Figure 7. The algorithm finds the maximum gap position to trim off data in all sequences scanning each vertical block running from the beginning of each sequence into the inside until all sequences appear no gap at the position.

For example, the maximum gap position is located in the first block as shown in Figure 5(a), whereas we indicates the trimming position by a vertical line. Then, the algorithm deletes all bases at the same position for all sequences. The next process is the suffix gaps trimming using our proposed algorithm shown in Figure 8. Our algorithm initially determines the MSA blocks from the last block. The algorithm finds the minimum base position to trim off all sequences applying this position for each block. For example, Figure 6(c) shows the first last-block of the 13-CDS dataset. The algorithm deletes this block because there are at least one sequence containing no base as well as the second last-block shown in Figure 6(b). Therefore, the algorithm determines that the minimum base position is located in the third last-block as shown in Figure 6(a). The trimming position is indicated with the vertical line. Then, the algorithm deletes all bases at the same position for all sequences.

4. IMPLEMENTATION AND DEMONSTRATION OF PROPOSED ALGORITHMS

In this section, we present details for implementation and deployment of the proposed *in-silico* methods, demonstrated in our workflows composed using Taverna Workbench.

Input: The MSA result in the Phylip format of the CDS dataset generated by the *CLUSTAL-OMEGA* web *service*; *msaResult*
Output: Sequence blocks of the MSA result; *seqsBlock*

```
1:  procedure SPLITBLOCK (msaResult)
2:    strSeqNo ← msaResult.split(NEWLINE)
3:    lineCnt ← 1
4:    for seqNoIndex in strSeqNo.length do
5:        if seqNoIndex == 0 then
6:          perform firstLine.split(SPACE)
7:          seqNo ← firstLine.seqNo
8:          noBase ← firstLine.noBase
9:        else if seqNoIndex == 1 then
10:           for seqIndex in seqNo do
11:               perform firstBlock.split(SPACE)
12:               extract allSeqNo from firstBlock
13:               extract firstMSABlock from firstBlock
14:           end for
15:            perform seqsBlock.add(firstMsaBlock)
16:       else
17:           otherMsaBlocks ← strSeqNo[seqNoIndex]
18:           lineCnt + +
19:           if lineCnt > seqNo then
20:             perform seqsBlock.add(otherMsaBlocks)
21:           end if
22:       end if
23:   end for
24: return seqsBlock
25: end procedure
```

Fig.4: *Algorithm for splitting blocks of the MSA in the Phylip format*

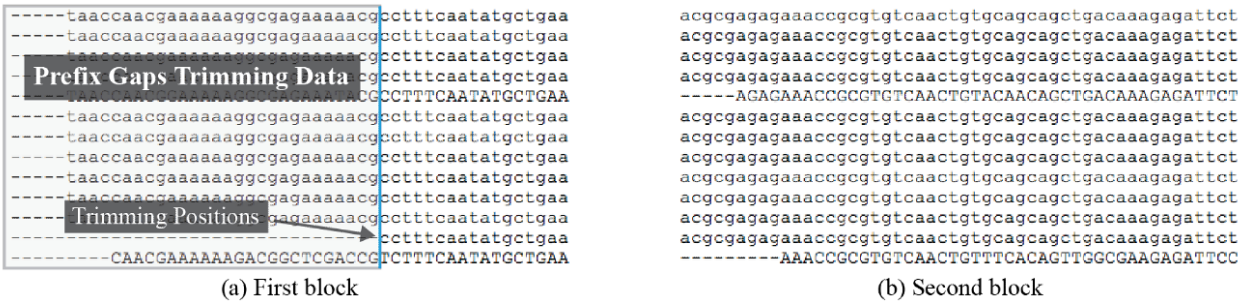


Fig.5: *Example of MSA blocks in the Phylip format shows the prefix gaps trimming position for all sequences of the 13-CDS Dengue Viruses dataset appears in the first block*

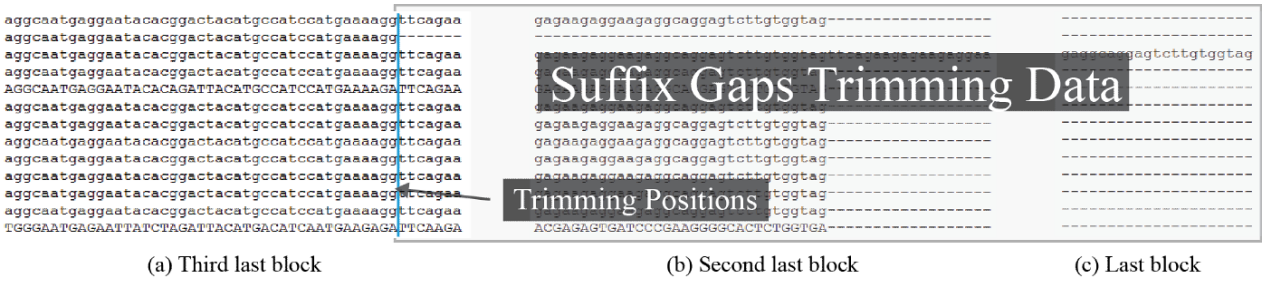


Fig.6: *Example of MSA blocks in the Phylip format shows the suffix gaps trimming position for all sequences of the 13-CDS Dengue Viruses dataset appears in the third block*

