UNIVERSITY OF CALIFORNIA
RIVERSIDE

Improving Multiple Sequence Alignments by Revising Sequence Families with Alignment Scoring Approaches

A Thesis submitted in partial satisfaction
of the requirements for the degree of

Master of Science

in

Computer Science

by

Aleksandr Olegovich Levchuk

December 2011

Thesis Committee:
Dr. Thomas Girke, Chairperson
Dr. Eamonn Keogh
Dr. Stefano Lonardi

The Thesis of Aleksandr Olegovich Levchuk is approved

Committee Chairperson

University of California, Riverside

Acknowledgments

I am grateful to all the people who supported me on this journey. I would like to thank my advisor Dr. Thomas Girke without whom this project would not have been possible. I thank my committee members Dr. Eamonn Keogh and Dr. Stefano Lonardi for critically reading this research thesis, and for inspiring me through courses and projects in Data Mining and Bioinformatics. In addition, I thank my friends, family, and fellow graduate students who supported me continuously and gave practical advise.

ABSTRACT OF THE THESIS

Improving Multiple Sequence Alignments by Revising Sequence Families with Alignment Scoring Approaches

by

Aleksandr Olegovich Levchuk

Master of Science, Graduate Program in Computer Science
University of California, Riverside, December 2011
Thomas Girke, Chairperson

Characterizing the functional, structural, and evolutionary relationships of biological sequences is an important task in modern genomics and computational biology. Most of these applications involve the assembly of sequence families by similarity searching, subsequent formation of multiple sequence alignments (MSAs) and downstream phylogenetic analyses. Especially, MSAs play a central role in this modeling workflow. Thus, the quality of the MSAs is of critical importance for its success. In this study I present an approach to improve the quality of MSAs by using a sequence family revision approach that can automatically remove false positive candidates from sequence families and then recompute an improved MSA. The approach is able to combine sequence-level scores from two MSA scoring methods, norMD and GUIDANCE. It automatically selects an optimized score threshold for removing sequences from MSAs. To test the performance of this method, I developed several automated procedures to add to curated MSAs from the CDD database controlled numbers of randomly selected nonmember sequences. Then I performed Receiver Operating Characteristic (ROC) analysis on the classification results incorporating automatic threshold selection approaches. Surprisingly, the sequence-level scores, provided by the two MSA scoring methods, were less successful than a simple all-against-all BLAST-based pairwise alignment scoring approach. However, I was able to improve one of the MSA scoring methods by extending it with a dynamic threshold selection approach. The extended method outperformed the performance of the BLAST-based method in detecting false positives in sequence families.

Contents

1 Introduction 1

1.1 Pairwise and Multiple Sequence Alignment 4

1.2 Methods for Generating MSAs 4

1.3 Types of Sequences 5

1.4 MSA Improvement Strategies 7

1.5 Application of MSA scoring 7

1.6 Ways of introducing randomness into MSAs 8

1.7 Objectives 9

2 MSA Scoring Methods 9

2.1 SP-score with Reference 9

2.2 SP-score without Reference 10

2.3 norMD 11

2.4 Neighbor-joining tree assisted norMD sequence removal 12

2.5 GUIDANCE 12

2.6 Gblocks 13

2.7 PredictedSP 14

2.8 PSAR 14

2.9 Scoring Methods Summary Table 15

3 Methods 16

3.1 Experimental Strategy 16

3.2 Member and Nonmember Sequences 16

3.3 Separation of Sequence-level MSA scores 17

3.4 Spiking: Adding Nonmembers to Sequence Families 19

3.5 Datasets 19
3.6 Scoring with GUIDANCE 21
3.7 Scoring with norMD 21
3.8 Scoring with BLAST 22
3.9 Classifier Evaluation 22
3.10 Combined Classification 24
3.11 Threshold Selection 25
3.11.1 Limits of Static Threshold Selection 25
3.11.2 Dynamic Threshold Selection 26
3.11.3 Limits of Dynamic Threshold Selection 27
3.11.4 Dynamic Threshold Selection by Min/Max Scaling 28
3.11.5 Dynamic Threshold Selection by Balancing MCC 29

4 Results and Discussion 32
4.1 Scores of Entire MSAs 32
4.2 Score Separation 34
4.3 Score Separation on Families with Highly Related Members 35
4.4 Static Threshold Selection 37
4.5 Dynamic Threshold by Calibration of Two Scoring Methods 40
4.6 Dynamic Threshold Selection by Min/Max Scaling 42
4.7 Scoring with BLAST 43
4.8 Comparing Scoring Methods with ROCs 44
4.9 BLAST-based and GUIDANCE-based cleanup methods 48

5 Conclusions 49

6 Contributions 50

7 Future work 51

8 Implementation 52

9 Availability 53

10 Supplements 53

References 54

List of Figures

1 Typical Workflow for Modeling Sequence Families 2

2 A multiple sequence alignment 3

3 GUIDANCE MSA-level scores 32

4 norMD whole MSA scores 33

5 Separation of sequence-level scores 34

6 Separation of norMD scores of 50% identity sequences Sample 2 subset 35

7 GUIDANCE and norMD sequence-level scores of MSAs with highly related member sequences and 5% nonmembers. 36

8 Static Thresholds 38

9 Counts of mistakes in static MCC threshold selection 39

10 Dynamic threshold selection 41

11 Performance of the static threshold selection method 41

12 Dynamic threshold selection by scaling 42

13 Comparison of BLAST and combined GUIDANCE+norMD scores . . 43

14 ROC for norMD, GUIDANCE, and BLAST without threshold selection 45

15 Comparison of Area Under the ROC curve (AUC) for different methods 47

16 AUC Zoom-ins for best results from Figure 15 48

List of Tables

1 MSA Scoring methods comparison 15

2 Scaling scores around a threshold T 28

3 MSA sequence score scaling 28

4 Static Thresholds 37

5 Acronyms of Test Approaches 46

6 External and Internal dependencies 52

1 Introduction

Biological sequences are related through a common evolutionary history and function. The members of a family have diverged from each other by primary sequence changes during evolution, by duplication events in the genome, or by speciation events giving rise to related sequences in different species. Normally, diverging sequences maintain the same or related functions. For functional and evolutionary analyses of sequences, it is important to identify all or most members in a family through database searches by sequence similarity, align them, and represent their relationships in a tree (Figure 1; Durbin et al., 1999a; Eddy, 1998; Bateman et al., 2004).

1. Database Searches to Identify Family Members

Important: sequences should show significant similarity.

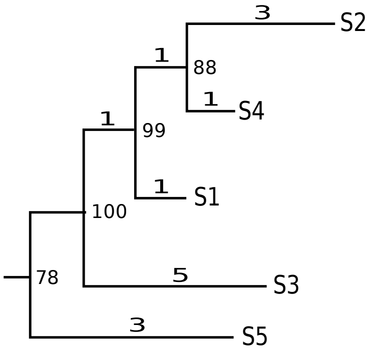
2. Multiple Sequence Alignment of Family Members

Important: unalignable sequence areas should be removed.

```
S1          FMPFSAGKRICAGEGLARMELFLFLT   450
S2          FMPFSAGKRICVGEALAGMELFLFLT   450
S3          .LAFGCGARVCLGEPLARLELFVVLT   443
S4          SLPFGFGKRSCMGRRLAELELQMALA   470
S5          YTPFGSGPRNCIGMRFALMNMKLALI   457
consensus   ..PFg.GkR.C.Ge.LA.mELfl.Lt
```

3. Compute a Distance Matrix for Multiple Sequence Alignment

	S1	S2	S3	S4	S5
S1	0.0	0.43	0.71	0.71	0.48
S2		0.0	0.57	0.57	0.39
S3			0.0	0.29	0.21
S4				0.0	0.13
S5					0.0



4. Calculate Phylogenetic Tree

Important: tree building, rooting, and bootstrapping methods.

