An optimal Mesh Algorithm for Remote Protein Homology Detection

Firdaus M Abdullah[1] , Razib M. Othman[1,]*, Shahreen Kasim[2], Rathiah Hashim[2],
Rohayanti Hassan[1], Hishammuddin Asmuni[1], Jumail Taliba[1]

[1]Laboratory of Computational Intelligence and Biotechnology,
Universiti Teknologi Malaysia, 81310 UTM Skudai, MALAYSIA.
surayatiismail@gmail.com, razib@utm.my, rohayanti@utm.my,
hishamudin@utm.my, jumail@utm.my
[2]Department of Web Technology,
Faculty of Computer Science and Information Technology, Universiti Tun Hussein
Onn Malaysia, 86400 Parit Raja, Batu Pahat, MALAYSIA.
shahreen@uthm.edu.my, rathiah@uthm.edu.my
**Corresponding author*

Abstract

Remote protein homology detection is a problem of detecting evolutionary relationship between proteins at low sequence similarity level. Among several problems in remote protein homology detection include the questions of determining which combination of multiple alignment and classification techniques is the best as well as the misalignment of protein sequences during the alignment process. Therefore, this paper deals with remote protein homology detection via assessing the impact of using structural information on protein multiple alignments over sequence information. This paper further presents the best combinations of multiple alignment and classification programs to be chosen. This paper also improves the quality of the multiple alignments via integration of a refinement algorithm. The framework of this paper began with datasets preparation on datasets from SCOP version 1.73, followed by multiple alignments of the protein sequences using CLUSTALW, MAFFT, ProbCons and T-Coffee for sequence-based multiple alignments and 3DCoffee, MAMMOTH-mult, MUSTANG and PROMALS3D for structural-based multiple alignments. Next, a refinement algorithm was applied on the protein sequences to reduce misalignments. Lastly, the aligned protein sequences were classified using the pHMMs generative classifier such as HMMER and SAM and also SVMs discriminative classifier such as SVM-Fold and SVM-Struct. The performances of assessed programs were evaluated using ROC, Precision and Recall tests. The result from this paper shows that the combination of refined SVM-Struct and PROMALS3D performs the best against other programs, which suggests that this combination is the best for RPHD. This paper also shows that the use of the refinement algorithm increases the performance of the multiple alignments programs by at least 4%.

Keywords: *Classification, Multiple Alignment, Remote Protein Homology, Support Vector Machines.*

1. Introduction

Remote protein homology detection (RPHD) is the inference of structural or functional information of proteins by finding a relationship or homology between new sequences and proteins of which its structural properties are already known at low levels of sequence similarity. Traditional laboratory methods to detect remote protein homology are lengthy and expensive, making it unpractical to be applied to the unprecedented amount of protein sequence data made available from the advances in molecular biology. As a result, researchers have turned to the computational methods which are more powerful and could accommodate such large amount of data.

Examples of computational methods in RPHD include multiple alignment and classification. The use of multiple alignment has been proven to improve RPHD [1]. There are two types of multiple alignments in bioinformatics; the multiple sequence alignments and the multiple structural alignments. Based on the observation that protein three-dimensional structure are surprisingly stable with respect to amino acid sequences [2], it is suggested that alignments derived from structural information should have an increased accuracy over sequence alignment alone. On the other hand, in terms of classification, studies have shown that discriminative classification algorithm such as Support Vector Machines (SVMs) outperform generative classification algorithm such as profile Hidden Markov Models (pHMMs) [3-5]. However, there is no study yet that has assessed the combination of these two types of classification algorithm with multiple sequence alignment and multiple structural alignments, further determining which combination is the best. Meanwhile, due to weaknesses in the multiple alignments algorithm, the tendency of misalignments to occur during the multiple alignment process is unacceptably high [6]. These misalignments have to be reduced as it would bring serious alignment errors that could affect the performance of the classification algorithm, thus influencing the determination on the best combination of the multiple alignments and classification algorithm.

Multiple alignments can be divided into two types; multiple sequence alignments and multiple structural alignments. A multiple sequence alignment is a sequence alignment of three or more biological sequences. Multiple sequence alignments arranges protein sequences into a rectangular array with the goal that residues in a given column are homologous, superposable or play a common functional role [7]. Examples of most recent multiple sequence alignment software includes MUMMALS [8], DIALIGN-TX [9], MAVID [10] and BAli-Phy [11]. On the other hand, a multiple structural alignment is a form of sequence alignment based on comparison of shape. Multiple structural alignments has been proven to be helpfulin protein structure classification and structure-based functionprediction by highlighting structurally conserved regions of functional significance as well as selectivity determinants [12, 13]. Examples of most current multiple structural alignment softwares include CAALIGN [14], Vorolign [15], Matt [16] and POSA [17].

In order to provide classification on the multiple alignment of protein, two types of classification algorithm are usually used in bioinformatics; the generative classification algorithm and the discriminative classification algorithm. Generative classification algorithm offers a probabilistic measure of relationship between a new sequence and a particular family. These methods such as pHMMs can be trained iteratively in a semi-supervised manner using both positively labelled and unlabelled samples of a particular family. Examples of pHMMs softwares includes HMMEditor [18], Profile Comparer [19], Meta-MEME [20] and GENEWISE [21]. In the meantime, discriminative classification algorithm focuses on learning a combination of the features that discriminate between the families. Discriminative classification algorithm are able to gain additional accuracy by modelling the difference between positive and negative examples explicitly. Thus, providing state-of-the-art performance with appropriate kernels. Examples of SVM softwares includes SVM-Gist [22], SVM Classifier [23], SVM-Prot[24] and SVM-Fold [25].

The use of heuristic approaches in multiple alignments which usually sacrifices accuracy in favour of computational efficiency has led to the problem of errors called misalignments. In progressive alignment strategy for example, errors during the early steps of the alignment process are propagated to the final output. This is certainly undesirable as multiple alignments are very important in many of bioinformatics application such as RPHD, protein structure prediction and phylogenetic analysis. Therefore, approaches to reduce the misalignments has been introduced which include the use of phylogenetic tree, consistency-based objective function and iterative realignment. Example of work using phylogenetic tree is a tool developed by Manohar and Batzoglou [26]. In their tool named TreeRefiner, sum-of-pairs function in a restricted three-dimensional space around the alignment is optimized when the tool is given a multiple alignment, a phylogenetic tree and scoring parameters as inputs.In the meantime, a consistency-based objective function incorporates multiple sequence information in scoring pairwise alignments. Example of work using consistency-based objective function is a work by Notredame et al. [27].In their work, a tool named COFFEE,evaluation is made by comparing each pair of aligned residues observed in the multiple alignments to those present in the library of a collection of all-against-all pairwise alignment on a set of given protein sequences. Meanwhile, iterative approaches usually work by performing a certain refinement function iteratively. Iterative approaches have been used either alone or in combination with other methods. For example, Edgar [28] has applied iterative refinement on MUSCLE algorithm using a variant of tree-dependant restricted partitioning. The advantage of using iteration over other approaches is that it is usually very simple whether in terms of coding the algorithm for the integration or in terms of the complexity of computational time and memory required [29].

2. Framework for Finding the Optimal Mesh Algorithm

The framework to find an optimal mesh algorithm for RPHD is consisted of three stages as shown in Figure 1. In the first stage, the multiple alignments are produced using multiple sequence alignment and multiple structural alignment programs. Four multiple sequence alignment programs namely CLUSTALW [30], MAFFT [31], ProbCons [32] and TCOFFEE [33] are chosen to be used in this paper while another four that are 3DCOFFEE [34], MAMMOTH-mult [35], MUSTANG [36] and PROMALS3D [37] are used for multiple structural alignments. In the second stage, a refinement algorithm is performed on the results of multiple alignments. The purpose is to reduce the misalignments made during the alignment process thus increasing its accuracy. The last stage is consisted of generative classification algorithm and discriminative classification algorithm. pHMMs software that are HMMER [38]and SAM[39] are used as the generative classsifier while SVMs software that are SVM-Struct [40] and SVM-Fold [41] are used as discriminative classification algorithm. Even though a similar approach has been conducted by Bernardes et al.[42] in their work by comparing the performance of various multiple alignment software, yet they only use pHMMs for classification. They also assessed only four multiple sequence alignment software and two multiple structural alignment software. This paper will take into account the use of refinement algorithm to refine multiple alignments, the use of SVMs for classification and add another two multiple structural alignment softwares for evaluation. Meanwhile, Chakrabarti et al.[43]also has applied similar refinement algorithm but they only used HMMER and SALTO_global [44] to test their work. Moreover, the dataset which they used are different from this paper where the algorithm is tested on 362 CDD [45] multiple alignments and 900 Pfam [46] alignments. They also only tested the algorithm with multiple sequence alignment using CLUSTALW and T-Coffee. This paper used datasets from the latest version of Structural Classification of Proteins (SCOP) [47] database which is version 1.73 and measured using Receiver Operating Characteristics (ROC)[48]. Precision and Recall test are also applied to support ROC because of its tendency to provide an exaggeratedly optimistic view of the classification results.

2.1. Dataset Generation

SCOP is a manually examined database of protein folds and structures that provides a comprehensive ordering according to their evolutionary and structural relationships. In this database, protein domains from all species are classified hierarchically into families, superfamilies, folds and classes. This database also includes all entries in the Protein Data Bank (PDB) [49]. Therefore, it is perfect for works on RPHD. In this paper, SCOP database version 1.73 that is the latest version of the database is used to provide the datasets. This paper only considers protein with identity below 30% to be used in this work. This paper is conducted at superfamily level because at this level families are grouped such that a common evolutionary origin is not obvious from sequence identity, but instead probable through functional features and structure analysis. Figure 2 displays the diagram and flowchart of the dataset preparation.