4.1 Implementation

Our all proposed algorithms have been implemented using Java and deployed into web service interfaces. The web services have been utilized by the Apache Axis2, version 1.7.0 that pro-

Input: Sequence blocks of the MSA result in the Phylip format of the CDS dataset generated by the *CLUSTAL-OMEGA* web service; $seqsBlock$
A number of sequences in the CDS dataset; $seqNo$
Output: Prefix gaps trimmed of the MSA result; $vecTrimmedPrefix$

```
1:  procedure GAPSTRIMMINGPREFIX (seqsBlock, seqNo)
2:    for blockIndex in seqsBlock.size do
3:      for seqNoIndex in seqNo do
4:        eachSeqsBlock ← seqNo[seqNoIndex]
5:        baseDetectPosition[] ← findFirstOccurredBasePosition(eachSeqsBlock, blockIndex)
6:        maxGapPosition ← findMaximumBasePosition(baseDetectPosition[])
7:        seqsBlock ← prefixTrimming(maxGapPosition)
8:      end for
9:    end for
10:   vecTrimmedPrefix ← seqsBlock
11: return vecTrimmedPrefix
12: end procedure
```

Fig.7: *Algorithm for prefix gaps trimming of the MSA Phylip blocks*

Input: Prefix Gaps trimmed sequence blocks of the MSA result in the Phylip format of the CDS dataset generated by the Algorithm in Figure 7; $vecTrimmedPrefix$
A number of sequences in the CDS dataset; $seqNo$
Output: Suffix gaps trimmed of the MSA result; $vecTrimmedSuffix$

```
1:  procedure GAPSTRIMMINGPREFIX (vecTrimmedPrefix, seqNo)
2:    seqBlock ← vecTrimmedPrefix
3:    blockIndex ← seqsBlock.size
4:    while blockIndex >= 0 do
5:      for seqNoindex in seqNo do
6:        eachSeqsBlock ← seqNo[seqNoIndex]
7:        gapDetectPosition[] ← findFirstOccurredGapPosition(eachSeqsBlock, blockIndex)
8:        minGapPosition ← findMinimumGapPosition(gapDetectPosition[])
9:        seqsBlock ← prefixTrimming(minGapPosition)
10:     end for
11:     blockIndex − −
12:   end while
13:   vecTrimmedSuffix ← seqsBlock
14: return vecTrimmedSuffix
15: end procedure
```

Fig.8: *The algorithm for suffix gaps trimming of the MSA Phylip blocks*

vides WSDL endpoints of Plain Old Java Objects (POJOs) [37] for publicly access. Tomcat, version 8.0.26 is our web services container running on the Cent OS Linux 6.0. The WSDL endpoint of the MSA similarity score estimation web service is `http://bioservices.sci.psu.ac.th/axis2/services/MSASimilarityEstimation?wsdl`.

The WSDL endpoint of the MSA gaps trimming is `http://bioservices.sci.psu.ac.th/axis2/servies/MSA-GapsTrimming?wsdl` and we also distribute it as the Java Archive (JAR) for a standalone-desktop application.

4.2 Demonstration

In this paper, we demonstrate our web services' capacities by composing them into workflows using Taverna Workbench. Then, we import both endpoints as mentioned earlier into Taverna, and then compose the workflows to utilize our web services. Figure 9 shows the workflow for estimating the MSA similarity score composed into the stateful WSDL service in Taverna. We also implement a beanshell script for identifying the suitable tree inferring algorithm selected applying the MSA similarity score as shown in Figure 1. Figure 10 shows a workflow for the MSA gaps trimming into the stateful WSDL service in Taverna.

Figure 11 shows the workflow consisting of two relevant web services, i.e. *Clustal Omega* of the EMBL-EBI for performing MSA which produces the PIM values, and *MSASimilarityEstimation*, our deployed web service, receiving the PIM values from the previous service. Figure 12 also shows the workflow

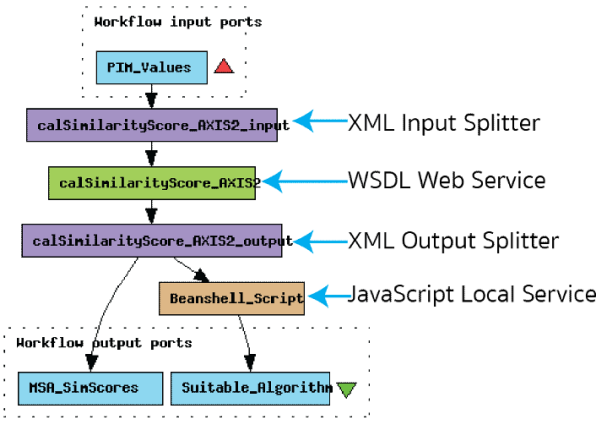


Fig.9: *Our WSDL web service of the MSA similarity score estimation is composed into a workflow using Taverna Workbench [38].*