Figure 1: *Typical Workflow for Modeling Sequence Families.*

One of the best models for representing the similarities among the members of a sequence family is a multiple sequence alignment (MSA). An MSA is a prerequisite for most comparative sequence studies in molecular biology. Specifically, MSAs are used for analyzing phylogenies, conserved motifs, domains and protein structures (Penn et al., 2010). MSAs are also useful for large-scale database searches using profiles of sequence families (Durbin et al., 1999a; Girke et al., 2004).

The construction of MSAs involves a stepwise process that models the substitutions, insertions and deletions occuring over time in gene and protein sequences (Felsenstein, 2003). Protein or DNA sequences are formed into an MSA alignment by superimposition of three or more sequences so that the same or related residues are exactly aligned to one another. The most direct representation of an MSA is a matrix of characters where gaps are inserted in order to fill the voids between unaligned characters.

```
VSCDG-CGK--SNFT-GRRYKCLIC---YDYDLCADCYDSGVT--------TERHLFDHPMQCI
VSCDA-CLK--GNFR-GRRYKCLIC---YDYDLCASCYESGAT--------TTRHTTDHPMQCI
VSCDA-CLK--GNFR-GRRYKCLIC---YDYDLCASCYESGAT--------TTRHTTEHPMQCI
VSCDG-CAF--TAFA-GNRYKCLRC---SDYDLCFSCFTTKNYGDQQTIADIPIHDESHPMQLI
ATCDG-CDLWGNGIT-GCRYKCLKC---ADFDLCKSCYDAKVV--------SGRHKSEHPMQCL
VGCDS-CGM--YPIR-GKRYKCKDCTELIGFDLCEECYNTKSKLPG---RFNQHHTPDHRMELD
AGCDS-CGV--YPII-GDRYRCKDCKEEIGYDLCKDCYETPSKVPG---RFNQQHTPDHRLELA
```

Figure 2: *A multiple sequence alignment.* 7 sequences of the cd02338 protein group.

The alignment of the sequence family cd02338 (Bauer, 2011) is an example of an MSA shown in Figure 2. In this representation every letter represents an amino acid residue and every row is a protein sequence - a member of this sequence family. One can use such an MSA for a wide spectrum of downstraem analysis methods. These include the following approaches:

A **profile Hidden Markov Model (HMM)** (Eddy, 1998) is a model of the conserved regions in an MSA. These domain models can be used as a search query for finding closely and distantly related family members in a protein database. Typically, this approach will be more sensitive in finding distantly related members than database searches based on pairwise alignment approaches, such as BLAST (Altschul et al., 1990).

A **Phylogenetic Tree** can be inferred from an MSA by a vast array of tree building methods (Felsenstein, 2003). One of these techniques is the neighbor-joining method (Saitou and Nei, 1987) that hierarchically clusters the sequences by their pairwise similarities while constructing a phylogram. Phyogenetic approaches can predict evolutionary events such as ancestry and time of speciation.

1.1 Pairwise and Multiple Sequence Alignment

Pairwise sequences alignments are the basis of most approaches for searching sequence databases. This includes the widely used BLAST software (Altschul et al., 1990), which has many additional applications, including sequence clustering. When pairs of sequences are aligned, a static evolutionary rate is assumed (*e.g.* a static amino acid substitution matrix like BLOSUM-62). This is an oversimplification because within one sequence the evolutionary rate can vary from total invariance to extreme variability (Thompson et al., 2001). When sequences alignments are performed on multiple related sequences, the evolutionary rates can be inferred from MSA column statistics which are not available when only two sequences are used. For this reason database searches, performed with sequence family models (e.g. profile HMMs), are more sensitive.

1.2 Methods for Generating MSAs

Forming MSAs of more than two sequences is an NP-complete problem (Wang and Jiang, 1994). Thus, approximation approaches are employed to construct MSAs in a time-efficient manner. The progressive alignment method is a common heuristic used by MSA Various methods are used to approximate the relationships without aligning the sequences, for example MUSCLE aligner (Edgar, 2004b) uses the kmer distance (Edgar, 2004a). The relationships are summarized as a tree where the leaves represent the sequences and the branches represent the pair-wise relationships. The sequences are then pairwise aligned in the order that is determined by the guide tree starting with the most similar pairs. The pair-wise alignments seek to maximize an objective function. In its pure form the progressive alignment heuristic aligns any two sequence only once. Once set, the alignment of residues relative to each other is not changed as more sequences are incorporated into a progressively growing MSA.

Iterative alignment is a strategy of refining an existing MSA in a series of iterations that modify the alignment in order to increase the score of an objective function. In modern programs such as MUSCLE and MAFFT, the iterative procedure is applied after preforming a progressive alignment. MUSCLE (Edgar, 2004b) goes through two stages of progressive alignment and then refines the MSA in an iterative stage where it applies a variation of tree-dependent restricted partitioning (Hirosawa et al., 1995). MAFFT applies Fast Fourier transform iterative refinement method FFT-NS-i (Katoh et al., 2002).

MSA programs produce alignments which may disagree with the evolutionary correct solution. These mistakes may occur due to a variety of reasons: First, most MSA programs implement heuristic approaches that can lead to suboptimal solutions. Second, progressive alignment techniques also approximate the guide tree. Third, several co-optimal solutions are common. Fourth, the objective function is a simplification of the process of evolution. Fifth, due to the stochastic nature of sequence evolution the biologically correct alignment may be sub-optimal. (Landan and Graur, 2007; Penn et al., 2010). Sixth, the sequences of a family may be too diverse to generate any reliable pairwise alignments or MSAs. In addition, one can expect that alignment mistakes will be made more often when unrelated sequences are provided as input (e.g. false positives). In fact when too diverse sequences are provided as input to an MSA software, then the results are often of poor quality.

1.3 Types of Sequences

The residue sequences of DNA, RNA, and protein molecules are represented by different alphabets. DNA and RNA sequences have an alphabet of size four, while the typical alphabet for protein sequences is composed of twenty characters. Specialty characters are sometimes used as place holders for cases when there is ambiguity in a particular sequence position.

When comparing diverse sequences, and there is a choice of using either gene (DNA/RNA) or protein sequences, then the use of the latter is often preferred, because the protein alphabet is more information-rich which increases the sensitivity of many analysis routines, such as similarity searching or alignment methods. Additional reasons for this preference are: (1) Amino acid substitutions in protein sequences are functionally more relevant than nucleotide substitutions in DNA sequences, because amino acids differ in their physicochemical and functional properties. As a result, substitutions in protein sequences can be scored with empirically derived substitution matrices, such as PAM (Dayhoff and Schwartz, 1978) or BLOSUM (Henikoff and Henikoff, 1992; Gonnet et al., 1992) which estimate the likelihood of one amino acid mutating into another. In contrast to this, nucleotide substitutions are more neutral with relatively homogenous substitution frequencies. (2) Due to the redundancy of the genetic code, differences in DNA sequences can be synonymous, meaning they can encode the same protein sequence. (3) Protein secondary structures can be inferred from protein sequences and used to construct a better alignment. For example hydrophobicity patterns and other molecular content information is used by PRALINEˆTM a transmembrane-aware protein aligner (Pirovano et al., 2008). For research that deals with non-coding regions of the genome it is often not meaningful to use the protein alphabet. This study focuses exclusively on diverse sequences of protein encoding genes. Thus, it is restricted to the use of protein sequences.