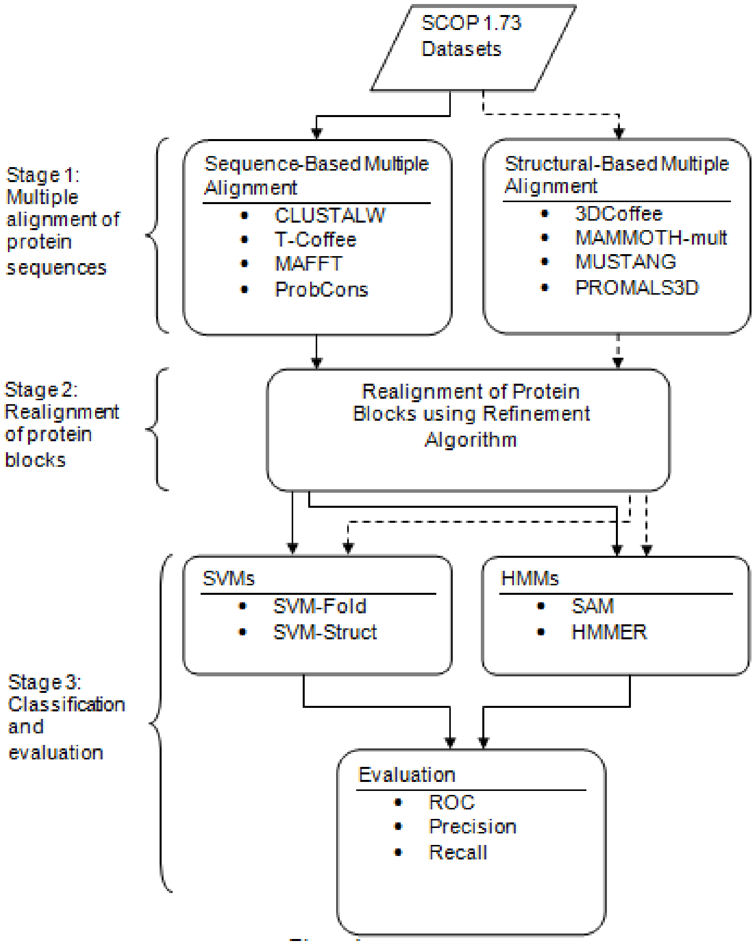
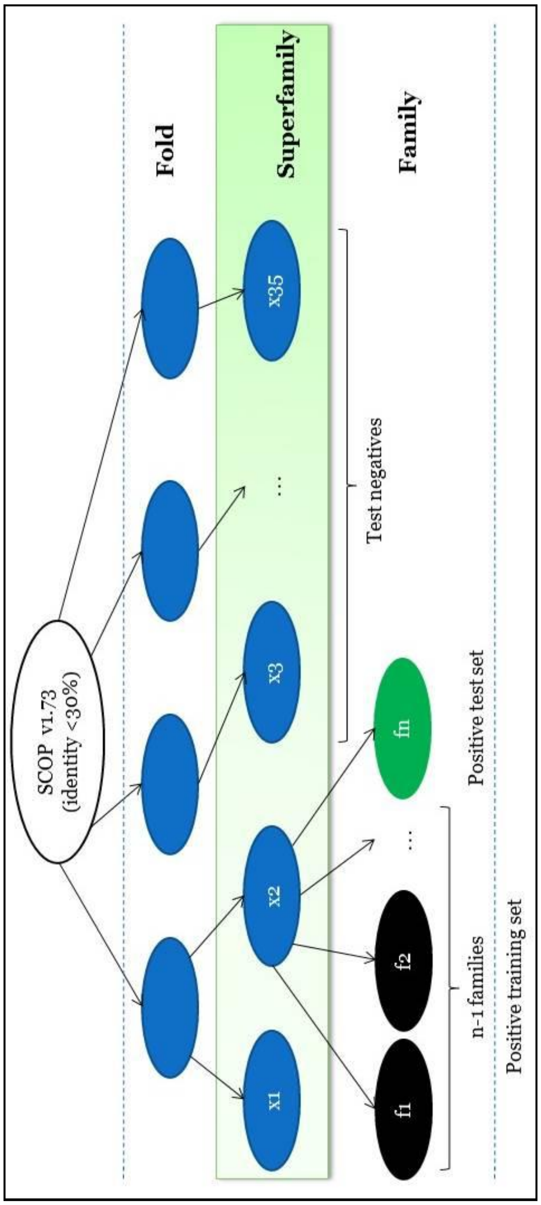
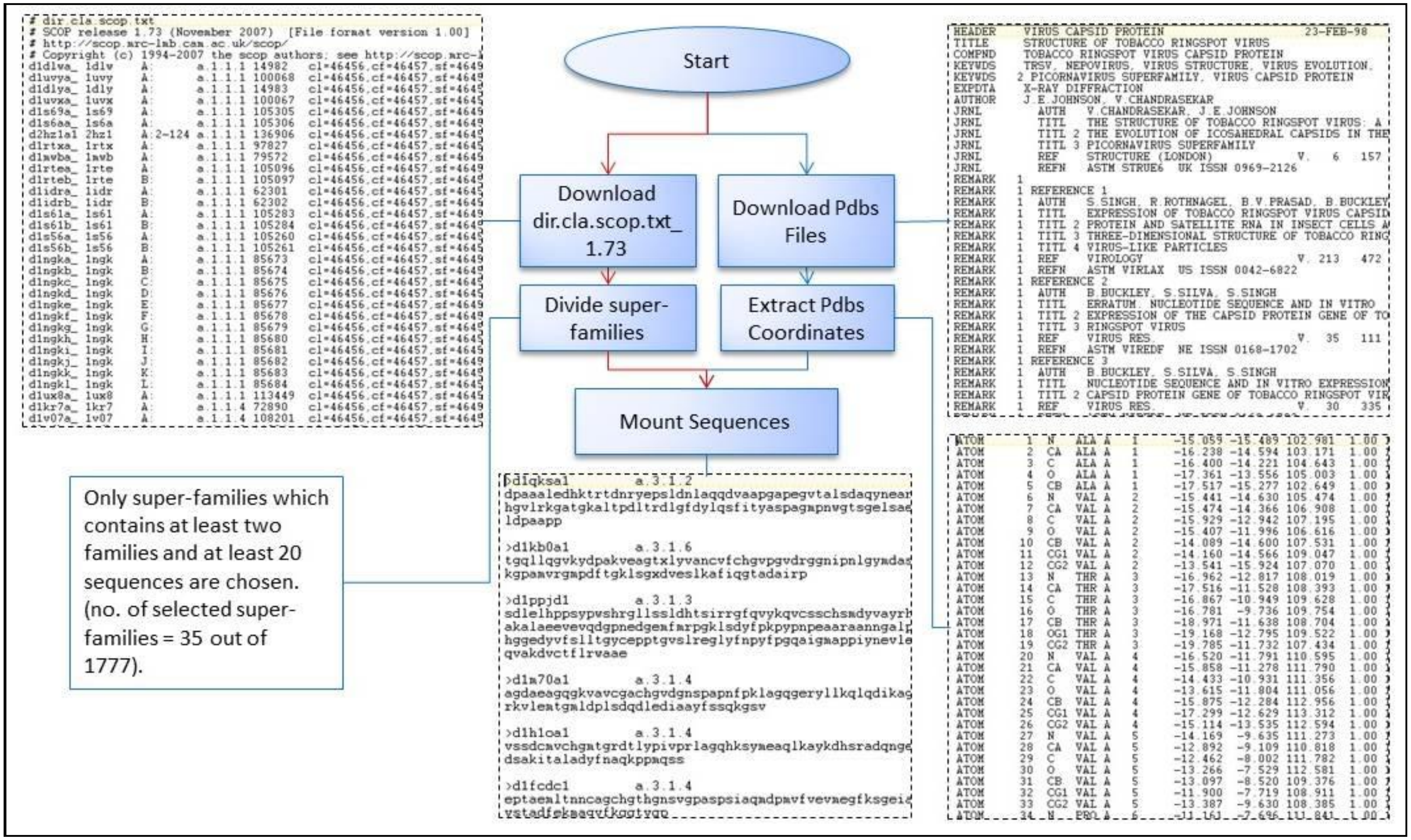


Figure 1

The datasets used in this paper are cross-validated in order to make our approach different. First, protein sequences from SCOP database are divided by super-family. Only super-families that have at least two families and 20 sequences are chosen. As a result, 35 super-families are obtained as listed in Table 1. Meanwhile, from the superfamilies we obtained 411 families and the total numbers of protein sequences are 6,496. Then, leave-one-family-out cross validation are performed where for any super-family *x* having *n* families, *n* profiles are built so that each profile *P* was built from the sequences in the remaining *n*-1 families. Lastly, the *n*-1 sequences are defined as the training set for profile *P* while the remaining sequences forms the test positives set for profile *P* and all other database sequences forms the test negatives set.



A



```
# dir.cla.scop.txt
# SCOP release 1.73 (November 2007)  [File format version 1.00]
# http://scop.mrc-lmb.cam.ac.uk/scop/
# Copyright (c) 1994-2007 the scop authors; see http://scop.mrc-l
d1dlva_ 1dlv    A:        a.1.1.1 14982   cl=46456,cf=46457,sf=4649
d1uvya_ 1uvy    A:        a.1.1.1 100068  cl=46456,cf=46457,sf=4649
d1dlya_ 1dly    A:        a.1.1.1 14983   cl=46456,cf=46457,sf=4645
d1uvxa_ 1uvx    A:        a.1.1.1 100067  cl=46456,cf=46457,sf=4645
d1s69a_ 1s69    A:        a.1.1.1 105305  cl=46456,cf=46457,sf=4649
d1s6aa_ 1s6a    A:        a.1.1.1 105306  cl=46456,cf=46457,sf=4649
d2hz1a1 2hz1    A:2-124   a.1.1.1 136906  cl=46456,cf=46457,sf=4645
d1rtxa_ 1rtx    A:        a.1.1.1 97827   cl=46456,cf=46457,sf=4645
d1mvba_ 1mvb    A:        a.1.1.1 79572   cl=46456,cf=46457,sf=4645
d1rtea_ 1rte    A:        a.1.1.1 105096  cl=46456,cf=46457,sf=4649
d1rteb_ 1rte    B:        a.1.1.1 105097  cl=46456,cf=46457,sf=4649
d1idra_ 1idr    A:        a.1.1.1 62301   cl=46456,cf=46457,sf=4645
d1idrb_ 1idr    B:        a.1.1.1 62302   cl=46456,cf=46457,sf=4645
d1s61a_ 1s61    A:        a.1.1.1 105283  cl=46456,cf=46457,sf=4649
d1s61b_ 1s61    B:        a.1.1.1 105284  cl=46456,cf=46457,sf=4649
d1s56a_ 1s56    A:        a.1.1.1 105260  cl=46456,cf=46457,sf=4645
d1s56b_ 1s56    B:        a.1.1.1 105261  cl=46456,cf=46457,sf=4645
d1ngka_ 1ngk    A:        a.1.1.1 85673   cl=46456,cf=46457,sf=4649
d1ngkb_ 1ngk    B:        a.1.1.1 85674   cl=46456,cf=46457,sf=4649
d1ngkc_ 1ngk    C:        a.1.1.1 85675   cl=46456,cf=46457,sf=4645
d1ngkd_ 1ngk    D:        a.1.1.1 85676   cl=46456,cf=46457,sf=4645
d1ngke_ 1ngk    E:        a.1.1.1 85677   cl=46456,cf=46457,sf=4649
d1ngkf_ 1ngk    F:        a.1.1.1 85678   cl=46456,cf=46457,sf=4649
d1ngkg_ 1ngk    G:        a.1.1.1 85679   cl=46456,cf=46457,sf=4649
d1ngkh_ 1ngk    H:        a.1.1.1 85680   cl=46456,cf=46457,sf=4645
d1ngki_ 1ngk    I:        a.1.1.1 85681   cl=46456,cf=46457,sf=4645
d1ngkj_ 1ngk    J:        a.1.1.1 85682   cl=46456,cf=46457,sf=4649
d1ngkk_ 1ngk    K:        a.1.1.1 85683   cl=46456,cf=46457,sf=4649
d1ngkl_ 1ngk    L:        a.1.1.1 85684   cl=46456,cf=46457,sf=4645
d1ux8a_ 1ux8    A:        a.1.1.1 113449  cl=46456,cf=46457,sf=4645
d1kr7a_ 1kr7    A:        a.1.1.4 72890   cl=46456,cf=46457,sf=4649
d1v07a_ 1v07    A:        a.1.1.4 108201  cl=46456,cf=46457,sf=4649
```

```
HEADER    VIRUS CAPSID PROTEIN                    23-FEB-98
TITLE     STRUCTURE OF TOBACCO RINGSPOT VIRUS
COMPND    TOBACCO RINGSPOT VIRUS CAPSID PROTEIN
KEYWDS    TRSV, NEPOVIRUS, VIRUS STRUCTURE, VIRUS EVOLUTION,
KEYWDS   2 PICORNAVIRUS SUPERFAMILY, VIRUS CAPSID PROTEIN
EXPDTA    X-RAY DIFFRACTION
AUTHOR    J.E.JOHNSON, V.CHANDRASEKAR
JRNL        AUTH   V.CHANDRASEKAR, J.E.JOHNSON
JRNL        TITL   THE STRUCTURE OF TOBACCO RINGSPOT VIRUS: A
JRNL        TITL 2 THE EVOLUTION OF ICOSAHEDRAL CAPSIDS IN THE
JRNL        TITL 3 PICORNAVIRUS SUPERFAMILY
JRNL        REF    STRUCTURE (LONDON)             V.   6   157
JRNL        REFN   ASTM STRUE6  UK ISSN 0969-2126
REMARK   1
REMARK   1 REFERENCE 1
REMARK   1  AUTH   S.SINGH, R.ROTHNAGEL, B.V.PRASAD, B.BUCKLEY
REMARK   1  TITL   EXPRESSION OF TOBACCO RINGSPOT VIRUS CAPSID
REMARK   1  TITL 2 PROTEIN AND SATELLITE RNA IN INSECT CELLS A
REMARK   1  TITL 3 THREE-DIMENSIONAL STRUCTURE OF TOBACCO RING
REMARK   1  TITL 4 VIRUS-LIKE PARTICLES
REMARK   1  REF    VIROLOGY                       V. 213   472
REMARK   1  REFN   ASTM VIRLAX  US ISSN 0042-6822
REMARK   1 REFERENCE 2
REMARK   1  AUTH   B.BUCKLEY, S.SILVA, S.SINGH
REMARK   1  TITL   ERRATUM. NUCLEOTIDE SEQUENCE AND IN VITRO
REMARK   1  TITL 2 EXPRESSION OF THE CAPSID PROTEIN GENE OF TO
REMARK   1  TITL 3 RINGSPOT VIRUS
REMARK   1  REF    VIRUS RES.                     V.  35   111
REMARK   1  REFN   ASTM VIREDF  NE ISSN 0168-1702
REMARK   1 REFERENCE 3
REMARK   1  AUTH   B.BUCKLEY, S.SILVA, S.SINGH
REMARK   1  TITL   NUCLEOTIDE SEQUENCE AND IN VITRO EXPRESSION
REMARK   1  TITL 2 CAPSID PROTEIN GENE OF TOBACCO RINGSPOT VIR
REMARK   1  REF    VIRUS RES.                     V.  30   335
```

```
ATOM      1  N   ALA A   1     -15.059 -15.489 102.981  1.00
ATOM      2  CA  ALA A   1     -16.238 -14.594 103.171  1.00
ATOM      3  C   ALA A   1     -16.400 -14.221 104.643  1.00
ATOM      4  O   ALA A   1     -17.361 -13.556 105.003  1.00
ATOM      5  CB  ALA A   1     -17.517 -15.277 102.649  1.00
ATOM      6  N   VAL A   2     -15.441 -14.630 105.474  1.00
ATOM      7  CA  VAL A   2     -15.474 -14.366 106.908  1.00
ATOM      8  C   VAL A   2     -15.929 -12.942 107.195  1.00
ATOM      9  O   VAL A   2     -15.407 -11.996 106.616  1.00
ATOM     10  CB  VAL A   2     -14.089 -14.600 107.531  1.00
ATOM     11  CG1 VAL A   2     -14.160 -14.566 109.047  1.00
ATOM     12  CG2 VAL A   2     -13.541 -15.924 107.070  1.00
ATOM     13  N   THR A   3     -16.962 -12.817 108.019  1.00
ATOM     14  CA  THR A   3     -17.516 -11.528 108.393  1.00
ATOM     15  C   THR A   3     -16.867 -10.949 109.628  1.00
ATOM     16  O   THR A   3     -16.781  -9.736 109.754  1.00
ATOM     17  CB  THR A   3     -18.971 -11.638 108.704  1.00
ATOM     18  OG1 THR A   3     -19.168 -12.795 109.522  1.00
ATOM     19  CG2 THR A   3     -19.785 -11.732 107.434  1.00
ATOM     20  N   VAL A   4     -16.520 -11.791 110.595  1.00
ATOM     21  CA  VAL A   4     -15.858 -11.278 111.790  1.00
ATOM     22  C   VAL A   4     -14.433 -10.931 111.356  1.00
ATOM     23  O   VAL A   4     -13.615 -11.804 111.056  1.00
ATOM     24  CB  VAL A   4     -15.875 -12.284 112.956  1.00
ATOM     25  CG1 VAL A   4     -17.299 -12.629 113.312  1.00
ATOM     26  CG2 VAL A   4     -15.114 -13.535 112.594  1.00
ATOM     27  N   VAL A   5     -14.169  -9.635 111.273  1.00
ATOM     28  CA  VAL A   5     -12.892  -9.109 110.818  1.00
ATOM     29  C   VAL A   5     -12.462  -8.002 111.782  1.00
ATOM     30  O   VAL A   5     -13.266  -7.529 112.581  1.00
ATOM     31  CB  VAL A   5     -13.097  -8.520 109.376  1.00
ATOM     32  CG1 VAL A   5     -11.900  -7.719 108.911  1.00
ATOM     33  CG2 VAL A   5     -13.387  -9.630 108.385  1.00
ATOM     34  N   PRO A   6     -11.161  -7.696 111.841  1.00
```