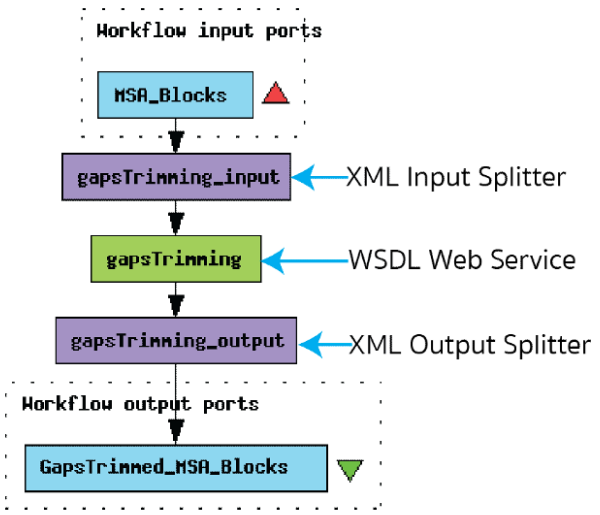


Fig.10: *Our WSDL web service of the MSA gaps trimming is composed into a workflow using Taverna Workbench [39].*

consisting of two relevant web services including the *Clustal Omega* for performing MSA which produces the MSA blocks and the *MSAGapsTrimming*, our deployed web service, receiving the MSA blocks from the other services. Finally, the workflows shown in Figure 11 and Figure 12 are integrated into the integrated workflow shown in Figure 13.

5. WORKFLOW RUNNING RESULTS

Our integrated workflow is shown in Figure 13 with the 13-CDS dataset of the Dengue Viruses in the Fasta format. The workflow's I/O ports and data links has been validated by Taverna. The workflow execution time is 12.3 minutes. The execution times for each nested workflow are shown in Table 2. The MSA for Phylip blocks and PIM

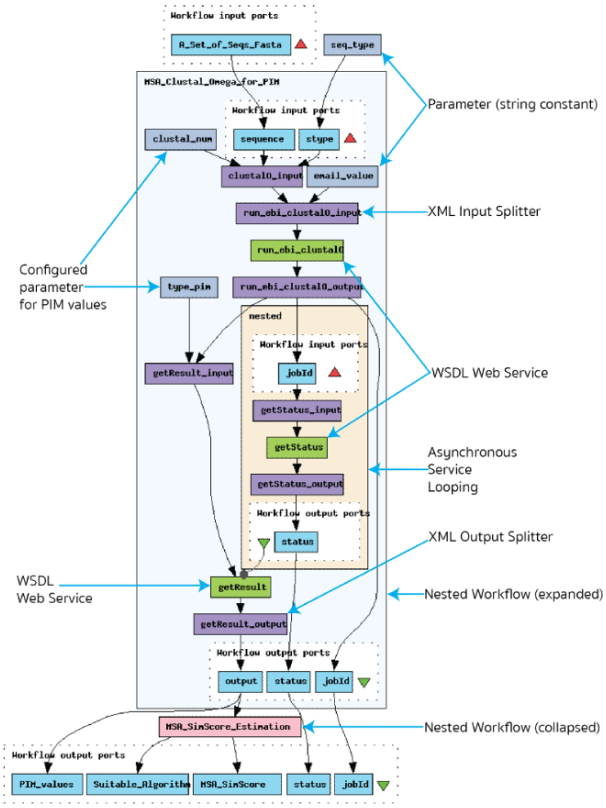


Fig.11: *Workflow for estimating the MSA similarity score composed into a stateful WSDL workflow with Clustal Omega*

data values generation on both consume 12.2 minutes. The workflow running returned the MSA similarity score of 94.40 and the PARS has been suggested as the suitable interring tree algorithm. Our enhanced web services execution times are less than one minute. We found that the MSA blocks contain no prefix gaps but they contain suffix gaps in the last three blocks as shown in Figure 14(a). The last two blocks have at least one sequence containing all gaps (Lines 2844 to 2856 and Line 2858 to 2863). Our gaps trimming web service has deleted these two blocks. Therefore, the suffix gaps position is located at Line 2831. Then, the web service has returned the correct trimmed version of the MSA blocks as shown in Figure 14(b). We have further inspected our trimmed MSA blocks using the *fseqboot*, a web service for bootstrapping replicates at the SIB (`http://wsembnet.vital-it.ch/soaplab2/`). Our trimmed MSA blocks have been validated for the *fseqboot*'s input. The service worked properly and produced correct bootstrapped results.