1.4 MSA Improvement Strategies

After an MSA is attained from an alignment software, it can be further improved by a number of existing methods. For phylogenetic analysis it is often important to trim off unrelated sequence regions from the MSA, such as overhanging ends or long gaps. TrimAl (Capella-Gutiérrez et al., 2009) is an example of an MSA trimming method, and Gblock (Castresana, 2000) is an MSA scoring method designed for MSA trimming. Other MSA refinement methods, such as RASCAL (Raymond et al., 2002) operate on MSAs by shifting residues within the alignments in order to improve poorly aligned regions. . A third type of MSA improvement strategies is MSA scoring. Alignment scoring approaches such as norMD (Thompson et al., 2001), GUIDANCE (Penn et al., 2010), and PSAR (Kim and Ma, 2011) can be used to remove badly aligned or unrelated sequences from MSAs. Once the sequences are removed, the columns that only consist of gaps are collapsed or the entire MSA is re-computed. The current project focuses only on the latter type of cleanup by identifying difficult to align sequences and then recomputing the MSA.

1.5 Application of MSA scoring

MSA assembly is required at early stages of a typical sequence families modeling analysis (Figure 1). The quality of the MSA effects all downstream stages of the analysis because all downstream stages rely on the MSA. Therefore, an MSA scoring method is applicable to sequence family modeling for MSA quality control and MSA quality improvement.

A particular example where MSAs scoring becomes necessary is the database search for sequence family members as it was done in the Cell Wall Navigator Database (CWN). The protein sequence database UniProt (UniProt Consortium,

2011) was used by CWN to update protein groups of interesting functional, structural, or evolutionary relationships.(Girke et al., 2004). In such a scenario, manual curation of the sequence families becomes infeasible due to the abundance of available protein sequences which is growing at an exponential rate. For example, UniProt in 2004 consisted of 1.2 million while the October 2011 version consists of 18 million protein sequences (UniProt Consortium, 2011).

1.6 Ways of introducing randomness into MSAs

In order to test MSA scoring methods, one can use manually curated MSAs of high-quality as correct dataset (known ground-truth) into which one can inject random residues or entire sequences to test the performance of scoring methods in identifying true and false positives. Introducing randomness into the curated MSAs provides a controlled experimental setup for testing MSA scoring methods. Several approaches exist for introducing randomness into MSAs. One way is to simulate sequences and add them to the MSAs. This simulation can be done with models that generate amino acid frequencies which are similar to real biological sequences (Durbin et al., 1999b). Another way is to randomly inject residues into the MSAs to purposely create mismatches in the columns of an MSA, as it was done for testing norMD by Thompson et al. (2001). In addition, methods exist to simulate DNA and protein sequence evolution. For example the Dawg DNA simulation program (Cartwright, 2005) was used to asses the performance of PSAR's MSA scoring method (Kim and Ma, 2011). Another approach is to take existing sequences in MSAs and mutagenize them to the desired level of randomness. Finally, real sequences can be randomly picked from large protein databases and added to an MSA. For most test cases the newly added sequences should not belong to the same sequence family as an MSA under study. This latter approach is the one that was used by this project. It was chosen here, because it best mimics the situation of most sequence family assembly, alignment and modeling routines (Figure 1).

1.7 Objectives

This project has two main objectives. The first one is to test the performance of two methods for scoring the membership of each sequence in an MSA. The scores can be used for identifying sequences that were false positive in the upstream or are too difficult to align to generate a meaningful MSA. The second objective is to develop an MSA cleanup method that utilizes the scores of both methods.

2 MSA Scoring Methods

2.1 SP-score with Reference

The sum of pairs score to reference (SPS) can be used to evaluate the MSA assembly methods when biologically correct reference MSAs are available. SPS is the count of how many residue pairs of the test MSA are aligned exactly as the corresponding pairs in the reference MSA.

In a test MSA of N sequences and M columns the ith column can be denoted as A_{i1}, A_{i2}, ..., A_{iN}. For each pair of residues A_{ij} and A_{ik}, define p_{ijk} such that $p_{ijk} = 1$ if residues A_{ij} and A_{jk} are aligned in the reference MSA, otherwise $p_{ijk} = 0$. The ith column score is:

$$S_i = \sum_{j=1, j \neq k}^{n} \sum_{k=1}^{n} p_{ijk} \tag{1}$$

The SPS score for the whole MSA is:

$$SPS = \sum_{i=1}^{M} S_i / \sum_{1}^{M_r} S_{ri} \tag{2}$$

Where M_r and S_{ri} are the number of columns and the column score S_i in the reference MSA (Thompson et al., 1999).

2.2 SP-score without Reference

The SP-score without a reference was introduced by Carrillo and Lipman (1988) and reintroduced for comparison to norMD by Thompson et al. (2001). For each pair of sequences in an MSA, a score is calculated based on the similarity between the sequences. The SP-score is then the sum of all the pairwise scores (Thompson et al., 2001). The pairwise scores can be the count of identical residues or can be the sum of residue similarities obtained through an amino-acid substitution matrix. In addition, the SP-score includes costs for insertions or deletions in MSA. Gaps can also be penalized according to a number of schemes. Thompson et al. (2001) compared norMD to the SP score which was calculated using the same amino-acid substitution matrix as used in norMD, Gonnet 250 (Benner et al., 1994). SP-score was computed with affine gap penalties (Altschul and Erickson, 1986) with manually set gap opening and a gap extension parameters. The SP-score was defined as:

$$SP = \sum_{i=1}^{M} \sum_{j=1}^{M} \sum_{l=1}^{M} C(A_{il}, A_{jl}) - (gn + hl) \tag{3}$$

where g and h are the manually set parameters (gap opening and a gap extension penalty). In the alignment of a pair of sequences i, j as subset from the MSA, n is the total number of gaps and l is the sum of the total lengths of the gaps among the sequence pair. It was shown that the highest SP-scores scores are obtained for MSAs with a large count of long sequences, irrespective of the quality of the MSA.(Thompson et al., 2001).

2.3 norMD

Normalized Mean Distance score (norMD) is an MSA scoring method with is based on MSA column statistics and ab inito MSA properties. The top level formula of norMD is:

$$norMD = \frac{MD - GAPCOST}{MaxMD * LQRID} \tag{4}$$

The MD component involves consulting an amino-acid substitution matrix for every pair of residues of a given MSA column. The scores of residues pairs are down-weighted by the percent identity (PCID) between the two sequences. The pairs that belong to sequences that have a higher PCID contribute less to the overall score. Then all the pair scores are added to form one column score. The column scores are then normalized by multiplying by the number of sequences that have a non-gap in that column. The resulting MD score is the sum of all the column scores.

The purpose of the MaxMD is to counteract the effect of MSAs with longer sequences getting higher MD values. MaxMD is the maximum attainable MD score given the sequences' length of the scored MSA. GAPCOST charges the MSA a penalty for opening a gap plus a gap length penalty. In the official norMD implementation the default GAPCOST is 0. The purpose of the lower quartile range of the pairwise hash score (LQRID) is to make the whole MSA score independent from how closely or distantly related are the members of the sequence group. LQRID avoids a bias towards a particular alignment method by using an alignment independent metric. To estimate the percent identity between sequences LQRID uses hash scores (Wilbur and Lipman, 1983) which calculated by extracting diagonals from dot-plots (Thompson et al., 2001).

2.4 Neighbor-joining tree assisted norMD sequence removal

A modified norMD integrated with the neighbor-joining tree building method (Saitou and Nei, 1987) was suggested to produce a method that removes badly aligned or unrelated sequences form MSAs. This method was introduced as an addition to the norMD MSA scoring method by Thompson et al. (2001). However, the use of this method is out of scope of this project because this method is unable to provide per-sequence scores (discussed in section 7).

2.5 GUIDANCE

GUIDe tree based AligNment ConfidencE (GUIDANCE) (Penn et al., 2010) measures and per-residue robustness of a progressive MSA aligner to guide tree perturbations when aligning protein groups.

1. GUIDANCE uses a progressive MSA software to construct an alignment. For this step it uses MAFFT (Katoh et al., 2002) and the resulting MSA is used as base-result for the downstream permutation steps. The guide tree used to construct the base-alignment is also extracted.