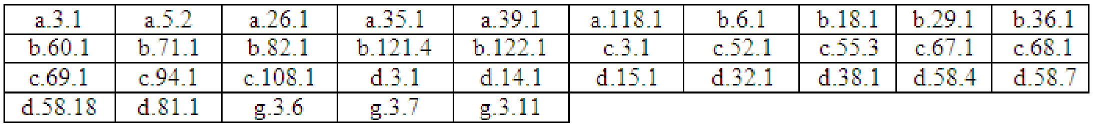
```
>d1qksa1          a.3.1.2
dpaaaledhktrtdnryepsldnlaqqdvaapgapegvtalsdaqynear
hgvlrkgatgkaltpdltrdlgfdylqsfityaspagmpnvgtsgelsae
ldpaapp

>d1kb0a1          a.3.1.6
tgqllqgvkydpakveagtxlyvancvfchgvpgvdrggnipnlgymdas
kgpamvrgmpdftgklsgxdveslkafiqgtadairp

>d1ppjd1          a.3.1.3
sdlelhppsypwshrgllssldhtsirrgfqvykqvcsschsmdyvayrh
akalaeevevqdgpnedgemfmrpgklsdyfpkpypnpeaaraanngalp
hggedyvfslltgycepptgvslreglyfnpyfpgqaigmappiynevle
qvakdvctflrwaae

>d1m70a1          a.3.1.4
agdaeagqgkvavcgachgvdgnspapnfpklagqgeryllkqlqdikag
rkvlemtgmldplsdqdlediaayfssqkgsv

>d1h1oa1          a.3.1.4
vssdcmvchgmtgrdtlypivprlagqhksymeaqlkaykdhsradqnge
dsakitaladyfnaqkppmqss

>d1fcdc1          a.3.1.4
eptaemltnncagchgthgnsvgpaspsiaqmdpmvfvevmegfksgeig
vstadfskmagvfkggtygp
```

Only super-families which contains at least two families and at least 20 sequences are chosen. (no. of selected super-families = 35 out of 1777).

B

Figure 2

Table 1. Superfamily Ids. The table list all the superfamily ids for the protein sequence selected after leave-one family-out cross validation.



a.3.1	a.5.2	a.26.1	a.35.1	a.39.1	a.118.1	b.6.1	b.18.1	b.29.1	b.36.1
b.60.1	b.71.1	b.82.1	b.121.4	b.122.1	c.3.1	c.52.1	c.55.3	c.67.1	c.68.1
c.69.1	c.94.1	c.108.1	d.3.1	d.14.1	d.15.1	d.32.1	d.38.1	d.58.4	d.58.7
d.58.18	d.81.1	g.3.6	g.3.7	g.3.11					

2.2. Multiple Alignments

Cross-validated protein sequences from the dataset generation process are input into multiple alignment programs. In this paper, four multiple sequence alignment programs and four multiple structural alignment programs are used to provide the multiple alignments. The multiple sequence alignment programs that were used are as follows:

- CLUSTALW (http://www.clustal.org/) is a progressive alignment algorithm that improvise the sensitivity of original progressive multiple sequence alignment methods via sequence weighting, position-specific gap penalties and weight matrix choices. Initially, it obtains a guide tree and then uses a greedy search to calculate the best match for the selected sequences over aligned clusters of sequences and lines them up so that the identities, similarities and differences can be seen.
- T-Coffee (http://www.tcoffee.org) also applied a progressive alignment algorithm. It works by improving the quality of the initial pairwise sequence alignment via taking into account the alignment between all the pairs as it carries out every step in the progressive alignment algorithm. By default, T-Coffee compares all sequences two by two, producing a global alignment and a series of local alignments using *lalign*. It then combines all these alignments into a multiple alignment.
- MAFFT (http://align.bmr.kyushu-u.ac.jp/mafft) package is consisted of five alignment methods that are FFT-NS-1, FFT-NS-2, NW-NS-1, NW-NS-2 and FFT-NS-i. First, it applies Fast Fourier Transform (FFT) to rapidly identify homologous regions and then it uses a simplified scoring system for reducing Central Processing Unit (CPU) time and to increase the accuracy of alignments. In its latest version, improvement have been made and new iterative refinement options, H-INS-I, F-INS-i and G-INS-I are offered [45].

- ProbCons (http://probcons.stanford.edu) merges the use of probabilistic modeling and consistency-based alignment techniques. It introduces probabilistic consistency, a novel scoring function that is based on paired HMMs using progressive multiple sequence alignment. Its emission probabilities which correspond to traditional substitution scores are based on the BLOSUM62 matrix [46]. Transition probabilities which correspond to gap penalties are trained with unsupervised expectation maximization.

Meanwhile, in order to provide multiple structural alignments, four programs that were used in this paper are as follows:

- 3DCoffee (http://www.tcoffee.org) is an aligner based on T-Coffee, but uses pairwise structure comparison to improve accuracy. Pairwise structure comparison is performed by SAP [47] if both structures are known. If only one structure is known, 3DCoffee uses the FUGUE threading method [50].

- MAMMOTH-mult (http://ub.cbm.uam.es) is a progressive multiple alignments program that uses a sequence independent heuristic to obtain a fully structural alignment. It starts from a *Cα* trace to obtain an alignment. Second, it finds an alignment of local structures based on computing a similarity score from the URMS metrics [51]. Third, it finds similar local structures with their *C*α close in Cartesian space.
- MUSTANG (http://www.cs.mu.oz.au/~arun/mustang/) is a reliable and robust algorithm for the alignment of multiple protein structures. Given a set of protein structures, the program constructs a multiple alignment using the spatial information of the *C*α atoms in the set. This algorithm gains accuracy through novel and effective refinement stages which is broadly based on the progressive pairwise heuristic. The output of MUSTANG consists of the multiple sequence alignment and the corresponding structural superpositions.
- PROMALS3D (http://prodata.swmed.edu/promals3d/) is an extension of PROMALS [50], a progressive method that clusters similar sequences and aligns them by a simple and fast algorithm which is based on the use of 3D structural information to guide sequence alignments. This method appliesmore highly structured and elaborate techniques to align the relatively differing clusters to each other. As an extension, firstly PROMALS3D identifies homologs with known 3D structures for the input sequences automatically. Second, it derives structural constraints through structure-based alignments and then combines them with sequence constraints to construct consistency-based multiple sequence alignments. The output is a consensus alignment that brings together sequence and structural information about input proteins and their homologs. The advantage of PROMALS3D is that it gives researchers an easy way to produce high-quality alignments consistent with both sequences and structures of proteins.

2.3. Refinement Algorithm

The results from the multiple alignment process are then input into a refinement algorithm. This refinement algorithm takes into account the block model of the protein family: a representation of conserved sequence or structure regions that are almost impossible to contain gaps. The conserved regions are also common to all the family members. These block models comprises a prearranged set of one or more non-overlapping blocks: regions where every sequences are aligned without gaps.

This algorithm iteratively selects random sequences from the multiple alignments and realigns it with the family block model. The iteration continues until the multiple alignment score comes to a stable value or until the iteration cycle terminates. The order in which the sequences are refined is randomized in order to avoid bias and make the use of multiple iterations more effective. In this paper, multiple alignments from the multiple alignment softwares are first input into ReadSeq (http://iubio.bio.indiana.edu/soft/molbio/readseq/java/): a software to convert the multiple alignments format from Clustal to Fasta. Then, the multiple alignments are converted to .cn3 files using fa2cd (ftp://ftp.ncbi.nih.gov/pub/REFINER/). The multiple alignments have to be converted in such a way because the refinement algorithm only understand inputs in the form of CD format, that is the same format used by CDD[39] database. One or more iterations of refinement which contains a stage of ‘block shifting’ followed by a ‘block editing’ stage are performed in the algorithm.