Another case study dataset is shown in Table 1. First, we have extracted complete CDS for each group of the dataset and appended an out-group sequence. We have run the integrated workflow applying all groups of the dataset and found that our enhancement services produced correct outputs in-

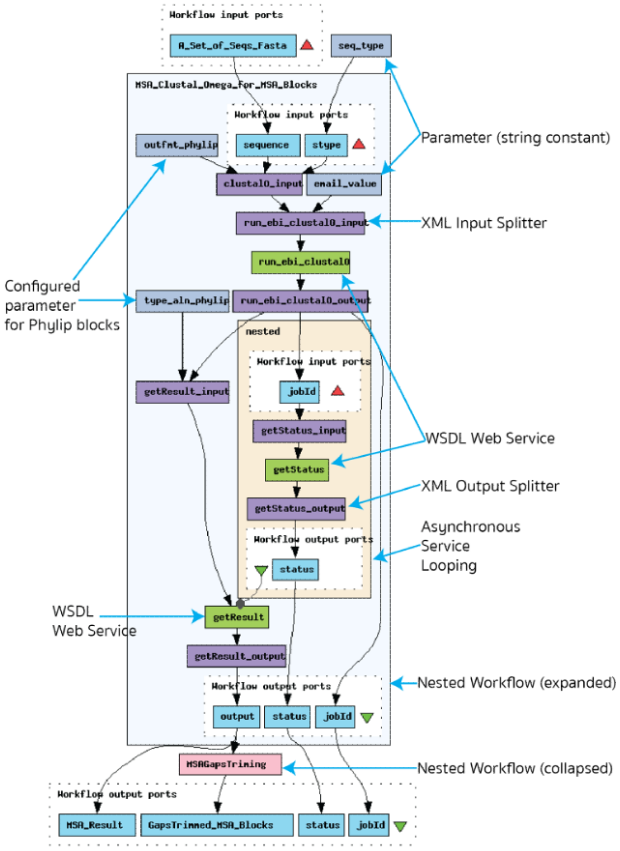


Fig.12: *Workflow for MSA gaps trimming composed into a stateful WSDL workflow Clustal Omega*

cluding when there is no prefix and suffix gaps.

Next step, we have conducted workflow running experiments with complete genome of the four groups dataset for inducing the *Clustal Omega* to generate MSA blocks containing infix and suffix gaps. Table 3 shows details of the runnings' results. The *Clustal Omega* returned infix and suffix gaps of the MSA blocks for all groups. We have found that our proposed gaps trimming web service can handle the large complete genome and produce correct trimmed results. For example, the dataset Group No. 4 took about nine seconds for operating gaps trimming and a half second for estimating the MSA similary score. The integrated workflow took about one hour for completing its job.

In this case, all dataset groups reveal the similarity score more than 60. These scores support bioinformaticians to select an inferring tree algorithm for the downstream process. For this kind of work, Page and Holmes (1998) [1] suggested the PARS algorithm for all datasets.

Figure 15(a) shows the example of MSA blocks of the complete genome for Group No.1 dataset containing many prefix gaps. The prefix gaps in the first (upper) block has been deleted and the second (lower) block has been trimmed as shown in Figure 15(b).

Table 2: *Execution times of the integrated workflow for DNA of the 13-CDS dataset of Dengue Viruses - 2*

Nested Workflows	Execution Times
CLUSTALO_PHYLIB	12.2 m
CLUSTALO_PIM	12.2 m
MSAGapsTrimming	475 ms
MSA_SimScore_Estimation	401 ms

6. DISCUSSIONS

As we have mentioned in the Sub Section 2.2, we have designed the control groups and out-group of case study datasets for demonstrating differences of the MSA results, which can be used to evaluate gaps trimming correctness. In a practical scenario, it is normal that MSA blocks contain infix gaps due to different sequences of nucleotide (DNA) and amino acid (Protein) bases. There may be infix gaps generated by an MSA tool. For example, Figure 5(b) also contains the infix gaps in the upper block. However, this paper proposes only prefix and suffix gaps trimming of the MSA blocks. In practical, bioinformaticians usually analyze a CDS dataset that has passed a quality control for the phylogenetic tree before performing the MSA. Most of infix and prefix gaps are often located a few blocks at the beginning and the end of the MSA result. There are usually not many infix gaps in these few blocks. However, it is rare to have infix gaps. In case that the MSA blocks contain many gaps overlapping between their blocks, this may influence the output tree of phylogeny. Therefore, the quality control is done by repeating the process once again.

All above processes are time consuming. In Table 3, the execution time comparison of our proposed method and other methods cannot be fairly made because of the differences in the order of applied processes and output formats. For example, one tool [20] may remove suspected sequences before performing the MSA process, meanwhile, our bioinformaticians suggest a pre-screening step to remove suspected MSA gaps of these sequences from the dataset and separately perform further analyses. By the way, we can roughly compare the execution times of our integrated workflow and the manual cut-and-paste activities between tools. For example, in case of the 4th group dataset, the integrated workflow takes about one hour whereas, the manually steps take a few days. The workflow significantly saves time. However, the infix gaps trimming remains a challenge that is worth further investigation in the future.