2. A bootstrapping approach (Felsenstein, 1985) is used to generate 100 perturbed versions of the guide tree.

3. The MSA is ran again on the original protein group but this time the 100 perturbed guide trees are used to generate 100 separate MSAs.

4. The 100 perturbed MSAs are compared to the base MSA to obtain the scores.

In order to estimate the robustness of each to guide tree perturbations GUIDANCE calculates the SP-score to a reference (Thompson et al., 1999) placing the perturbed MSAs as subjects and the base MSA as the reference. To get column scores GUIDANCE calculates a Sum-of-Pairs column score (SPC) which is equivalent to the score S_i for the ith column in Equation 1. The average SPC over all perturbed MSAs is the GUIDANCE CS column score. In addition a GUIDANCE residue pair score is calculated for each residue pair in the base MSA. One is assigned to a pair in a perturbed MSA when the corresponding pair is aligned in the base MSA, zero is assigned to all other pairs. The GUIDANCE residue pair score is the average score over all perturbed MSAs. In addition to per-column scores the software package provides MEAN_COL_SCORE and MEAN_RES_PAIR_SCORE for the entire MSA . Finally, the software package provide per-sequence scores (Privman, 2011a).

The GUIDANCE website provides a means for “removal of sequences that cause errors in the MSA because their alignment with the rest of the sequences is unreliable” (Privman, 2011b). It allows the user to choose a threshold bellow which a sequence will be removed but the website also notes that “There is no specific recommended value for this cutoff because its effect on the alignment varies considerably among datasets. The web server provides a list of cutoffs with their respective effects on the remaining proportion of sequences and users are encouraged to experiment with several cutoffs.” (Privman, 2011b)

2.6 Gblocks

Gblockes (Castresana, 2000) is designed to remove badly aligned and divergent regions in an MSA. It provides are score of 1 or 0 to every column on the alignment. The set of rules used to determine the score are very simple and are based on the number of exactly aligned residues, length of the aligned regions, and conservation of the flanking positions. The method does not use an amino-acid substitution matrix, instead looking only at the number of identical residues.

2.7 PredictedSP

The predicted sum-of-pairs (Ahola et al., 2008) is a statistical model which was trained on the SP-score-to-reference (Equation 1) measurements using the Homstrad database as reference. When evaluating the score the model only considers the conservation level ConsAA (Ahola et al., 2006), the percent identity, the number of sequences, and the length of the MSA. Based on this information, PredictedSP predicts the SP-score-to-reference. This method has been shown to outperform norMD (Ahola et al., 2008). I was not able to evaluate this method because the implementation was not available since September 2010 and I was not able to reach the author due to email change of the primary contact for Ahola et al. (2008). Recently the author made the implementation available.

2.8 PSAR

PSAR (Kim and Ma, 2011) is an MSA scoring method based on probabilistic sampling of suboptimal alignments. It obtains a number of suboptimal MSAs and compares them with the input MSA. The computation of the score against the input MSA is very similar to and inspired by GUIDANCE. In simulations of DNA MSAs the authors show that PSAR outperforms GUIDANCE. PSAR does not support protein sequence. Validation was done only on DNA. However, there seems to be no inherent obstacles for implementing protein sequence support.

2.9 Scoring Methods Summary Table

I indiscriminately studied and expanded a collection of literature about MSA scoring methods. Two MSA scoring methods needed be chosen for testing their capability to revise sequence families. I looked for several criteria to try to increase my chances of building an automatic classifier that can detect unrelated or poorly aligned sequences in MSAs: (1) Scoring should be done on protein sequences and utilize amino-acid substitution matrices in order increase sensitivity of the involved sequence family analysis routines as discussed in section 1.3; (2) Scoring should not require a reference ground-truth MSA; (3) Scoring should be independent of ab initio parameters of the MSA such as number or length of sequence; (4) Per-sequences scoring capability is preferable; (5) Scoring should either be trivial to implement or a working implementation should be available.

	Protein	**Sub**	**Per-MSA**	**Per-Seq**	**Size-Indep**	**Fast**
Reference SP	**Yes**	No	**Yes**	No*	Not Known	**Yes**
SP-score	**Yes**	**Yes**	**Yes**	No*	No	**Yes**
norMD	**Yes**	**Yes**	**Yes**	No*	**Yes**	**Yes**
NJ norMD seq. remover	**Yes**	**Yes**	No	No	**Yes**	**Yes**
Gblocks	**Yes**	No	No	No	**Yes**	**Yes**
GUIDANCE	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	No
PredictedSP	**Yes**	**Yes**	**Yes**	No*	**Yes**	**Yes**
PSAR	No	**Yes**	**Yes**	**Yes**	**Yes**	No

Table 1: *MSA Scoring methods comparison* **Protein** indicates the capability to score protein sequences; **Sub** indicates the capability to utilize a character substitution or similarity matrix for comparing residues; **Per-MSA** indicates the capability to produce a score for an entire MSA; **Per-Seq** indicates the capability to produce a score for each sequence in the MSA; **Size-Indep** indicates that the method is mostly independent of ab initio parameters of the MSA such as number or length of sequences; **Fast** are the tools that take one or more orders of magnitude less running time to commute the score than to it would take to re-generate the MSA that is being scored.

No* indicates that although the method cannot provide sequence-level (Per-Seq.) scores, the sequences-level scores can be inferred from the MSA-level scores (Per-MSA).

3 Methods

3.1 Experimental Strategy

By using expert curated MSAs, one can assume nearly no false positive sequences contained in these families as well as a very low number of badly aligned members. I am able to control the amount of randomness that I add to these MSAs by spiking in unrelated sequences. This will provide a setup with known member and non-member sequences that can be used to explore the ability of MSA scoring methods to discriminate between protein sequences belonging to a family. In addition, I will have control over the amount of added nonmembers and can use that to explore how the MSA scores change as a function of the amount of added noise.

Some MSA scoring methods can provide per-sequence scores and for others one can build ways of inferring per-sequence scores from the whole MSA scores. I build a binary linear classifier by choosing a cutoff among sequence scores. By looking at all possible cutoffs I can plot a Receiver Operating Characteristic (ROC) for the classifier and use the area under the curve (AUC) to directly compare classifiers. I have one classifier for every MSA scoring method. In addition, I analyze combined classification approaches.

An automatic cleanup of a sequence group can be performed by scoring the group's MSA and removing sequences which bring the score bellow a threshold. This project explores and analyzes methods for selecting optimum thresholds.

3.2 Member and Nonmember Sequences

Sequence family modeling aims to determine the membership of sequences to a functional, structural, or evolutionary protein family. For many reasons sequences can be placed into protein groups incorrectly, for example due to a false positive identified by a sequence similarity search against protein databases. When badly aligned sequences are present in an MSA it may not be possible to determine without external knowledge if the sequence is related to the group. On the other had, a correct alignment of the sequence can provide evidence for relatedness to a sequences family.

I will control and track sequences that are unrelated to protein groups. A sequence is termed *nonmember sequence* when it is unrelated to the protein group. All other sequences in the protein groups will originate from curated datasets of protein families, I term those sequences as *member sequences.* When evaluating the performance of the automatic classifier sequences will be called *true positives*, *false positives*, *true negatives*, or *false negatives.* Sequences whose scores fall bellow the selected threshold will be classified as predicted member sequences. If the classification is correct then those sequences will be counted as true positives. Nonmembers incorrectly classified as member sequences will be counted as false positives.

3.3 Separation of Sequence-level MSA scores

To better understand the behavior of scoring methods, I look at score density plots of member and nonmember sequences. Removal of badly aligned or nonmember sequences was one of the design goals of MSA scoring methods that I investigate. It is expected that member sequences will be enriched on one side of the density plot and the nonmember sequences on the other side. However, I do not expect a clear separation between the two because many diverse protein families show a continuum of detectable to non-detectable similarities within the evolutionary sequence space. (Rost, 1999)

Nevertheless, for some test cases it can be informative to test the MSA scoring methods not only on very diverse families, but also on more homogeneous families where all sequence members share relatively high similarity. For a more homogeneous dataset of member sequences, one can usually expect a clearer separation of the member and nonmember sequences in the score density plots. For the latter case I used sequence subfamilies where all members shared at least 50% identity. Then I add nonmember sequences in the same way as for the diverse families (please refer to the Methods section 3.4).