In the block shifting stage of the refinement, a dynamic programming(DP)[52] module that works as an engine runs on each sequence of the original multiple alignment to set new block positions. The DP module finds the optimal placement of every block in the block model on the protein sequence: rows in the multiple alignments, by scoring each allowed arrangement of the blocks on the sequence using the Position Specific Scoring Matrices(PSSM)[53]of the multiple alignments. The DP engine uses row scores: a function to determine the refined block positions for the selected sequence. The function is denoted as follows:

$$S_r = \sum_{i=1}^{L} PSSM\left(i, AA_i^r\right), \tag{1}$$

where for an alignment of length L, S_r is the sum of scores derived from the PSSM over all aligned positions of the sequence r. Therefore, The PSSM is indexed by alignment column i and the corresponding residuesAA from the sequence r. Meanwhile, the alignment score is calculated using an objective function which is denoted as follows:

$$S_N = \sum_{r=1}^{N} S_r, \tag{2}$$

where for a multiple alignment with N rows, S_N are the sum of all the row scores S_r. If the block shifting stage of the refinement has an effect on a sequence, the position of some of the blocks of the alignment on that sequence are changed or updated. The updates are represented with a function B which is denoted as follows:

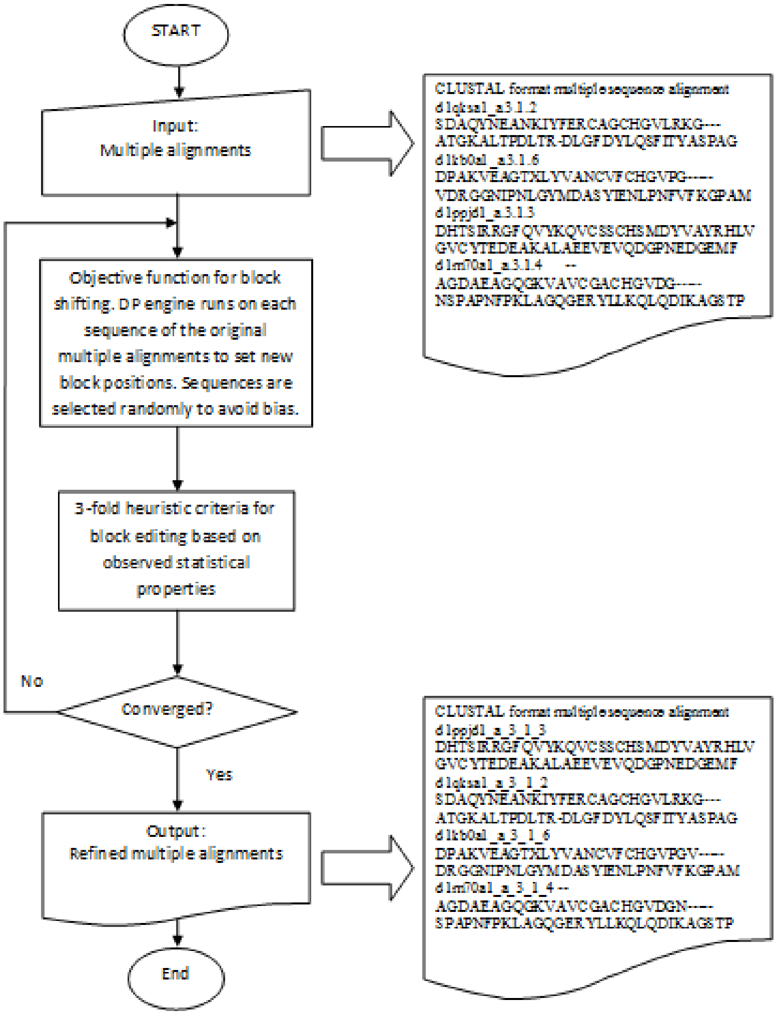
$$B = \frac{\Delta_{shift}}{l_b}, \tag{3}$$

where for a given sequence, Δ_{shift} is the difference in the position of block before and after the refinement and l_b is the length of the block$b$. The shifted blocks retain the same relative order after DP. This block shifting stage will iterates until all sequences within the multiple alignments are used.

Next, after all sequences in the multiple alignments have been used, the subsequent block editing stage examines each block across all the sequences to see if it is advantageous to extend it. A 3-fold heuristic criteria is used to control block editing based on the statistical properties observed for block and loop regions.After each of the block shifting stage, columns in loop regions adjacent to blocks are inspected and added to existing blocks until a column fails the 3-fold heuristic criteria. The 3-fold heuristic criteria are described as follows:

- First, the residues aligned in block-forming columns should have a median PSSM matrix value of at least 3 for a column to be added. In their work, Chakrabarti et al. [43]concluded that this criterion have to be met before a column can be added based on the pattern of the median PSSM value in block-forming columns in their test datasets.
- Second, the frequency of occurrence of negative scores should not exceed 0.3. The frequency of negative scores is simply computed as the ratio of the number of sequences with a negative PSSM value to the number of alignment rows.
- Third, the relative weight of negative PSSM values also should not exceed 0.3. The relative weight of negative PSSM values are computed as the absolute sum of negative PSSM values in a column divided by the sum of all PSSM values in that column.

The second and third threshold value is the characteristic of alignment column where block extension is favorable as suggested by Chakrabarti et al. [43].Convergence is declared when no further improvement of overall alignment score is observed or all iterations have expired. Lastly, the outputs of the refinement algorithm are converted back to Fasta formats followed by another conversion to Clustal formats using the ReadSeq software. This conversion is done in order to suit the requirement of the classification algorithm which only understands inputs in Clustal formats. The flow of the refinement process is represented in Figure 3.The refinement algorithm applied in this paper is not novel; it has been introduced by Chakrabarti et al.[43] and named as REFINER. However, this algorithm has not yet been tested on multiple alignments programs other than CLUSTALW. This algorithm also has not yet been tested on datasets from SCOP 1.73. Therefore, the contribution of this paper is that of the application of this refinement algorithm on the multiple alignments from different programs towards reducing the misalignment problems on datasets from SCOP 1.73.


START
Input:
Multiple alignments
Objective function for block shifting. DP engine runs on each sequence of the original multiple alignments to set new block positions. Sequences are selected randomly to avoid bias.
3-fold heuristic criteria for block editing based on observed statistical properties
No
Converged?
Yes
Output:
Refined multiple alignments
End

```
CLUSTAL format multiple sequence alignment
d1qksa1_a3.1.2
SDAQYNEANKIYFERCAGCHGVLRKG---
ATGKALTPDLTR-DLGFDYLQSFITYASPAG
d1kb0a1_a3.1.6
DPAKVEAGTXLYVANCVFCHGVPG-----
VDRGGNIPNLGYMDASYIENLPNFVFKGPAM
d1ppjd1_a3.1.3
DHTSIRRGFQVYKQVCSSCHSMDYVAYRHLV
GVCYTEDEAKALAEEVEVQDGPNEDGEMF
d1m70a1_a3.1.4      --
AGDAEAGQGKVAVCGACHGVDG-----
NSPAPNFPKLAGQGERYLLKQLQDIKAGSTP
```

```
CLUSTAL format multiple sequence alignment
d1ppjd1_a_3_1_3
DHTSIRRGFQVYKQVCSSCHSMDYVAYRHLV
GVCYTEDEAKALAEEVEVQDGPNEDGEMF
d1qksa1_a_3_1_2
SDAQYNEANKIYFERCAGCHGVLRKG---
ATGKALTPDLTR-DLGFDYLQSFITYASPAG
d1kb0a1_a_3_1_6
DPAKVEAGTXLYVANCVFCHGVPGV-----
DRGGNIPNLGYMDASYIENLPNFVFKGPAM
d1m70a1_a_3_1_4 --
AGDAEAGQGKVAVCGACHGVDGN-----
SPAPNFPKLAGQGERYLLKQLQDIKAGSTP
```

Figure 3

2.4. Classification Algorithm

The result from each of the multiple alignment programs are classified using pHMMs and SVMs. The pHMMs programs that were used are as follows:

- HMMER (http://hmmer.janelia.org/) works by differentiating between match alignment columns and insert alignment columns in model building. Then, given the model HMMER allocates columns to the match or insert states. This is done in order to increase the posterior probability of the aligned sequences to the maximum. A Dirichlet mixture with 9 stages is used for priors and Viterbi algorithm is used to do the scoring. HMMER version 2.3.3 was used in this works where three stages in the software are used that are model building, model calibration and model scoring. In model building, *hmmbuild* process is carried out to build HMMER models. Next, models are calibrated using *hmmcalibrate*. Then, *hmmsearch* process is executed for scoring. Throughout the software execution, HMMER default parameters are applied.
- The main difference between SAM (http://compbio.soe.ucsc.edu/sam2src/) and HMMER is that SAM uses a script called SAM-T2K script which performs iterative process to generate multiple alignments and HMMs. Furthermore, the developer of this software intends to improve it by using information on protein structure and prior probabilities. SAM also applies standard pHMMs architecture with 9 transitions yet however it does not dfferentiate between match and insert columns as HMMER does. This software uses a Dirichlet mixture with 20 stages for priors and Forward algorithm is used for scoring. This paper applies SAM version 3.4 where two stages in the software are used. In the first stage, *modelfromalign* is used to build the models. This is followed by the second stage where *hmmscore* is used for score computation. SAM default parameters are applied throughout the program execution.

Meanwhile, in order to provide us with discriminative classification the results from the multiple alignment process are input into SVMs. The SVMs softwares that were used are as follows:

- SVM-Struct (http://svmlight.joachims.org/) performs supervised learning by approximating a mapping of $h: X \rightarrow Y$ using labeled training examples $(x_1, y_1), ..., (x_n, y_n)$. Different with regular SVMs, SVM-Struct can predict complex objects y such as trees, sequences or sets. Examples of problems with complex outputs are natural language parsing, sequence alignment in protein homology detection and Markov models for part-of-speech tagging. SVM-Struct can also be used for linear-time training of binary and multi-class SVMs under the linear kernel. In this paper, a program called pafig (http://www.mobioinfor.cn/pafig/) is used to convert multiple aligments to feature vectors. Then, the feature vectors are input into two programs that are *svm_learn* and *svm_classify*. The latest version of SVM-Struct is version 3.1.
- SVM-Fold (http://svm-fold.c2b2.columbia.edu/) combines SVMs kernel methods with a novel multi-class algorithm, delivering efficient and accurate protein fold and superfamily recognition. It detects subtle protein sequence similarities by learning from all available annotated proteins, as well as utilizing potential hits as identified by PSI-BLAST [53]. SVM-Fold uses an efficient implementation of a state-of-the-art string kernel for sequence profiles, called the profile kernel where the underlying feature representation is a histogram of inexact matching *k-mer* frequencies. SVM-Fold also employs a novel machine learning that is the one-vs-all approach to solve the difficult multi-class problem of classifying a sequence of amino acids into one of many known protein structural classes.

2.5. Performance Evaluation

In this paper, we compare the ROC value of each combination of multiple alignment and classifier program to evaluate its performance. ROC score is the normalized area under a curve that plots true positives against false positives for different possible thresholds for classification [54]. The ROC contingency table as shown in Table 2 is referred in order to analyze evaluation measures in family classification.

Entries in the contingency table with *n* number of sequences are described as follows:

TP = number of examples correctly classified as positives
FN = number of positive examples incorrectly classified as negative
FP = number of negative examples incorrectly classified as positive
TN = number of negative examples correctly classified as negative
n = *TP* + *FN* + *FP* + *TN* (total number of sequences)

In ROC space, False Positive Rate (FPR) are plotted on the *x*-axis and True Positive Rate (TPR) are plotted on the *y*-axis. Both FPR and TPR are calculated using the following formulas:

$$\mathrm{FPR} = \frac{FP}{FP + TN}, \tag{4}$$

where FPR is the fraction of negative examples that are misclassified as negatives and

$$\mathrm{TPR} = \frac{TP}{TP + FN}, \tag{5}$$

where TPR is the fraction of positive examples that are correctly classified as positive. Meanwhile, in Precision Recall (PR) space Recall are plotted on the *x*-axis and Precision are plotted on the *y*-axis. The formula for PR is defined as follows:

$$\mathrm{Recall} = \frac{TP}{TP + FN}, \tag{6}$$

where Recall is the fraction of positive examples that are correctly classified as positive and

$$\mathrm{Precision} = \frac{TP}{TP + FP}, \tag{7}$$

where Precision is the fraction of examples classified as positives (i.e: TP and FP) that are correctly positive.