In case that a sequence name in the Fasta format has more than eight characters, the *Clustal Omega* returns the first block of MSA without spaces left to identify which is the sequence name or its bases. We have also found that our gaps trimming service returned an error in this case. The gaps trimming ser-

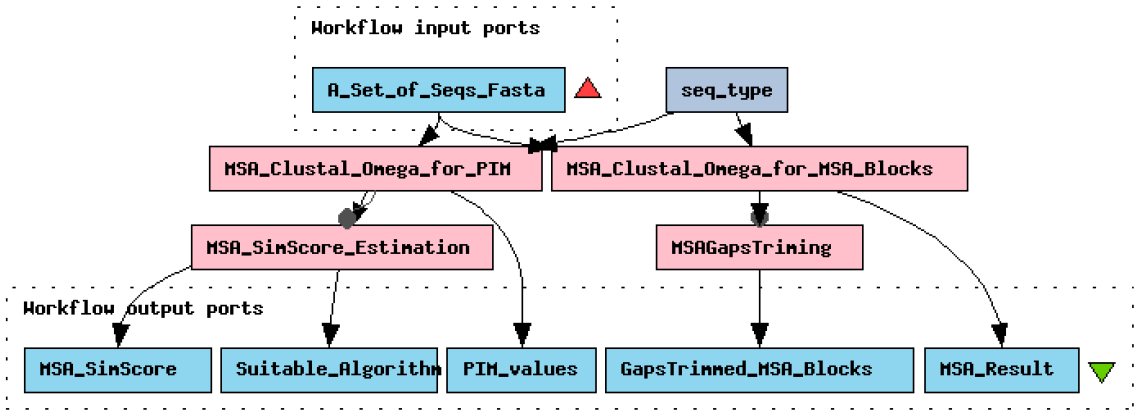
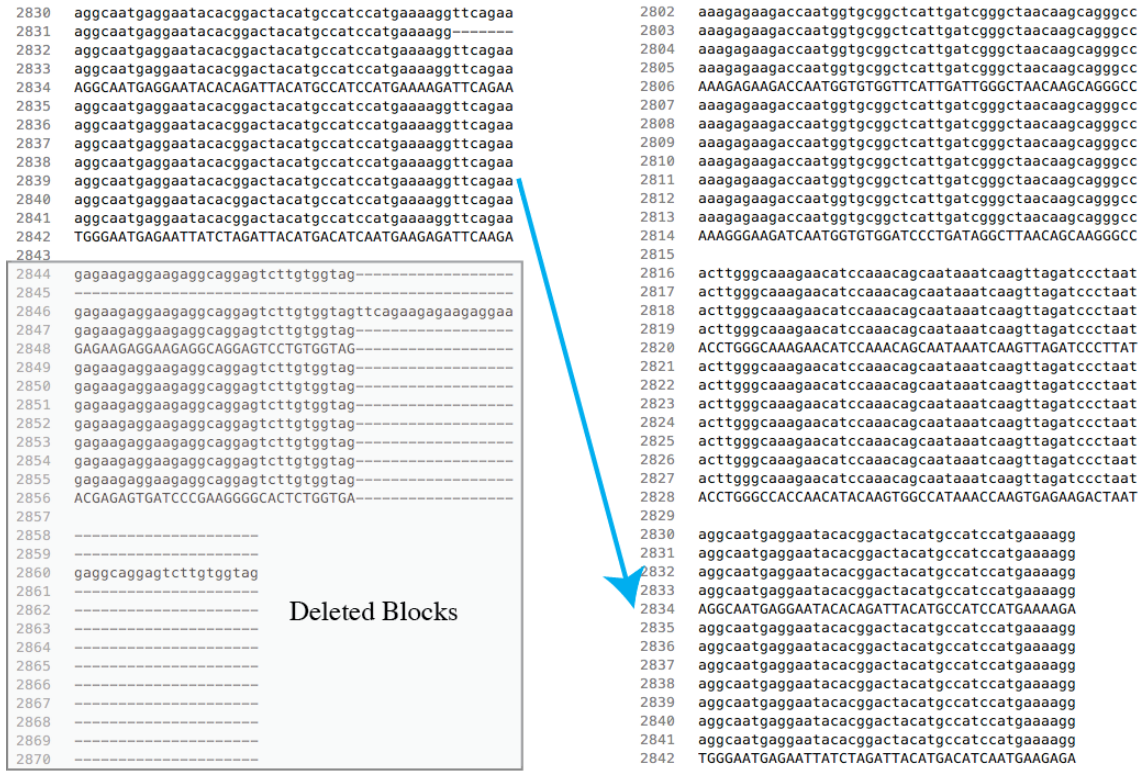


Fig.13: *Our integrated workflow for the MSA similarity score estimation and gaps trimming displayed in a collapsed display style [40]*



(a) Suffix Gaps (b) Suffix Gaps Trimmed

Fig.14: *Example of MSA blocks of the 13-CDS dataset shows suffix gaps and its trimming*

Table 3: *Execution times of the integrated workflow for DNA of the complete genome dataset of Dengue Viruses - 2 from Table 1*

Dataset Group No.	No. of seqs.	No. of Bases After Gaps Trimming	Similarity Score	Execution Times (min.)	Times Used by Manual Steps
1	8 + an out-group seq	10182	91.91	8.6	Hours
2	5 + an out-group seq	10182	88.94	6.3	Hours
3	8 + an out-group seq	10182	92.32	9.6	Hours
4	44 + an out-group seq	10806	90.69	59.0	Days

(a) Prefix Gaps (b) Prefix Gaps Trimmed

Fig.15: *Example of MSA blocks of the complete genome for Group No.1 dataset shows prefix gaps and its trimming*

vice can not detect and extract sequence names from the MSA block. Moreover, we have found that the sequence names have different lengths, for example, DQ181800 has eight characters where as the U87411 has six characters. Then, the *fseqboot*, bootstrapping replicates, produces a wrong format of outputs even though its status is reported *"successful."* This is the limitation of the bootstrapping service. Therefore, we suggest that the sequence name should not have the numbers of characters more than eight.