3.4 Spiking: Adding Nonmembers to Sequence Families

To test the different MSA scoring methods, I developed an automatic procedure to add to the curated MSAs from CDD (member sequences) controlled numbers of nonmember and randomly selected sequences from UniProt. The number of member sequences will vary from 19 to 200 per MSA. To be independent of the number of sequences in each MSA, I measure the amount of nonmember sequences to be added in percent. To also prevent the nonmember sequences from creating unaligned overhangs, I spike-in sequences of the same length as observed in the original MSA. This poses some restrictions on how I can control the nonmember spike-in amounts. Since the smallest number of sequences in our MSAs is 19, a convenient increment to use is 1 sequence or 5% in residues. I will aim to add 5%, 10%, 15%, 20%, and 25% nonmember ratios. The nonmember ratio is bound to be less then the targeted ratio within 5% residues. Every time a set of nonmember sequences is added I run an MSA program on the combined protein group to generate a new MSA. In the following these modified MSAs are called "Spiked MSAs".

3.5 Datasets

As member sequences (please refer to section 3.2), I used sequences from MSAs of the Conserved Domain Database (CDD) (Marchler-Bauer et al., 2011). This databases contains 20,196 MSAs in the August 2011 release (version v2.31). I did preprocessing as described at the end of this section. Preprocessing reduced the database to 18,486 MSAs. I randomly sampled these 18.4 thousand MSAs to obtain 3 samples of 100 MSAs. I will refer to these samples as *Sample 1*, *Sample 2*, *Sample 3*. Only the assignments of sequences to families was used from CDD, but not the MSAs.

As nonmember sequences, I used sequences from UniProt (UniProt Consortium, 2011). This databases contains 16,504,022 proteins in version 2011-08. I did per-processing as later described in this section, which reduced the database to 12,802,581 protein sequences. I developed and used a routine that scans these 12.8 million sequences to obtain random samples for a given number and length of sequences (please refer to section 3.4). I apply this routine to obtain five different amounts of nonmember sequences for each sequence family in Sample 1, Sample 2, and Sample 3. Amounts of 5%, 10%, 15%, 20%, 25% of nonmember sequences were added to the three random samples.

Both databases CDD and UniProt consist of real biological data. UniProt is the largest publicly available collection consisting of two sections: UniProtKB/Swiss-Prot and UniProtKB/TrEMBL. The first section set of sequences that are manually-annotated from literature. UniProtKB/TrEMBL is a set of computationally analyzed sequences. CDD is likely the largest collection (Shameer, 2011) of MSAs from various databases that are focused on structural and functional sequence family modeling. CDD consisting of the following databases: NCBI CDD curation effort (Marchler-Bauer et al., 2011), SMART (Letunic et al., 2009), PFAM (Finn et al., 2010), COGs (Tatusov et al., 2003), PRK (Klimke et al., 2009), TIGRFAM (Haft et al., 2003).

Alignment software has a limit on the number and length of sequences that can be aligned within a few hours of an up to date CPU. In order to process samples within a reasonable amount of time I filtered out all CDD entires that had sequences with more than 1000 residues and alignments with more than 200 sequences or more than 100,000 residues total.

CDD has non-unique sequence identifiers in protein groups and this poses a problem for some downstream software (e.g. BLAST). I make the identifiers unique without loosing information by adding unique prefixes to the CDD identifiers.

From Uniport I needed to be able to randomly sample sequences that can be added into existing protein groups without altering the average sequence length of the protein group. In order to perform such additions I split Uniport into files each containing sequences of a given length. I was able to have at least 1000 proteins for any given length between 9 residues and 999. Length of 8 and 1000 is was for 841 and 999 proteins respectively. All lengths from 8 to 1000 were included in the dataset.

3.6 Scoring with GUIDANCE

The protein family scoring method GUIDANCE was used to score each of the spiked MSAs. Default parameters of GUIDANCE were used. Since GUIDANCE provides per-sequence scores, all I need for classification was run GUIDANCE on Samples 1, 2, and 3 (please refer to section 3.5), parse the sequence-level score values from the GUIDANCE results files, and choose a threshold (please refer to section 3.11) for dividing the sequence-level scores into member and nonmember classes (please refer to section 3.2).

3.7 Scoring with norMD

To obtain MSAs, I align the spiked protein groups with MAFFT, using the same parameters as used by GUIDANCE. The norMD scoring method provides MSA-level scores. Additional processing steps are required to extract its sequence-level scores. For this, I run norMD on each MSA generated by MAFFT. If N is the number of sequences in an MSA, then I have one run for the whole MSA and N runs for the MSA, leaving one sequence out. I use only the Modified-norMD score

(Equation 8) in which it is obtained from the verbose mode of the norMD as I found that the modified variant is more sensitive to nonmember sequences (please see the Results section). I obtain the sequence-level scores by subtracting the MSA-level score from the score with the sequences left out. Then I choose a threshold (please refer to section 3.11) for dividing the sequence-level scores into member and nonmember classes (please refer to section 3.2). An alternative approach is discussed in section 7.

3.8 Scoring with BLAST

MSAs capture more information about the evolution of the aligned proteins then pairwise alignments. However, pairwise sequence alignment should be usable for detecting unreliable or badly aligned sequences within a protein group. For this study, I have developed and tested the following MSA cleanup method based on BLAST (Altschul et al., 1990).

In each group the protein sequences are aligned in an all-against-all fashion using the *blastp* mode of the *blastall* executable. Only the hits with an E-value below 0.01 are recorded. The final score of a sequence is the number of pairwise hits below this threshold to the other sequences in a family. After the sequence-level scores were obtained, I perform classification of sequences by choosing a threshold (please refer to section 3.11) for dividing the sequence-level scores into the member and nonmember classes (please refer to section 3.2).

3.9 Classifier Evaluation

Having the MSA scoring methods, I build a simple classifier that will predict which sequences are members and which are nonmembers based on the MSA scores. Sequences with a score bellow a specific threshold T are predicted to be nonmembers (to be removed) and all other sequences are predicted as members (to be kept). The performance of the classifier can be tested because the true memberships of all sequences is known.

More than one classifier is build in this way because there are several scoring methods and several ways to do combined classification. Please refer to the Combined Classification section in Methods. Every classifier that I build is evaluated on a random sample of MSAs from CDD with various amounts of nonmember sequences added from UniProt. For details, please consult the "Datasets" section. Each evaluation is visualized in the form of Receiver operating characteristic (ROC) curves.

To generate a ROC curve, all thresholds that generate unique predictions must be considered. Given N scores in sorted order I need to evaluate up to $N + 1$ thresholds: one threshold between every two adjacent scores and two threshold on the extremes. The number of thresholds that I need to consider may be less than $N + 1$ only if some scores are not unique.

In order to compare the performances of classifiers at different sensitivity / specificity levels, I superimpose ROC curves. This representation makes performance comparisons among different methods possible because in the ROC curves the thresholds for different classifier evaluations become mapped into a normalized 0 to 1 range which is independent of the particular scores and thresholds. Any point represents a True Positive Rate and a True Negative Rate. The overall performance of each method can be expressed by the area under the ROC curve (AUC). I use the AUC to compare the classifiers in an automated way.

3.10 Combined Classification

The scoring methods norMD and GUIDANCE measure different aspects of MSAs. norMD uses conservation in MSA columns and *ab initio* information such as the total number of sequences and the length of the longest sequence. GUIDANCE measure the resistance of the alignment to controlled perturbations of the progressive alignment guide tree. This difference lends itself to combined a classification approach.