Information encoded in the contingency table can be used not only in the evaluation measurements of RPHD, but can also be applied in general classification problems. In this paper, the best combinations of a multiple alignment and a classification program are denoted by looking at the highest ROC scores.

Table 2. ROC Contingency Table. The table shows the ROC contingency matrix.

	Actual Positive	**Actual Negative**
Predicted Positive	True Positives *(TP)*	False Negatives *(FN)*
Predicted Negative	False Positives *(FP)*	True Negatives *(TN)*

3. Results

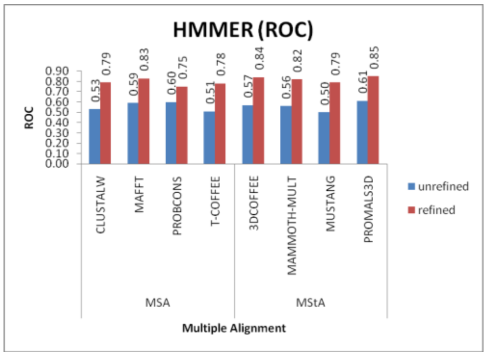
3.1. HMMER Performance

HMMER performance was assessed using multiple alignments from CLUSTALW, MAFFT, ProbCons and T-Coffee for MSA and 3DCoffee, MAMMOTH-mult, MUSTANG and PROMALS3D for MStA. Figure 4(a) shows the ROC results of unrefined and refined HMMER. Figure 4(b) and Figure 4(c) shows the Precision and Recall result of unrefined and refined HMMER respectively. For MSA, refined HMMER-MAFFT performs the best in terms of ROC with value of 0.83. Meanwhile, unrefined HMMER-CLUSTALW performs the worst with an ROC of 0.53. The result for Precision test shows that refined HMMER-ProbCons performs the best with value of 0.02893. On the other hand, unrefined HMMER-CLUSTALW performs the worst with a value of 0.00694. For Recall test, refined HMMER-T-Coffee performs the best with a value of 0.03124. Unrefined HMMER-CLUSTALW instead performs the worst with a value of 0.00756.

For MStA, refined HMMER-PROMALS3D performs the best in terms of ROC with the value of 0.85. Meanwhile, unrefined HMMER-MUSTANG performs the worst with an ROC of only 0.5. Precision test results further shows that refined HMMER-PROMALS3D performs the best with the value of 0.03377. On the other hand, unrefined HMMER-MAMMOTH-mult performs the worst with only 0.01206. As for Recall test, refined HMMER-PROMALS3D performs the best with a value of 0.03957. Meanwhile, unrefined HMMER-MUSTANG performs the worst with a value of only 0.01394.

Overall experiment result for HMMER derived from MSA and MStA shows that refined HMMER-PROMALS3D performs the best in terms of ROC with the value of 0.85. Meanwhile, unrefined HMMER-MUSTANG performs the worst with value of only 0.5. Precision test further shows that refined HMMER-PROMALS3D performs the best compared to other multiple alignment tools with a value of 0.03377. Meanwhile, the combination of refined HMMER-CLUSTALW performs worst with Precision value of 0.00694. Recall test results also shows that refined HMMER-PROMALS3D performs the best with a value of 0.03957. The combination of refined HMMER-CLUSTALW on the other hand performs the worst with a value of 0.00756.

A



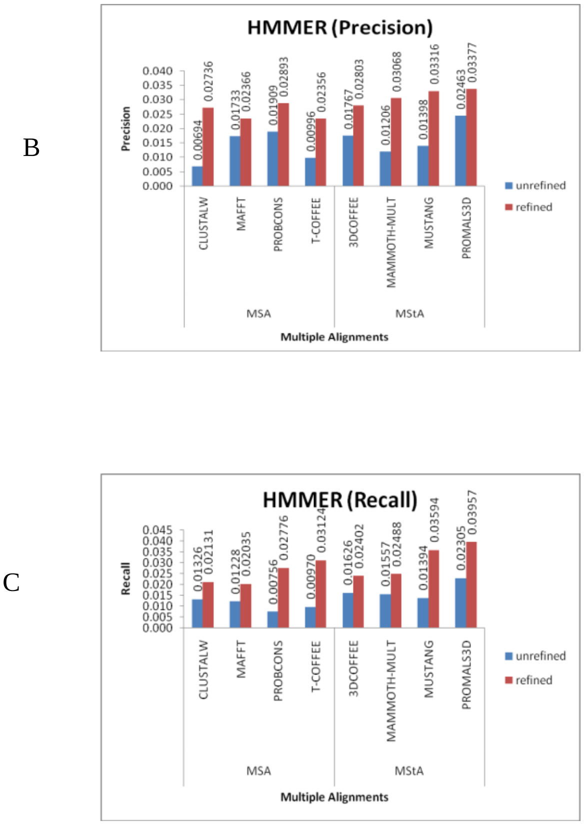


Figure 4

3.2. SAM Performance

SAM performance was also assessed using multiple alignments from CLUSTALW, MAFFT, ProbCons and T-Coffee for MSA and 3DCOFFEE, MAMMOTH-mult, MUSTANG and PROMALS3D for MStA. Figure 5(a) shows the ROC result of unrefined and refined multiple alignments combined with SAM. Figure 5(b) and Figure 5(c) both shows the result of Precision and Recall test respectively. ROC results for MSA shows that, refined SAM-MAFFT performs the best with ROC of 0.86. On the other hand, unrefined SAM-MAFFT, unrefined SAM-T-Coffee and refined SAM-ProbCons perform worst with an ROC of only 0.6. Results from Precision test in the meantime shows that refined SAM-ProbCons performs the best with a value of 0.03094. Meanwhile, the worst performance for this test is shown by the combination of unrefined SAM-T-Coffee with a value of only 0.009. As for Recall test, refined SAM-ProbCons displays the best performance with a value of 0.02797. The combination of refined SAM-T-Coffee in the meantime performs worst with a value of only 0.00973.

For MStA, the combination of refined SAM-PROMALS3D performs the best in terms of ROC with a value of 0.87. On the other hand, unrefined SAM-3DCoffee and SAM-MAMMOTH-mult performs worst with an ROC of only 0.59. Precision test also shows that the combination of refined SAM-PROMALS3D performs best with a value of 0.03893. Meanwhile, the worst combination for this test is shown by the combination of unrefined SAM-3DCoffee with a value of only 0.019. The Recall test further shows that refined SAM-

PROMALS3D performs the best with a value of 0.04147. Meanwhile, the combination of unrefined SAM-3DCoffee performs worst with a value of only 0.02.

The overall result of the comparison between unrefined and refined SAM combined with multiple alignment tools shows that refined SAM-PROMALS3D performs best with an ROC of 0.87. Meanwhile, the worst performance is shown by the combination of unrefined SAM-3DCoffee and SAM-MAMMOTH-mult with an ROC of 0.59 respectively. In terms of Precision test, the combination of refined SAM-PROMALS3D performs the best with a value of 0.03893. Meanwhile, unrefined SAM-T-Coffee performs the worst with a value of only 0.009. Recall test also shows that the combination of refined SAM-PROMALS3D performs the best with a value of 0.04147. Meanwhile, the worst performance is shown by the combination of refined SAM-T-COffee with a value of only 0.00973.

3.3. HMMER and SAM Performance

The comparison between HMMER and SAMperformance derived from the refined multiple alignments is also compared. Figure 6(a) shows the comparison of ROC between HMMER and SAM. Figure 6(b) and Figure 6(c) both shows the results comparison of Precision and Recall test respectively. For MSA, the combination of refined SAM-MAFFT performs the best in terms of ROC with a value of 0.86. Meanwhile, the worst combination is shown by refined SAM-ProbCons with an ROC of only 0.6. In terms of Precision test, the combination of refined SAM-ProbCons performs the best with a value of 0.03094. However, the combination of refined SAM-T-Coffee performs the worst with a value of 0.02061. Meanwhile, Recall test shows that the combination of refined HMMER-T-Coffee performs the best with a value of 0.03124. On the other hand, the combination of refined SAM-T-Coffee performs worst with a value of only 0.00973.

For MStA, the combination of refined SAM-PROMALS3D performs the best in terms of ROC with a value of 0.87. The worst combination is shown by the combination of refined HMMER-MUSTANG with an ROC value of only 0.79. In terms of Precision test, the combination of refined SAM-PROMALS3D are also shown to performs the best with a value of 0.03893. Refined SAM-3DCoffee on the other hand performs the worst with a value of only 0.02670. Recall test further shows that refined SAM-PROMALS3D performs the best with a value of 0.04147. In the mean time, refined HMMER-3DCoffee performs the worst with a value of only 0.02402.

From the overall result comparison, refined SAM-PROMALS3D are shown to performs the best in terms of ROC with a value of 0.87. Meanwhile, refined SAM-ProbCons performs the worst with an ROC of only 0.60. In terms of Precision, refined SAM-PROMALS3D are shown to performs the best with a value of 0.03893. On the other hand, refined SAM-T-Coffee combination performs the worst with a value of only 0.02061. In the mean time, Recall test result also shows that the combination of refined SAM-PROMALS3D performs the best with a value of 0.04147. Meanwhile, the worst combination is displayed by refined SAM-T-Coffee with a value of only 0.00973.

3.4. SVM-Fold Performance

This paper also assessed the performance of SVM-Fold using multiple alignments derived from CLUSTALW, MAFFT, ProbCons and T-Coffee for MSA and 3DCOFFEE, MAMMOTH-mult, MUSTANG and PROMALS3D for MStA. Figure 7 (a) displays the results of ROC for unrefined and refined SVM-Fold. Figure 7(b) and Figure 7(c) both shows the result of Precision and Recall test of unrefined and refined SVM-Fold respectively.

For MSA, refined SVM-Fold-MAFFT and SVM-Fold-T-Coffee both performs the best in terms of ROC with a value of 0.86 respectively. On the other hand, worst performance was displayed by the combination of refined SVM-Fold-ProbCons with an ROC value of only 0.61. For Precision test, refined SVM-Fold-ProbCons performs the best with a value of 0.029. Meanwhile, unrefined SVM-Fold-CLUSTALW shows the worst performance for this test with a value of only 0.018. In the meantime, Recall test shows that refined SVM-Fold-T-Coffee performs the best with a value of 0.028. Unrefined SVM-Fold-CLUSTALW on the other hand performs worst in this test with a value of only 0.011.