In addition, we are investigating another MSA tool, *MAFFT (Multiple Alignment using Fast Fourier Transform)*, a high speed multiple sequence alignment program using uses Fast Fourier Transforms. *MAFFT* is suitable for medium-large alignments. We will include *MAFFT* in the workflow as alternative options to users. However, there is a little different of output formats between *MAFFT* and *Clustal Omega* needed for improving our proposed gaps trimming algorithms to support the both MSA tools in the future.

7. CONCLUSIONS

In this work, we have compared the correctness of the results between the manual steps of cut-and-paste, which may induce errors and our *in-silico* approach.

The quality control of the MSA similarity score estimation and gaps trimming is done by the general tree re-construction. There are pre-processing processes before the bootstrapping and tree inferring steps. Phylogenies are approaches of hypotheses [35]. The trimmed blocks of the MSA do not necessarily produce the best evolution tree due to many factors concerning the complexity of the tree re-construction analysis.

The MSA similarity score estimation utilized in this work applies the conventional formula in the phylogenetic principle [1]. We have implemented the formula for automatic workflow invocation. However, for manual works, the researcher may re-constructs trees using popular tree inferring algorithms, and then reviews them, chooses the best one, and/or repeats the analysis.

We have enhanced the initial steps for the similarity score estimation and gaps trimming algorithms into automatic steps. The proposed trimming algorithms can handle prefix and suffix gaps trimming of MSA blocks. Its input is the MSA blocks of the Phylip format generated by the *Clustal Omega* web service at the EMBI-EBI and its downstream process is the bootstrapping replicates utilized by a web service at the SIB. Another algorithm is the MSA similarity score estimation. Its input is the PIM data values of the MSA result also generated by the *Clustal Omega*. We propose to utilize the similarity score for suitable selection of the tree inferring algorithms in the phylogenetic tree analysis.

These algorithms have been implemented using Java and publicly deployed as the WSDL endpoints of web services. In this paper, we have demonstrated our two web services by composing them into an integrated workflow using the Taverna Workbench. We have tested the workflow by running it applying two datasets of Dengue Viruses, including the set of complete CDS and the sets of complete genome fetched from the NCBI RefSeq. We have found that our services have correctly returned results which have been verified and satisfied by our bioinformaticians. In addition, users can download our demonstrated JAR for being used as a standalone off-line application. Our distributions, user manuals of the web services and the open source programs are provided at `http://bioservices.sci.psu.ac.th`.

ACKNOWLEDGMENT

The case study datasets were supported by the NCBI RefSeq nucleotide database. The authors are thankful for the scholarship granted by Prince of Songkla University.

References

[1] R. Page and E. Holmes, *Molecular evolution: a phylogenetic approach*. New Jersey, USA: Wiley-Blackwell, 1998.

[2] S. Guindon and O. Gascuel, "A simple, fast, and accurate algorithm to estimate large phylo- genies by maximum likelihood," *Systematic Biology*, vol. 52, no. 5, pp. 696-704, 2003.

[3] M. Wu and J. A. Eisen, "A simple, fast, and accurate method of phylogenomic inference," *Genome Biology*, vol. 9, no. 10, p. R151, 2008.

[4] M. Binet, O. Gascuel, C. Scornavacca, E. J. P. Douzery, and F. Pardi, "Fast and accurate branch lengths estimation for phylogenomic trees," *BMC Bioinformatics*, vol. 17, no. 1, p. 23, 2016. [Online]. Available: `http://dx.doi.org/10.1186/s12859-015-0821-8`

[5] J. Burleigh et al., "Genome-scale phylogenetics:inferring the plant tree of life from 18,896 gene trees," *Syst Biol.*, vol. 60, no. 2, pp. 117-125, Mar. 2011.

[6] S. Guindon, J. -F. Dufayard, V. Lefort, M. Anisimova, W. Hordijk, and O. Gascuel, "New algorithms and methods to estimate maximum- likelihood phylogenies: Assessing the performance of PhyML 3.0," *Systematic Biology*, vol. 59, no. 3, pp. 307-321, 2010.

[7] J. Felsenstein, "PHYLIP - Phylogeny inference package (version 3.2)," *Cladistics*, vol. 5, pp.164-166, 1989.

[8] P. Rice, I. Longden, and A. Bleasby, "EMBOSS:The European Molecular Biology Open Software Suite (2000)," *Trends in Genetics*, vol. 16, no. 6, pp. 276-277, 2000.