One way of combining GUIDANCE and norMD scores, is to add the two scores together. For example a sequence may have a low GUIDANCE score and a high norMD but the sum will be higher than the cleanup threshold and the sequence will not be removed. This way of combining the score can be better understood when each sequence is plotted as a point in two-dimensional space. Any partitioning of this space in two parts would be a binary classification. Items falling into one partition are classified as nonmembers (e.g. remove sequences), whereas items in the other partition are classified as members (e.g. keep the sequences). In this project I resort to one of the simplest types of such partitioning in which the space is divided by a straight line. Without combined classification the classifier would divide our two-dimensional space in half by a horizontal or a vertical line (depending on which scoring method is used). When using combined classification by adding the two scores the lines is a diagonal (please see Figure 12 for an example). I use only 3 tilts of the dividing line: horizontal, vertical, and diagonal. The threshold that I use will effect the positioning of the dividing line.

3.11 Threshold Selection

In the methods that I surveyed on removing nonmember or badly aligned sequences from MSAs (please refer to section 2) no automatic methods were described for selecting thresholds. Nevertheless, automatic score threshold selection is a necessary component for automatic MSA cleanup.

3.11.1 Limits of Static Threshold Selection

It is often an arbitrary decision to settle on one static threshold for all MSAs because the existing MSA scoring methods will assign scores to sequences with a bias to the overall quality of the MSA and a bias to the amount of nonmember or badly aligned sequences in the alignment.

I explore the dependency of static thresholds on the variance of the amounts of nonmember sequences in MSAs. In addition, I test the performance of static thresholds at set amounts of nonmember sequences. I have control over the amount of nonmember sequences (please refer to the Methods section 3.4). I regenerate the MSAs of protein groups after adding the nonmember sequences. As I have such control, it becomes possible to select and analyze thresholds for various amounts of nonmember sequences added. To evaluate the results, I plot the obtained thresholds against the added amounts of nonmember sequences. To explore the performance of static thresholds I train the thresholds (described later in this section) on one random sample of a fixed amount of nonmembers. Then I apply those thresholds to another random sample with the same amount of nonmember.

I look for a reasonable way to select a static threshold in a random sample for a known added amount of sequences using the known member and nonmember information. Even when the ground-truth is known, a sequence removal threshold on MSA scores is a compromise between the true positive rate (TPR) and the true negative rate (TNR). The only clear situation is when the MSA scoring method has done a perfect job at separating members and nonmembers - in that case a threshold exists where all members will be retained and nonmembers will be removed (TPR=1 and TNR=1). However in most cases there is a compromise: moving the threshold down will increase TPR while decreasing TNR. Moving the threshold up will decrease TPR while increasing TNR. I do not know in advance if a high TPR or a high TNR is preferential, so I would like to settle on a threshold that is a good balance of TRP and TNR.

A measure that may be able to strike such a balance is the Matthews correlation coefficient:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{5}$$

All MSAs of the sample are placed into one pool as a flat list of sequences. Each sequence of the MSA will have a numeric score. I create one pool of scored sequences per each amount of added nonmembers. Then I test all unique thresholds to find one threshold where the MCC has its maximum. For example, a particular pool will have 100 MSAs consisting of about 6,000 sequences and 5% in residue counts will be nonmember sequences. I then analyze the TPR and TNR rates that this threshold gives when applied to other samples.

3.11.2 Dynamic Threshold Selection

In the previous section, for MSA clean-up I used the MCC coefficient (Equation 5) for making automated decisions on the static thresholds. One threshold was selected for the entire pool of MSAs. However, if an MSA has an overall low quality then all sequences appear badly aligned. This poses a problem to objective scoring of sequences in MSAs. The quality of the MSA changes proportionally to the amount of nonmember sequences in the alignment (Figures 3 and 4). However, when MSA cleanup needs to be performed, it is not known how much nonmember or badly aligned sequences are present. Consequently, sequences that are related to the protein group will have a score that is dependent on the overall quality of the MSA. This dependency is a problem for threshold selection because at a set threshold sequences may be removed or kept for reasons other then their true membership to the group. This challenge calls for a threshold that is adjusted to the specific properties of an MSA. I term this approach *dynamic threshold selection.* A dynamic threshold selector will make an automated decision without the knowledge about the sequences and it will utilize the information of individual MSAs.

3.11.3 Limits of Dynamic Threshold Selection

Before attempting to develop a method that can select dynamic thresholds *ab initio*, I test what are the limits of dynamic thresholding given that I know which sequences are nonmember. I have the scores of sequences in a given MSA and I know which of these sequences are members and nonmembers. I then calculate the MCC coefficient for every possible outcome of partitioning the sequences via linear binary classification. The threshold that produces the highest MCC value is then chosen.

To evaluate the performance of classification based on this threshold method, I compare the TRP and TNR rates to other classification methods. I normalize all sequences scores using the thresholds to make further inquiries and comparisons such as (1) ROC curves / AUC values, (2) score separation density plots, and (3) placement of sequences in a two-dimensional plane of different scoring methods.

In case of the *Perfect-Balance* method, the information on the nonmember sequences was used to select the threshold where the largest MCC value (Equation 5) is observed. The scores were then normalized for each MSA on the threshold T using the R Code 2.

```
min1 <- min(scores)
scores <- scores - min(scores)

max1 <- max(scores)
scores <- scores / max(scores)

T <- T - min1
T <- T / max1

scores[scores > T]  <- 1 - (1 - scores[scores > T]) / (2 - 2 * T)
scores[scores <= T] <- scores[scores <= T] / (2 * T)
```

Code 2: *Scaling scores around a threshold T.* This code normalizes the scores to a range of 0-1. Two bifurcations are produced by using T to separated the scores.

3.11.4 Dynamic Threshold Selection by Min/Max Scaling

```
scores <- scores - min(scores)
scores <- scores / max(scores)
predicted_nonmember_sequences <- which(scores <= 0.5)
predicted_member_sequences <- which(scores > 0.5)
```

Code 3: *MSA sequence score scaling* A simplistic dynamic threshold selection procedure. The R code vector "scores" is scaled to a 0-1 range and classified into member and nonmember sequences.

$$T = \min(scores) + \frac{\max(scores) - \min(scores)}{2} \tag{6}$$

I perform dynamic threshold selection in order to reduce the dependency of the scores to the MSA-level quality (please refer to the discussion in section 3.11.1). A method that can potentially achieve this effect is to linearly scale the sequence scores against their minimum and maximum so that they are in a 0-1 range. I use Code 3 in my analysis. This code sales the scores. After scaling the scores, the threshold 0.5 is equivalent to the threshold defined by Equation 6 for unscaled scores. I chose to scale all the scores over using a single threshold T because scaling made it possible to compare scores of different sequence families. At this point the threshold selection has been completed. The inferred threshold has now been scaled to the new absolute universal value of 0.5 which can be used statically in downstream analysis. A z-score normalization would be another possible approach for the task at hand. By looking at ROC performance of the these scaled scores I evaluate the improvement obtained from this dynamic threshold selection procedure. This method will produce false negatives when there are no nonmember sequences in the dataset because the scaling formula will always place at least one sequence bellow the threshold.

3.11.5 Dynamic Threshold Selection by Balancing MCC

The scaling method in the previous section only looks at minimum and maximum scores to select a threshold. A different method that looks at all sequence scores is tested. This method contrasts the GUIDANCE and norMD scoring methods. The methods operates similar to my static threshold selection approach but is not ground-truth aware.

The method is a greedy algorithm which iterates over all possible thresholds for the two scoring methods. A total of N^2 thresholds are checked, where N is the numbers of sequences. For our dataset N^2 does not exceed 200^2. Let the set of sequence scores of the first scoring method be called *Subject Scores* and the set of scores of the second method the *Base Scores.* A threshold in the Base set is taken as the known ground-truth for the MCC calculation (Equation 5), a threshold in the

Subject set is taken as the prediction of member and nonmembers for TP, TN, FP, FN counts. The results are two thresholds that lead to the maximum MCC among all N^2 thresholds.