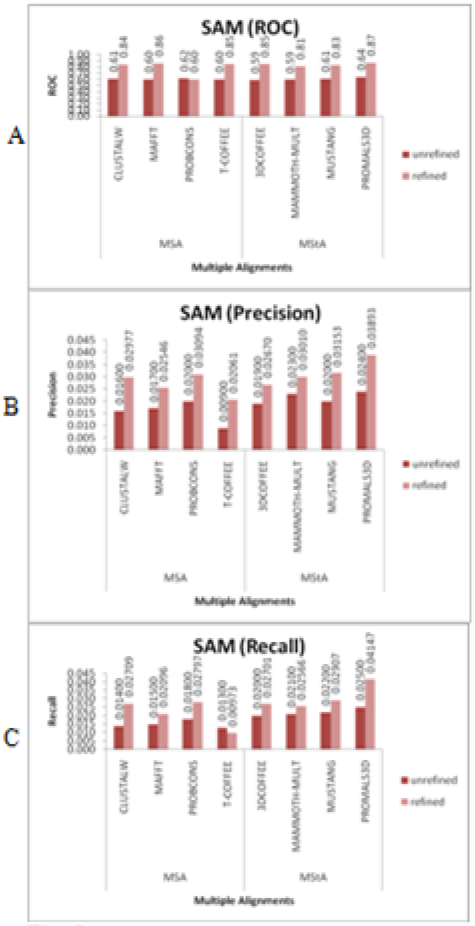


Figure 5

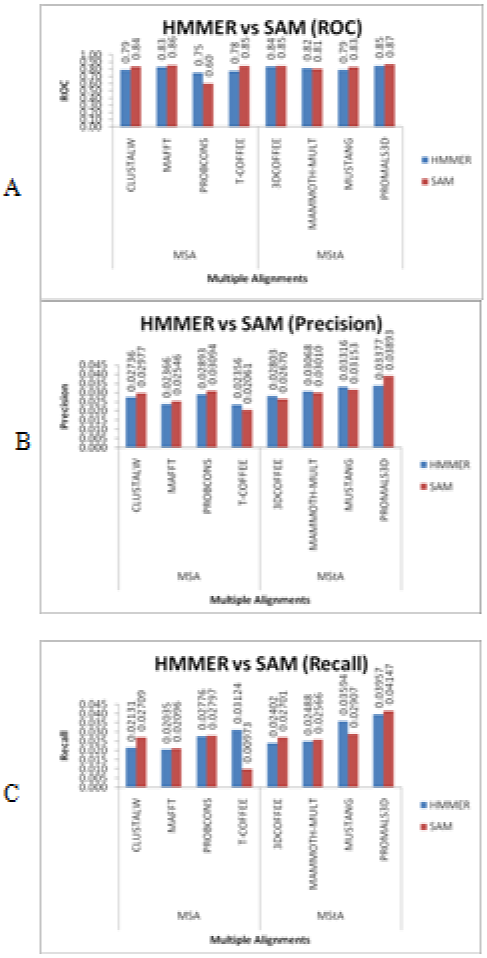


Figure 6

As for MStA, ROC test shows that refined SVM-Fold-3DCOFFEE performs the best with 0.87. Unrefined SVM-Fold-MAMMOTH-mult on the other hand performs worst with an ROC of only 0.57. Meanwhile, Precision test shows that refined SVM-Fold-PROMALS3D performs the best with a value of 0.035. In the meantime, unrefined SVM-Fold-MAMMOTH-mult performs the worst with a value of only 0.021. As for Recall test, refined SVM-Fold-PROMALS3D also performs the best with a value of 0.035. Meanwhile, unrefined SVM-Fold-MUSTANG displays the worst performance with a value of only 0.019.

Overall performance comparison between the combination of SVM-Fold with different type of multiple alignments shows that refined SVM-Fold-3DCOFFEE performs the best with an ROC of 0.87. On the other hand, worst performance is shown by the combination of unrefined SVM-Fold-MAMMOTH-mult with an ROC of only 0.57. Precision test in the meantime shows that the combination of refined SVM-Fold-PROMALS3D performs the best with a value of 0.035. On the other hand, unrefined SVM-Fold-CLUSTALW performs the worst in this test with a value of only 0.018. As for Recall test, refined SVM-Fold-PROMALS3D also performs the best with a value of 0.035. Unrefined SVM-Fold-CLUSTALW in the meantime performs the worst with a value of only 0.011.

3.5. SVM-Struct Performance

SVM-Struct performance in this paper is also assessed using multiple alignments from CLUSTALW, MAFFT, ProbCons and T-Coffee for MSA and 3DCOFFEE, MAMMOTH-mult, MUSTANG and PROMALS3D for MStA. Figure 8(a), Figure 8(b) and Figure 8(c) shows the ROC, Precision and Recall result of unrefined and refined SVM-Struct respectively. For MSA, the best performance in terms of ROC is shown by refined SVM-Struct-ProbCons with an ROC of 0.88. Meanwhile, unrefined SVM-Struct-T-Coffee performs worst with ROC of 0.69. Precision test also shows that refined SVM-Struct-ProbCons performs best with a value of 0.034. Unrefined SVM-Struct-CLUSTALW instead performs worst with a value of only 0.01732 compared to others in this test. As for Recall test, refined SVM-Struct-ProbCons once again comes best with a value of 0.035. Meanwhile, unrefined SVM-Struct-CLUSTALW performs the worst with a value of only 0.01363.

MStA derived SVM-Struct witnesses refined SVM-Struct-PROMALS3D to be performing the best compared to others in terms of ROC with a value of 0.89. On the other hand, unrefined SVM-Struct-MAMMOTH-mult performs the worst with an ROC of 0.6. Meanwhile, Precision test also shows that refined SVM-Struct-PROMALS3D performs the best with value of 0.039. Unrefined SVM-Struct-3DCOFFEE instead performs the worst with a value of only 0.0257. Results from Recall test also displays that refined SVM-Struct-PROMALS3D to perform the best with a value of 0.044. Meanwhile, unrefined SVM-Struct-3DCOFFEE shows the worst performance with a value of only 0.02301.

Overall comparison results between SVM-Struct MSA and MStA shows that refined SVM-Struct-PROMALS3D performs the best in terms of ROC with a value of 0.89. Meanwhile, the worst performance is shown by unrefined SVM-Struct-MAMMOTH-mult with an ROC of only 0.6. Precision test in the meantime also shows that refined SVM-Struct-PROMALS3D gives the best performance with a value of 0.039. Unrefined SVM-Struct-CLUSTALW however displays the worst performance with a value of only 0.01732. As for Recall test, the combinations of refined SVM-Struct-PROMALS3D are also shown to give the best performance with a value of 0.044. Meanwhile, the worst performance is displayed by the unrefined SVM-Struct-CLUSTALW with a value of 0.01363.

3.6. SVM-Fold and SVM-Struct Performance

In this section, results of refined SVM-Fold and SVM-Struct derived from multiple alignments tools that are CLUSTALW, MAFFT, ProbCons and T-Coffee for MSA and 3DCOFFEE, MAMMOTH-mult, MUSTANG and PROMALS3D for MStA are discussed. Figure 9 (a) shows the ROC results of refined SVM-Fold and SVM Struct. Figure 9 (b) and Figure 9 (c) both shows the results of refined Precision and Recall of SVM-Fold and SVM-Struct respectively.

For MSA, refined SVM-Struct-ProbCons performs the best with an ROC of 0.88. Meanwhile, the worst performance is shown by refined SVM-Fold-ProbCons with an ROC of only 0.61. Precision test on the other hand shows that refined SVM-Struct-ProbCons performs the best with a value of 0.034. In the meantime, the worst performance is displayed by refined SVM-Fold-CLUSTALW with a value of only 0.026. As for the Recall test, refined SVM-Struct-ProbCons also shows the best result of 0.035. Meanwhile, refined SVM-Struct-CLUSTALW performs the worst with value of only 0.021.

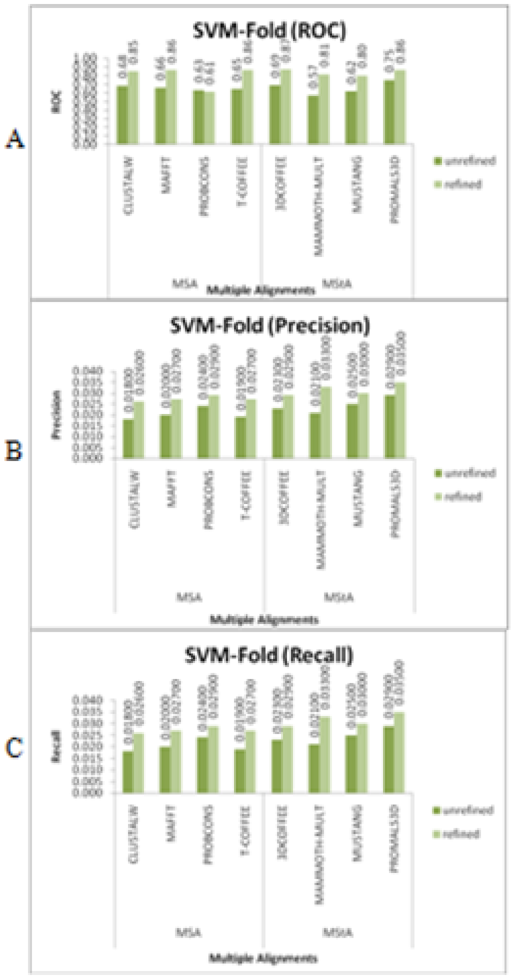


Figure 7

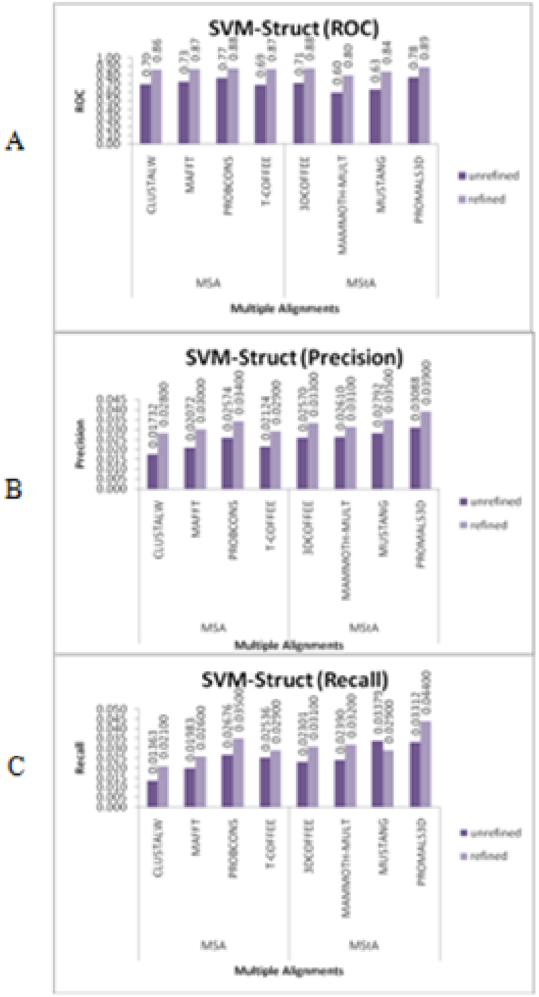


Figure 8

MStA derived refined SVM-Fold and SVM-Struct on the other hand shows that refined SVM-Struct-PROMALS3D performs the best in terms of ROC with a value of 0.89. Refined SVM-Struct-MAMMOTH-mult and SVM-Fold-MUSTANG instead performs the worst with an ROC value of only 0.8 respectively. Precision test also displays that refined SVM-Struct-PROMAL3D performs the best with a value of 0.039. Refined SVM-Fold-3DCOFFEE in the meantime performs the worst compared to other MStA tools with a value of only 0.029. Recall test on the other hand also shows that refined SVM-Struct-PROMALS3D performs the best with a value of 0.044. Meanwhile, the worst performance for Recall test is shown by refined SVM-Struct-MUSTANG with a value of only 0.029.

Overall performance comparison between refined SVM-Fold and SVM-Struct shows that refined SVM-Struct-PROMALS3D performs the best with an ROC of 0.89. Meanwhile, refined SVM-Fold-ProbCons shows the worst performance with an ROC of only 0.61. As for

Precision test, refined SVM-Struct-PROMALS3D also displays the best performance with a value of 0.039. Refined SVM-Fold-CLUSTALW in the meantime displays the worst performance with a value of only 0.026. Recall test also shows that refined SVM-Struct-PROMALS3D performs the best with a value of 0.044. Instead, the worst performance is shown by refined SVM-Struct-CLUSTALW with a value of only 0.021.

3.7. PHMMs and SVMs Performance

Lastly, the overall performance between pHMMs and SVMs derived from CLUSTALW, MAFFT, ProbCons and T-Coffee for MSA and 3DCOFFEE, MAMMOTH-mult, MUSTANG and PROMALS3D for MStA are also discussed. Figure 10(a) shows the comparison of ROC results of refined pHMMs and SVMs. Meanwhile, Figure 10(b) and Figure 10(c) displays the comparison of result of Precision and Recall tests for refined pHMMs and SVMs respectively.

For MSA, refined SVM-Struct-ProbCons performs the best in terms of ROC with a value of 0.88. Meanwhile, the worst performance is shown by refined SAM-ProbCons with an ROC of only 0.6. Precision test on the other hand shows that refined SVM-Struct-ProbCons performs the best with a value of 0.034. Refined SAM-T-Coffee in the meantime performs the worst with a value of 0.02061. As for Recall test, the best performance is also shown by refined SVM-Struct-ProbCons with a value of 0.035. Meanwhile, the worst performance is displayed by refined SAM-T-Coffee with a value of 0.00973.