[9] K. Tamura, G. Stecher, D. Peterson, A. Filipski, and S. Kumar, "MEGA6: Molecular Evolutionary Genetics Analysis version 6.0," *Molecular Biology and Evolution*, vol. 30, no. 12, pp.2725-2729, Oct. 2013.

[10] W. Li, A. Cowley, M. Uludag, T. Gur, H. McWilliam, S. Squizzato, Y. M. Park, N. Buso, and R. Lopez, "The EMBL-EBI bioinformatics web and programmatic tools frame-work," *Nucleic Acids Research*, vol. 43, pp.W580-W584, Apr. 2015.

[11] EMBL-EBI, "The European Bioinformatics Institute, Part of the European Molecular Biology Laboratory," `https://www.ebi.ac.uk/`, 2017, [Online; accessed 25-July-2017].

[12] M. Pagni, J. Hau, and H. Stockinger, "A Multi-protocol Bioinformatics Web Service: Use SOAP, Take a REST or Go with HTML," in *Proc. IEEE International Symposium on Cluster Computing and the Grid*, Lyon, France, pp. 728-734, May 2008.

[13] L. J. Revell and S. A. Chamberlain, "Rphylip:an R interface for PHYLIP," *Methods in Ecology and Evolution*, vol. 5, pp. 976-981, 2014.

[14] A. L. Bazinet, D. J. Zwickl, and M. P. Cummings, "A Gateway for Phylogenetic Analysis Powered by Grid Computing Featuring GARLI 2.0," *Syst Biol*, vol. 63, no. 5, pp. syu031v1-syu031, Apr. 2014.

[15] R. Snchez, F. Serra, J. Trraga, I. Medina, J. Carbonell, L. Pulido, A. de Mara, S. Capella-Guterrez, J. Huerta-Cepas, T. Gabaldn, D. J., and H. Dopazo, "Phylemon 2.0: a suite of web-tools for molecular evolution, phylogenetics, phylogenomics and hypotheses testing," *Nucleic Acids Research*, vol. 10, no. 1093, pp. 1-5, Jun. 2011.

[16] F. Sievers, A. Wilm, D. Dineen, T. J. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Soding, J. D. Thompson, and D. G. Higginsa, "Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega," *Molecular Systems Biology*, vol. 7, no. 539, pp. 1-6, Oct. 2011.

[17] L. Kannan and W. Wheeler, "Maximum parsimony on phylogenetic networks," *Algorithms for Molecular Biology*, vol. 7, no. 9, pp. 1-10, May 2012.

[18] N. Saitou and M. Nei, "The neighbor-joining method: a new method for reconstructing phylogenetic trees," *Molecular Biology and Evolution*, vol. 4, no. 4, pp. 406-425, 1987.

[19] J. Felsenstein, "Evolutionary trees from dna sequences: a maximum likelihood approach," *J Mol Evol*, vol. 17, pp. 368-376, 1981.

[20] S. Capella-Gutierrez, J. Silla-Martinez, and T. Gabaldon, "trimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses," *Bioinformatics*, vol. 25, no. 15, pp.1972-1973, Aug. 2009.

[21] K. Wolstencroft, R. Haines, D. Fellows, A. Williams, D. Withers, S. Owen, S. Soiland-Reyes, I. Dunlop, A. Nenadic, P. Fisher, J. Bhagat, K. Belhajjame, F. Bacall, A. Hardisty, A. Nieva de la Hidalga, M. P. Balcazar Vargas, S. Sufi, and C. Goble, "The Taverna workflow suite: designing and executing workflows of WebServices on the desktop, web or in the cloud," *Nucleic Acids Research*, vol. 41, no. Web Server issue, pp. W557-W561, May 2013.

[22] W. Tan, K. Chard, D. Sulakhe, R. Madduri, I. Foster, S. Soiland, and C. Goble, "Scientific workflows as services in caGrid: a Taverna and gRAVI approach," in *Proc. IEEE International Conference on Web Services*, Los Angeles, CA, pp. 413-420, Sep. 2009.

[23] T. Tatusova, S. Ciufo, B. Fedorov, K. O'Neill, and I. Tolstoy, "RefSeq microbial genomes database: new representation and annotation strategy," *Nucleic Acids Research*, vol. 42, no. 1, pp. D553-D559, Jan. 2014.

[24] C. Mathew, A. Guntsch, M. Obst, S. Vicario, R. Haines, A. Williams, Y. de Jong, and C. Goble, "A semi-automated workflow for biodiversity data retrieval, cleaning, and quality control," *Biodiversity Data Journal*, vol. 2, p.e4221, Dec. 2014.

[25] J.E. Ruiz, J. Garrido, J.D. Santander-Vela, S. Sánchez-Expósito and L. Verdes-Montenegro, "AstroTavernaBuilding workflows with Virtual Observatory services," in *Astronomy and Computing*, Volumes 78, Pages 3-11, 2014, special Issue on The Virtual Observatory: I.

[26] I. Altintas, J.Wang, D. Crawl, and W. Li, "Challenges and approaches for distributed workflow-driven analysis of large-scale biological data," in *Proc. Workshop on Data analytics in the Cloud at EDBT/ICDT 2012 Conference*, Berlin, Germany, pp. 73-78, Mar. 2012.