Algorithm 1 Dynamic Threshold Selection by Balancing the MCC

```
scuts ← all the unique thresholds of the Subject scores set
bcuts ← all the unique thresholds of the Base scores set
maxmcc ← $-1$
maxmcc_scut ← nothing yet
maxmcc_bcut ← nothing yet
for bc in bcutts do
    for sc in scuts do
        predicted_member ← all Subject sequences with scores > sc
        predicted_nonmember ← all Subject sequences with scores < sc
        tp ← number of sequences in predicted_member with scores > bc
        tn ← number of sequences in predicted_nonmember with scores < bc
        fp ← number of sequences in predicted_member with scores < bc
        fn ← number of sequences in predicted_nonmember with scores > bc
        mcc ←
            $\frac{(\mathrm{tp} \times \mathrm{tn}) - (\mathrm{fp} \times \mathrm{fn})}{\sqrt{(\mathrm{tp}+\mathrm{fp})(\mathrm{tp}+\mathrm{fn})(\mathrm{tn}+\mathrm{fp})(\mathrm{tn}+\mathrm{fn})}}$    (7)
        if mcc >= maxmcc then
            maxmcc ← mcc
            maxmcc_scut ← sc
            maxmcc_bcut ← bc
        end if
    end for
end for
return maxmcc_scut, maxmcc_bcut
```

The method iterates over all possible thresholds among the two sets of score values produced by the two scoring methods. I arbitrary pick one of the two sets of scores and define it as *Subject Scores*. The remaining set of scores gets defined as *Base Scores* which I temporally assume as the ground-truth within each iteration over a specific threshold. Having a pair of threshold and a temporally assumed ground-truth, I can calculate the resulting TP, TN, FP, FN counts for the Subject Scores. Using the TP, TN, FP, FN counts I compute the MCC correlation (Equation 5). Two thresholds that lead to the maximum MCC among all pairs of thresholds is the answer returned by the algorithm (please see Algorithm 1). The choice of which set to define as subject scores is irrelevant because

The running time of this algorithm is bounded because for our dataset the combination of all pairs of thresholds does not exceed $40,000$ (N^2 for $N = 200$).

In my test experiments, I used the above algorithm with the following two additions: (1) MCC will normally produce values from -1 to 1, but there may be cases with zero in the denominator and negative numbers under the square root. I handle these cases as follows: If FP=0 and FN=0 then I take 2 as the value. If TP=0 or FP=0 or the product under the square root is negative in MCC then -1 is used. (2) To handle that fact that there may be more than one maximum MCC value, in the extension to the above algorithm all co-optimal solutions are captured. The ties are resolved by checking which threshold among several options has the largest distance to the nearest sequence score. The final solution is the threshold with the largest distance among the co-optimal solutions.

4 Results and Discussion

4.1 Scores of Entire MSAs

When adding increasing amounts of nonmember sequences to the MSAs the corresponding GUIDANCE and norMD scores change proportionally. Figures 3 and 4 show the scores for entire MSAs for GUIDANCE and norMD, respectively, using a sample set of 100 MSAs from CDD with various amounts of nonmember sequences from UniProt.

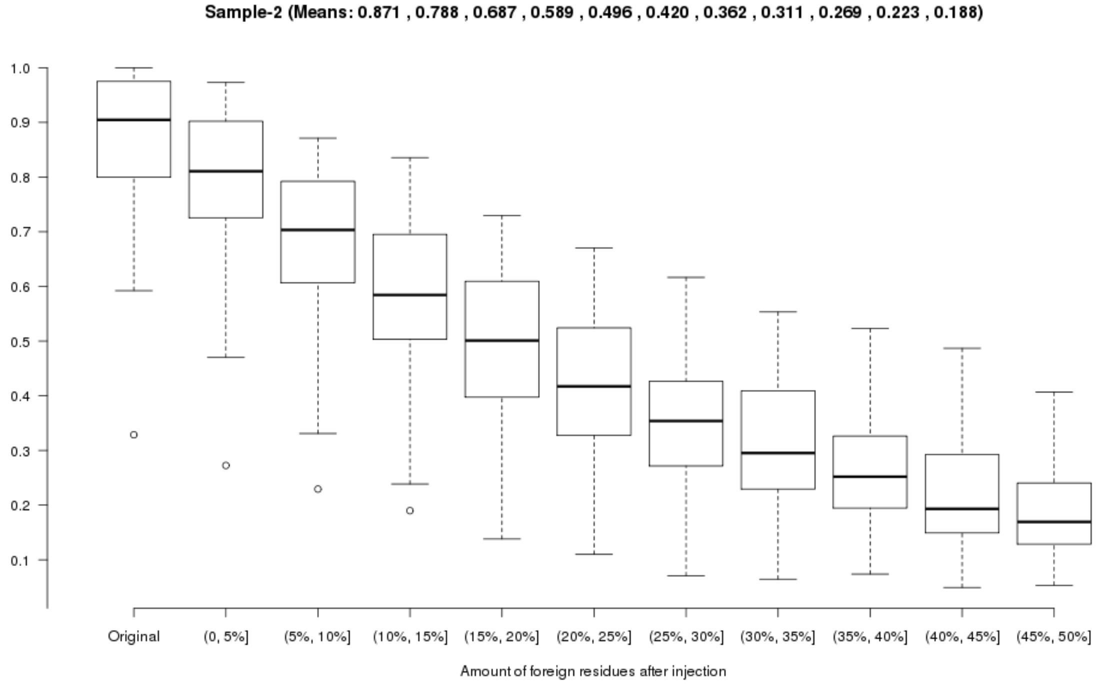


Figure 3: *GUIDANCE MSA-level scores* MSAs were scored with GUIDANCE using Sample 2 of size 100 (please refer to section 3.5). Scoring was performed with various amounts of nonmember sequences from UniProt added. In addition, the CDD protein groups were scored without adding. Box plots of raw MSA scores are shown.

A monotonic decline can be seen in the scores as I increase the amount of nonmembers. Another conclusion of this result is that it is not straight forward to derive a formula that would estimate the amount of nonmember sequences from the MSA-level scores because the decline of scores has a smaller slope at higher amounts of nonmember sequences.