For MStA, refined SVM-Struct-PROMALS3D performs the best in terms of ROC with a value of 0.89 respectively. The worst performance on the other hand is shown by refined HMMER-MUSTANG with a value of only 0.79. For Precision test, refined SVM-Struct-PROMALS3D displays the best performance with a value of 0.039. Meanwhile, the worst performance is displayed by refined SAM-3DCOFFEE with a value of only 0.0267. As for Recall test, the best performance is also shown by the combination of refined SVM-Struct-PROMALS3D with a value of 0.044. The worst performance in the meantime is shown by refined HMMER-3DCOFFEE with a value of only 0.02402.

Overall performance result comparison shows that refined SVM-Struct-PROMALS3D performs the best in terms of ROC with an ROC value of 0.89. On the other hand, the worst performance is shown by refined SAM-ProbCons with an ROC value of only 0.6. Precision test in the meantime witnesses that refined SVM-Struct-PROMALS3D also performs the best with a value of 0.039. On the other hand, refined SAM-T-Coffee shows the worst performance with a value of 0.02061. Recall test also displays that refined SVM-Struct-PROMALS3D performs the best with a value of 0.044. Meanwhile, the worst performance is shown by refined SAM-T-Coffee with a value of 0.00973.

4. Discussion

4.1. PROMALS3D: The Best Multiple Alignments

Based on the results from this paper, PROMALS3D can be seen to perform the best for multiple alignments. It is believed that PROMALS3D is able to achieve the best performance by deriving structural constraints from representative sequences with known structures. Furthermore, PROMALS3D uses 3D structural information to guide multiple alignments constructed using PROMALS. By automatically identifying homologs with known 3D structures, deriving structural constraints through structure-based multiple alignments and then combining them with sequence constraints to construct consistency-based multiple alignments, this software is capable to output a consensus multiple alignment which brought together sequence and structural information.

4.2. SVM-Struct: The Best Classification Algorithm

As for classification algorithm, SVMs which in this case SVM-Struct employs discriminative classification approaches that are proven to be superior compared to other methods including pHMMs. Discriminative approaches typically would estimate posterior probabilities directly without attempting to model underlying probability distributions [55]. This would further focuses computational resources more on specific task which would in turn resulted on increasing performance. The SVM-Struct is specifically designed for prediction of complex outputs such as multiple alignments and has more superior generalization performance. Unlike regular SVMs, SVM-Struct considers multivariate and structured outputs. Furthermore, it also employs the 1-slack cutting-plane algorithm which uses new but equivalent formulation of the structural SVMs quadratic program. It is also several orders of magnitude faster than other methods.

4.3. REFINER: The Impact or Refinement Algorithm

The performance of the optimal mesh algorithm is further enhanced through the application of the realignment algorithm which increases the quality of the multiple alignments through conserved core iterative realignment of the protein block model. The refinement algorithm is applied because multiple alignment which employs progressive approach have the problem whereby the misalignments made at previous stages cannot be corrected afterwards. This will further disseminate into serious alignment errors. Making things worse, the final alignment strongly depends on the order of sequences aligned. The use of a refinement algorithm can correct misalignments between a given sequence and the rest of the profile and at the same time preserves the family’s overall block model. The ROC test shows that the performance of RPHD has been increased for 14.1% through the application of the refinement algorithm.

4.4. PRS: The Optimal Mesh Algorithm

Based on the findings of this paper, the combination of PROMALS3D, REFINER and SVM-Struct (PRS) produces an optimal mesh algorithm for RPHD. By implementing an approach of utilizing 3D structural information to act as guide for multiple alignment construction, the algorithm is further enhanced through the use of discriminative classification approach that are proven to be superior. The algorithm also employs a refinement algorithm to further increase the quality of the multiple alignments, which in turn increased the performance for RPHD.

5. Conclusions

Throughout this paper, a data pre-processing procedure has been introduced to prepare the datasets for the RPHD from SCOP 1.73. Furthermore, different combinations between multiple sequence alignments and multiple structural alignments programs with pHMMs and SVMs has been introduced and integrated in order to construct an optimal mesh algorithm for RPHD. The optimal mesh algorithm consists of three main stages that are the multiple alignments stage, realignment of protein sequences stage and the classification and evaluation stage. A refining algorithm has also been applied to reduce misalignments in multiple

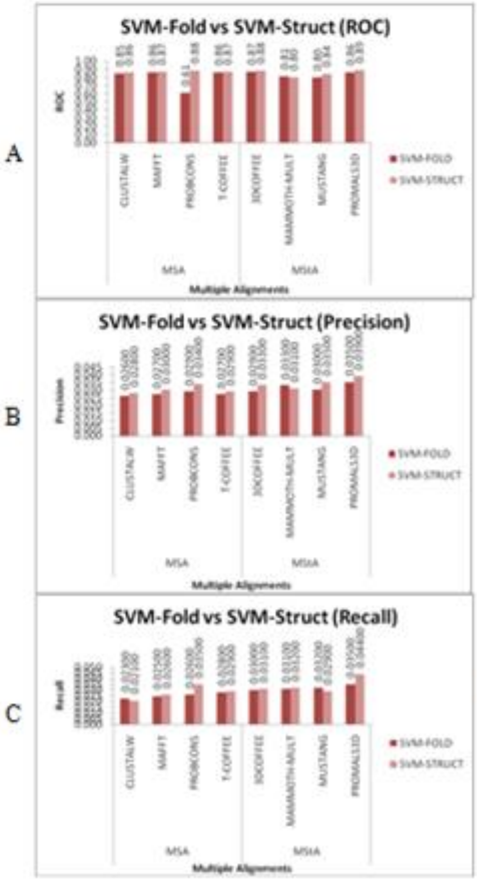


Figure 9

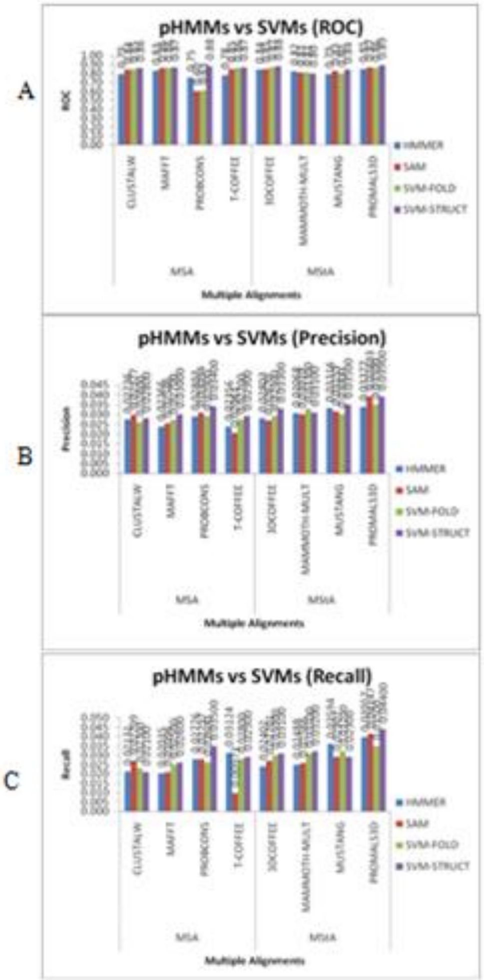


Figure 10

sequence alignments and multiple structural alignments, thus further assisting in accurate RPHD. The results for RPHD using SCOP 1.73 can be introduced to be utilized as a benchmark result for other researchers to further improve the detection of remote protein homology via identifying the best combination of multiple alignments and classifier programs that include pHMMs for generative classifier and SVMs for discriminative classifier.

Based on the result of the experiments, the use of realignment algorithm clearly improves the detection of remote protein homology by the programs chosen. This can be inducted from the performance graphs that shows realigned multiple alignments outperforms multiple alignment which is not realigned. Meanwhile, the optimal mesh algorithm is displayed by the refined SVM-Struct-PROMALS3D.

The results and findings from this paper can be further utilized such as by integrating biological information, for example from Gene Ontology in order to further increase its capability. Apart from that, the step of feature extraction can also be improvised by applying non-negative matrix factorization to further increase the performance of the optimized algorithm. Lastly, the algorithm designed in this paper can also be utilised by transforming it into a web service which will enable researchers to perform RPHD.

Acknowledgements

This project is funded by Malaysian Ministry of Higher Education under Fundamental Research Grant Scheme (project no. 78185). All authors conceived the study, participated in its design and coordination, helped to draft the manuscript, read, and approved the final manuscript.

References

[1] M. Madera, J. Gough, "A comparison of profile hidden markov model procedures for remote homology detection", Nucleic Acids Research, vol. 30, 2002, pp. 4321-4328.

[2] P. Bourne, H. Weissig (Eds.) "Structural Bioinformatics. Hoboken", NJ: Wiley-Liss; 2003.

[3] C.S. Leslie, E. Eskin, A. Cohen, J. Weston, W.S. Noble, "Mismatch string kernels for discriminative protein classification", Bioinformatics, vol. 20, 2004, pp. 467-476.

[4] T. Jaakkola, M. Diekhans, D. Haussler, "A discriminative framework for detecting remote protein homologies", Journal of Computational Biology, vol. 7, 2000, pp. 95-114.

[5] L. Liao, W.S. Noble, "Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships", Journal of Computational Biology, vol. 10, 2003, pp. 857-868.

[6] S. Chakrabarti, C.J. Lanczycki, A.R. Panchenko, T.M. Przytycka, P.A. Thiessen, S.H. Bryant, "Refining multiple sequence alignments with conserved core regions", Nucleic Acids Research, vol. 34, 2006, pp. 2598-2606.

[7] R.C. Edgar, S. Batzoglou, "Multiple sequence alignment", Current Opinion in Structural Biology, vol. 16, 2006, pp. 368-373.

[8] J. Pei, N.V. Grishin, "MUMMALS: Multiple sequence alignment improved by using hidden markov models with local structural information", Nucleic Acids Research, vol. 34, 2006, pp. 4364-4374.

[9] A. Subramanian, M. Kaufmann, B. Morgenstern, "DIALIGN-TX: Greedy and progressive approaches for segment-based multiple sequence alignment", Algorithms for Molecular Biology, vol. 3, 2008, pp. 6-17.

[10] N. Bray, L. Pachter, "MAVID: Constrained ancestral alignment of multiple sequences", Genome Research, vol. 14, 2004, pp. 693-699.

[11] M.A. Suchard, B.D. Redelings, "BAli-Phy: Simultaneous bayesian inference of alignment and phylogeny", Bioinformatics, vol. 22, 2006, pp. 2047-2048.

[12] F.B. Sheinerman, B. Al-Lazikani, B. Honig, "Sequence, structure and energetic determinants of phosphopeptide selectivity of SH2 domains", Journal of Molecular Biology, vol. 334, 2003, pp. 823-841.

[13] B. Al-Lazikani, F.B. Sheinerman, B. Honig, "Combining multiple structure and sequence alignments to improve sequence detection and alignment: application to the SH2 domains of Janus kinases", PNAS, vol. 98, 2001, pp. 14796-14801.

[14] T. Oldfield, "CAALIGN: A program for pairwise and multiple protein-structure alignment", Acta Crystallographica Section D, vol. 63, 2007, pp. 514-525.

[15] F. Birzele, J.E. Gewehr, G. Csaba, R. Zimmer, "Vorolign-fast structural alignment using voronoi contacts", Bioinformatics, vol. 23, 2007, pp. e205-e211.

[16] M. Menke, B. Berger, L. Cowen, "Matt: local flexibility aids protein multiple structure alignment", PLoS Computational Biology, vol. 4, 2008, no. e10.

[17] Y. Ye, A. Godzik, "Multiple flexible structure alignment using partial order graphs", Bioinformatics, vol. 21, 2005, pp. 2362-2369.

[18] J. Dai, J. Cheng, "HMMEditor: A visual editing tool for profile hidden markov mode", BMC Genomics, vol. 9, 2008, no. S8.