[27] Y. Zhao, Y. Li, I. Raicu, S. Lu, W. Tian, and H. Liu, "Enabling scalable scientific workflow management in the Cloud," *Future Generation Computer Systems*, vol. 46, no. Issue C, pp. 3-16, May 2015.

[28] Y. Zhao, Y. Li, I. Raicu, S. Lu, C. Lin, Y. Zhang, W. Tian, and R. Xue, "A service framework for scientific workflow management in the Cloud," *IEEE Transactions on Services Computing*, vol. PP, no. 99, pp. 1-14, Aug. 2014.

[29] Y. Zhao, Y. Li, I. Raicu, C. Lin, W. Tian, and R. Xue, "Migrating Scientific Workow Management Systems from the Grid to the Cloud," *Cloud Computing for Data Intensive Applications*, pp. 231-256, Nov. 2014.

[30] J. Thompson, D. Higgins, and T. Gibson, "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic Acids Research*, vol. 22, no. 22, pp. 4673-4680, Nov. 1994.

[31] T. Lassmann, O. Frings, and E. L. L. Sonnhammer, "Kalign2: high-performance multiple alignment of protein and nucleotide sequences allowing external features," *Nucleic Acids Research*, vol. 37, no. 3, pp. 858-865, Feb. 2009.

[32] K. Katoh and H. Toh, "Recent developments in the MAFFT multiple sequence alignment program," *Briefings in Bioinformatics*, vol. 9, no. 4, pp. 286-298, Mar. 2008.

[33] R. C. Edgar, "MUSCLE: multiple sequence alignment with high accuracy and high throughput," *Nucleic Acids Research*, vol. 32, no. 5, pp. 1792-1797, Mar. 2004.

[34] B. P. Blackburne and S. Whelan, "Measuring the distance between multiple sequence alignments," *BIOINFORMATICS*, vol. 28, no. 4, pp. 495-502, Dec. 2012.

[35] J. Felsenstein, *PHYLIP (Phylogeny Inference Package) version 3.6*. Seattle: Distributed by the author, Department of Genome Sciences, University of Washington, 2005.

[36] S. Kumar, G. Stecher, and K. Tamura, "MEGA7: Molecular Evolutionary Genetics Analysis Version 7.0 for Bigger Datasets," *Molecular Biology and Evolution*, vol. 33, no. 7, pp. 1870-1874, Mar. 2016.

[37] S. Perera, C. Herath, J. Ekanayake, E. Chinthaka, A. Ranabahu, D. Jayasinghe, S. Weerawarana, and G. Daniels2, "Axis2, middleware for next generation web services,"in *Proc. IEEE International Conference on Web Services (ICWS'06)*, Chicago, USA, pp. 833-840, Sep. 2006.

[38] K. Damkliang, "Workflow of MSA Similarity Score Estimation," `http://www.myexperiment.org/workflows/4803.html`, 2017, [Online; accessed 25-July-2017].

[39] K. Damkliang, "Workflow of MSA Gaps Trimming,"`http://www.myexperiment.org/workflows/4804.html`, 2017, [Online; accessed 25-July-2017].

[40] K. Damkliang, "Workflow of MSA, Similarity Score Estimation, and Gaps Trimming," `http://www.myexperiment.org/workflows/4805.html`, 2017, [Online; accessed 25-July-2017].

Kasikrit Damkliang received a BS degree in Computer Science in 2005 and an MEng degree in Computer Engineering in 2009 from Prince of Songkla University (PSU), Thailand. Currently, he is a lecturer in the Information and Communication Technology Programme (ICT), Faculty of Science, PSU and also a PhD student at the Department of Computer Engineering, Faculty of Engineering, PSU. His research interests include HPC, Web Service, Cloud Computing, Workflow Technology, and Bioinformatics.

Pichaya Tandayya graduated in Electrical Engineering (Communications) from Prince of Songkla University (PSU) in Thailand in 1990. She obtained her Ph.D. in Computer Science in 2001 from the University of Manchester in the area of Distributed Interactive Simulation. Currently, she is an Assistant Professor working at the Department of Computer Engineering, PSU. Her current research works concern Parallel and Distributed Computing and Systems, and Assistive Technology.

Unisa Sangket received the B.Sc., M.Sc. (Computer Science), and Ph.D. (Molecular Biology and Bioinformatics) degrees from Prince of Songkla University, Thailand, in 2002, 2006, and 2011, respectively. She is currently a lecturer at the Department of Molecular Biotechnology and Bioinformatics, Faculty of Science, Prince of Songkla University. Her main areas of research interest are variant, genome, and transcriptome analysis using bioinformatics tools.

Ekawat Pasomsub graduated in Medical Technology in 2001 and obtained his Ph.D. in Clinical Pathology in 2010 from Mahidol University (MU), Thailand. Currently, he is a lecturer in Department of Pathology, Faculty of Medicine, Ramathibodi Hospital, MU. His current research works concern laboratory diagnosis for viruses, HIV drug resistance, genetic association study, and applications on next generation sequencing technology.