[19] M. Madera, "Profile Comparer: A program for scoring and aligning profile hidden markov models", Bioinformatics, vol. 24, 2008, pp. 2630-2631.

[20] W.N. Grundy, T.L. Bailey, C.P. Elkan, M.E. Baker, "Meta-MEME: Motif-based hidden markov models of protein families", Computer Applications in the Biosciences, vol. 13, 1997, pp. 397-406.

[21] E. Birney, M. Clamp, R. Durbin, "GeneWise and Genomewise", Genome Research, vol. 14, 2004, pp. 988-995.

[22] P. Pavlidis, I. Wapinski, W.S. Noble, "Support vector machine classification on the web", Bioinformatics, vol. 20, 2004, pp. 586-587.

[23] M. Pirooznia, Y. Deng, "SVM Classifier - A comprehensive java interface for support vector machine classification of microarray data", BMC Bioinformatics, vol. 7, 2006, no. S25.

[24] C.Z. Cai, L.Y. Han, Z.L. Ji, X. Chen, Y.Z. Chen, "SVM-Prot: Web-based support vector machine software for functional classification of a protein from its primary sequence", Nucleic Acids Research, vol. 31, 2003, pp. 3692-3697.

[25] I. Melvin, E. Ie, R. Kuang, J. Weston, W. Noble, C. Leslie, " SVM-Fold: A tool for discriminative multi-class protein fold and superfamily recognition", BMC Bioinformatics, vol. 8, 2007, no. S2.

[26] A. Manohar, S. Batzoglou, "TreeRefiner: A tool for refining a multiple alignment on a phylogenetic tree", In Proceeding of the 4th International IEEE Computer Society Computational Systems Bioinformatics Conference, 2005, pp. 111-119.

[27] C. Notredame, L. Holm, D.G. Higgins, "COFFEE: An objective function for multiple sequence alignments", Bioinformatics, vol. 14, 1998, pp. 407-422.

[28] R. Edgar, "MUSCLE: A multiple sequence alignment method with reduced time and space complexity", BMC Bioinformatics, vol. 5, 2004, pp. 113-132.

[29] I.M. Wallace, O. O'Sullivan, D.G. Higgins, "Evaluation of iterative alignment algorithms for multiple alignment", Bioinformatics, vol. 21, 2005, pp. 1408-1414.

[30] M.A. Larkin, G. Blackshields, N.P. Brown, R. Chenna, P.A. McGettigan, H. McWilliam, F. Valentin, I.M. Wallace, A. Wilm, R. Lopez, et al, "Clustal W and Clustal X version 2.0", Bioinformatics, vol. 23, 2007, pp. 2947-2948.

[31] K. Katoh, K. Misawa, Ki. Kuma, T. Miyata, "MAFFT: A novel method for rapid multiple sequence alignment based on fast Fourier transform", Nucleic Acids Research, vol. 30, 2002, pp. 3059-3066.

[32] C.B. Do, M.S.P. Mahabhashyam, M. Brudno, S. Batzoglou, "PROBCONS: Probabilistic consistency-based multiple sequence alignment", Genome Research, vol. 15, 2005, pp. 330-340.

[33] C. Notredame, D.G. Higgins, J. Heringa, "T-Coffee: A novel method for fast and accurate multiple sequence alignment", Journal of Molecular Biology, vol. 302, 2000, pp. 205-217.

[34] O. O'Sullivan, K. Suhre, C. Abergel, D.G. Higgins, C. Notredame, "3DCoffee: Combining protein sequences and structures within multiple sequence alignments", Journal of Molecular Biology, vol. 340, 2004, pp. 385–395.

[35] D. Lupyan, A. Leo-Macias, A.R. Ortiz, "A new progressive-iterative algorithm for multiple structure alignment", Bioinformatics, vol. 21, 2005, pp. 3255-3263.

[36] A.S. Konagurthu, J.C. Whisstock, P.J. Stuckey, A.M. Lesk, "MUSTANG: A multiple structural alignment algorithm", Protein Science, vol. 64, 2006, pp. 559-574.

[37] M.G. Kann, P.A. Thiessen, A.R. Panchenko, A.A. Schaffer, S.F. Altschul, S.H. Bryant, "A structure-based method for protein sequence alignment", Bioinformatics, vol. 21, 2005, pp. 1451-1456.

[38] S.R. Eddy, "Profile hidden Markov models", Bioinformatics, vol. 14, 1998, pp. 755-763.

[39] K. Karplus, C. Barrett, R. Hughey, "Hidden Markov Models for detecting remote protein homologies", Bioinformatics, vol. 14, 1998, pp. 846-856.

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

[41] I. Melvin, E. Ie, R. Kuang, J. Weston, W. Noble, C. Leslie, "SVM-Fold: A tool for discriminative multi-class protein fold and superfamily recognition", BMC Bioinformatics, vol. 8, 2007, no. S2.

[42] J. Bernardes, A. Davila, V. Costa, G. Zaverucha, "Improving model construction of profile HMMs for remote homology detection through structural alignment", BMC Bioinformatics, vol. 8, 2007, pp. 435-447.

[43] S. Chakrabarti, C.J. Lanczycki, A.R. Panchenko, T.M. Przytycka, P.A. Thiessen, S.H. Bryant, "Refining multiple sequence alignments with conserved core regions", Nucleic Acids Research, vol. 34, 2006, pp. 2598-2606.

[44] A. Marchler-Bauer, J.B. Anderson, F. Chitsaz, M.K. Derbyshire, C. DeWeese-Scott, J.H. Fong, L.Y. Geer, R.C. Geer, N.R. Gonzales, M. Gwadz, et al, "CDD: specific functional annotation with the Conserved Domain Database", Nucleic Acids Research, vol. 37, 2009, pp. D205- D210.

[45] R.D. Finn, J. Tate, J. Mistry, P.C. Coggill, S.J. Sammut, H-R. Hotz, G. Ceric, K. Forslund, S.R. Eddy, E.L.L. Sonnhammer, A. Bateman, "The Pfam protein families database", Nucleic Acids Research, vol. 36, 2008, pp. D281- D288.

[46] A. Andreeva, D. Howorth, S.E. Brenner, T.J.P. Hubbard, C. Chothia, A.G. Murzin, "SCOP database in 2004: Refinements integrate structure and sequence family data", Nucleic Acids Research, vol. 32, 2004, pp. D226-D229.

[47] P. Sonego, A. Kocsor, S. Pongor, "ROC analysis: Applications to the classification of biological sequences and 3D structures", Briefings in Bioinformatics, vol. 9, 2008, pp. 198-209.

[48] J. Supper, L. Spangenberg, H. Planatscher, A. Draeger, A. Schroeder, A. Zell, "BowTieBuilder: modeling signal transduction pathways", BMC Systems Biology, vol. 3, 2009, no. 67.

[49] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne, "The protein data bank", Nucleic Acids Research, vol. 28, 2000, pp. 235-242.

[50] K. Katoh, K. Kuma, H. Toh, T. Miyata, "MAFFT Version 5: Improvement in accuracy of multiple sequence alignment", Nucleic Acids Research, vol. 33, 2005, pp. 511-518.

[51] S. Henikoff, J.G. Henikoff, "Amino acid substitution matrices from protein blocks", Proceeding of the National Academy of Sciences of the United States of America, vol. 89, 1992, pp. 10915-10919.

[52] W.R. Taylor, C.A. Orengo, "Protein structure alignment", Journal of Molecular Biology, vol. 208, 1989, pp. 1-22.

[53] J. Shia, T.L. Blundella, K. Mizuguchia, "FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties", Journal of Molecular Biology, vol. 310, 2000, pp. 243-257

[54] M. Gribskov, N.L. Robinson, "Use of Receiver Operating Characteristic (ROC) analysis to evaluate sequence matching", Computers & Chemistry, vol. 20, 1996, pp. 25-33.

[55] K. Kedem, L.P. Chew, R. Elber, "Unit-vector RMS (URMS) as a tool to analyze molecular dynamics trajectories", Proteins, vol. 37, 1999, pp. 554-564.

[56] J. Pei, N.V. Grishin, "PROMALS: towards accurate multiple sequence alignments of distantly related proteins", Bioinformatics, vol. 23, 2007, pp. 802–808.

[57] Q. Wang, E. Song, R. Jin, P. Han, X. Wang, Y. Zhou, J. Zeng, "Segmentation of lung nodules in computed tomography images using dynamic programming and multidirection fusion techniques", Academic Radiology, vol. 16, 2009, pp. 678-688.

[58] K. Sato, K. Morita, Y. Sakakibara, "PSSMTS: position specific scoring matrices on tree structures", Journal of Mathematical Biology, vol. 56, 2008, pp. 201-214.

[59] A.F. Neuwald, A. Poleksic, "PSI-BLAST searches using hidden Markov models of structural repeats: prediction of an unusual sliding DNA clamp and of ß-propellers in UV-damaged DNA-binding protein", Nucleic Acids Research, vol. 28, 2000, pp. 3570-3580.

[60] A.Y. Ng, M.I. Jordan, "On discriminative vs generative classification algorithm: A comparison of logistic regression and Naive Bayes", In Advances in Neural Information Processing Systems (NIPS) 14. Edited by Dietterich T, Becker S, Ghahramani Z. Vancouver, Canada: MIT Press; 2001: 841-848.

Authors

Firdaus M. Abdullah is a Researcher at the Laboratory of Computational Intelligence and Biotechnology at the Universiti Teknologi Malaysia. He received the B.Sc. and M.Sc. degrees in Computer Science both from the Universiti Teknologi Malaysia, in 2006 and 2010, respectively. His research interests focus on remote protein homology detection, machine learning algorithm, and computational biology.

Razib M. Othman is a Director of Laboratory of Computational Intelligence and Biotechnology at the Universiti Teknologi Malaysia. He received the B.Sc, M.Sc. and Ph.D. degrees in Computer Science from the Universiti Teknologi Malaysia, in 1993, 2003, and 2008, respectively. His research interests are in the areas of computational intelligence, computational biology, and software engineering.

Shahreen Kasim is a Tutor at the Faculty of Computer Science and Information Technology, the Universiti Tun Hussein Onn Malaysia. She received the B.Sc., M.Sc., and Ph.D degrees in Computer Science from the Universiti Teknologi Malaysia, in 2003, 2005, and 2011 respectively. Her research interests focus on gene function prediction, clustering algorithm, and computational biology.

Rathiah Hashim is a Senior Lecturer at the Faculty of Computer Science and Information Technology, the Universiti Tun Hussein Onn Malaysia. She received the Ph.D. degree in Visualization and Psychology from Swansea University, UK in 2008, the M.Sc. degree in Computer Science from the Universiti Teknologi Malaysia in 2000, and the B.Sc. degree in Computer Science from Wichita State University, USA in 1986. Her research interests focus on video visualization, image processing, psychology (visual perception), and human computer interface.

Rohayanti Hassan is a Lecturer at the Faculty of Computer Science and Information Systems, the Universiti Teknologi Malaysia. She received the B.Sc., M.Sc., and Ph.D degrees in Computer Science from the Universiti Teknologi Malaysia, in 2003, 2006, and 2011 respectively. Her research interests focus on protein structure prediction, clustering algorithm, and computational biology.

Hishammudin Asmuni is a Senior Lecturer at the Faculty of Computer Science and Information Systems, the Universiti Teknologi Malaysia. He received the Ph.D. degree in Computer Science from The University of Nottingham, UK in 2008, the M.Sc. degree in Computer Science from the Universiti Teknologi Malaysia in 1999, and the B.Sc. degree in Computer Science from the Universiti Malaya in 1996. His research interests focus on timetabling/scheduling, fuzzy systems, and bioinformatics.

Jumail Taliba is a Lecturer at the Faculty of Computer Science and Information Systems, the Universiti Teknologi Malaysia. He received the B.Sc. and M.Sc. degrees in Computer Science both from the Universiti Teknologi Malaysia, in 1997, and 2001, respectively. His research interests focus on protein-protein interaction prediction, image processing algorithm, and computational biology.