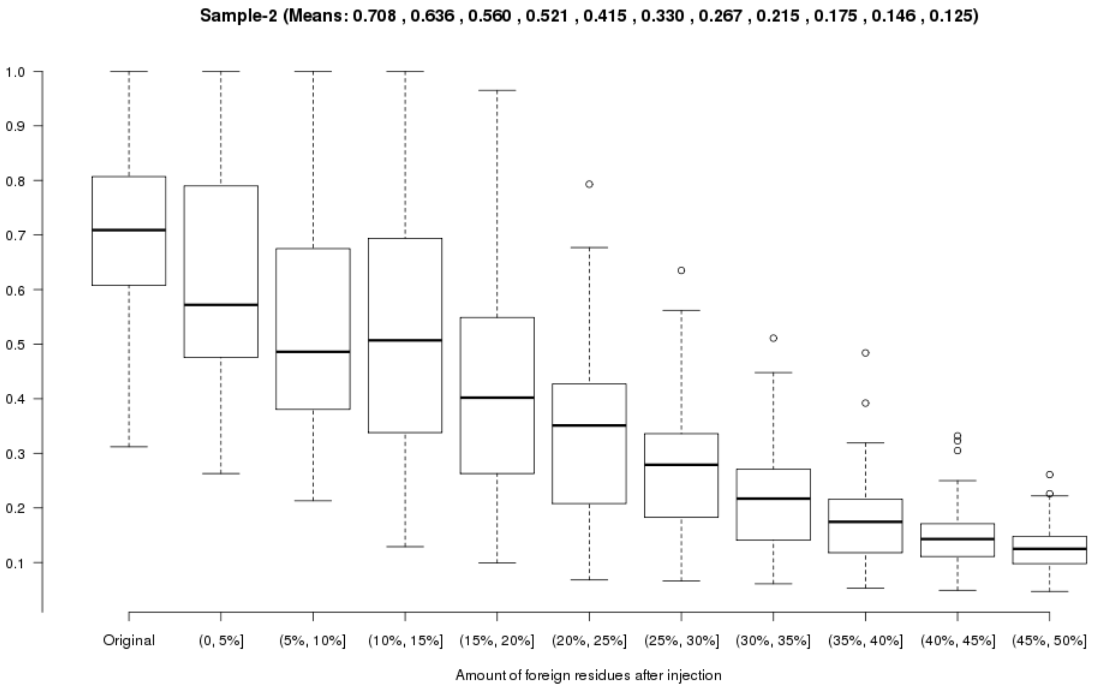


Figure 4: *norMD whole MSA scores.* Alignments were scored with norMD. Box plots of raw MSA scores are shown. Details are given in the caption of Figure 3.

For norMD the pattern is less regular than the one for GUIDANCE. The slight uptick of norMD's MSA-level scores at 10% nonmembers can be clearly seen in the 3 samples of the experiment. Please refer to the supplements section. A more regular decline in GUIDANCE is most likely related to the difference of the features that norMD and GUIDANCE are measuring. norMD is sensitive to changes in column conservation and MSAs parameters such as number of sequence. On the other hand, GUIDANCE measures a single high-level feature, robustness of the alignment to guide tree perturbation.

4.2 Score Separation

To asses the ability of the scoring methods to separate member and nonmember sequences, I plot the distribution of unnormalized scores (Figure 5). The member and nonmember assignments of sequences are defined in the Methods section 3.2.

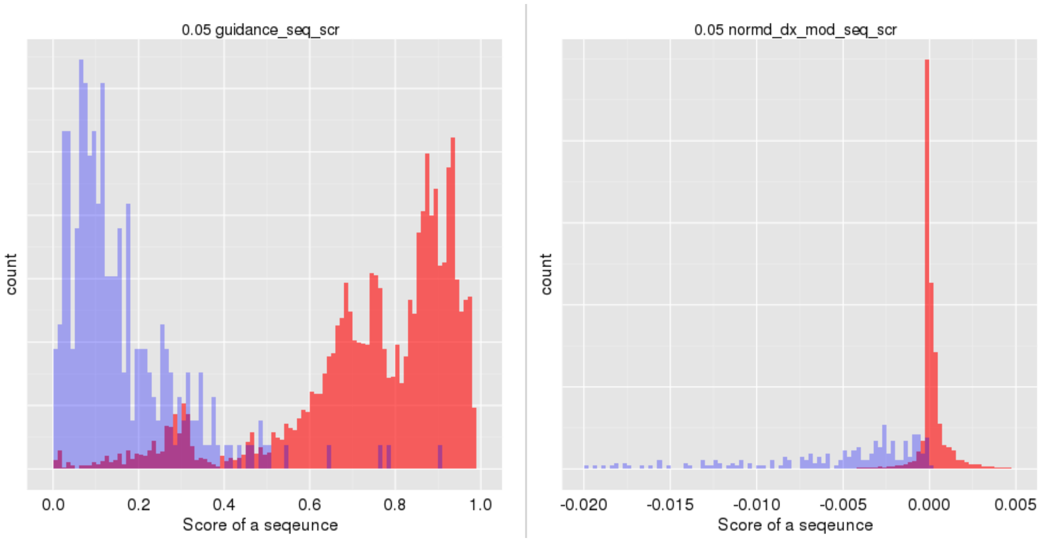


Figure 5: *Separation of sequence-level scores.* The y-axis shows the number of sequences scored within a bin of width 0.01 for GUIDANCE and 0.00025 for norMD. Blue bars represent the results of MSAs with 5% of nonmembers added. For visibility reasons, the height of nonmembers was multiplied making nonmember and member areas equal. The long tail of norMD extends beyond the plot.

The concentration of norMD scores around 0 is related to the way the sequence-level scores are derived from the MSA-level scores, as described in section 3.4. A sequence score of 0 means that the presence of a sequence in an MSA does not result in a change of the MSA-level score. Nonmember sequences mostly fall on the left of the distribution because their presence correlates with the decline of the MSA-level score. However, in large MSAs the presence of one nonmember sequence will result in a minor change of the whole MSA score, thus the placement will be close to zero on the distribution plot.

4.3 Score Separation on Families with Highly Related Members

To evaluate the score separation on MSAs where all sequence members share relatively high similarity, I subset the existing samples by taking only those sequences that have more than 50% identity among them. Then I add nonmember sequences and re-generate the MSAs (Please see Section 3.3).

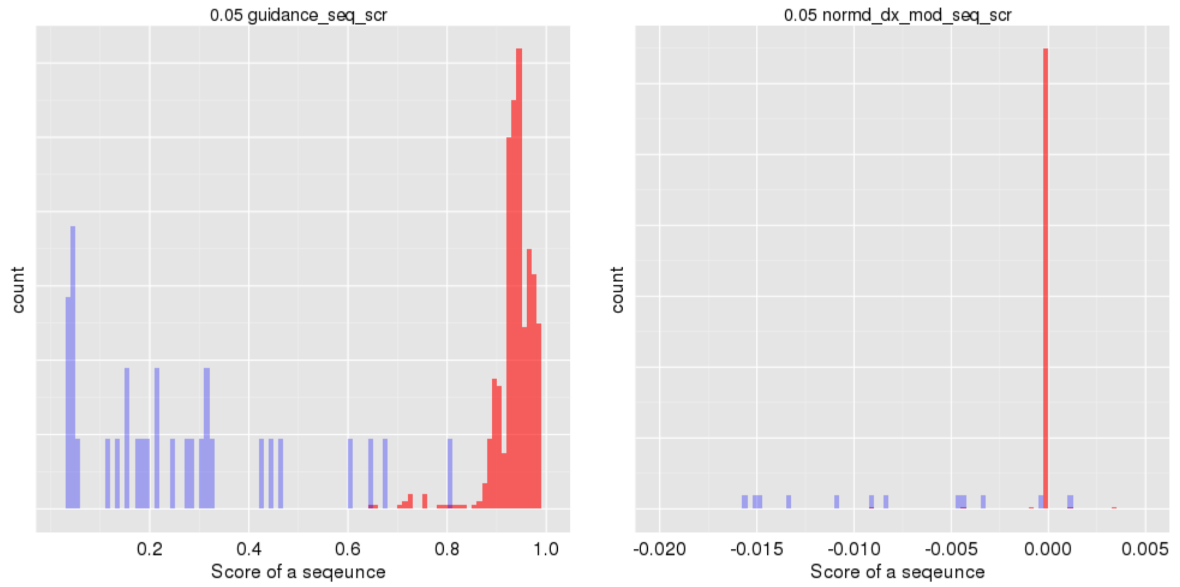


Figure 6: *Separation of norMD scores of 50% identity sequences Sample 2 subset.* Details are given in the legend of Figure 5.

When compared to the scores separation without the 50% identity subsetting (Figure 5), both MSA scoring method produce a more tightly grouped set of scores for member sequences (Figure 6). This is consistent with the characterization of the twilight zone of protein sequence alignments where it was shown that pairs of protein sequences above 30% identity produce alignments that unambiguously detect protein homology (Rost, 1999).

The overall effect of the 50% identity subsetting results in a better separation between the scores than MSAs with more diverse sequences. From this observation it is evident that liner classifier for MSA cleanup by sequence removal would perform significantly better if more than 50% identity was present amount all sequences in the MSA (Figure 7).

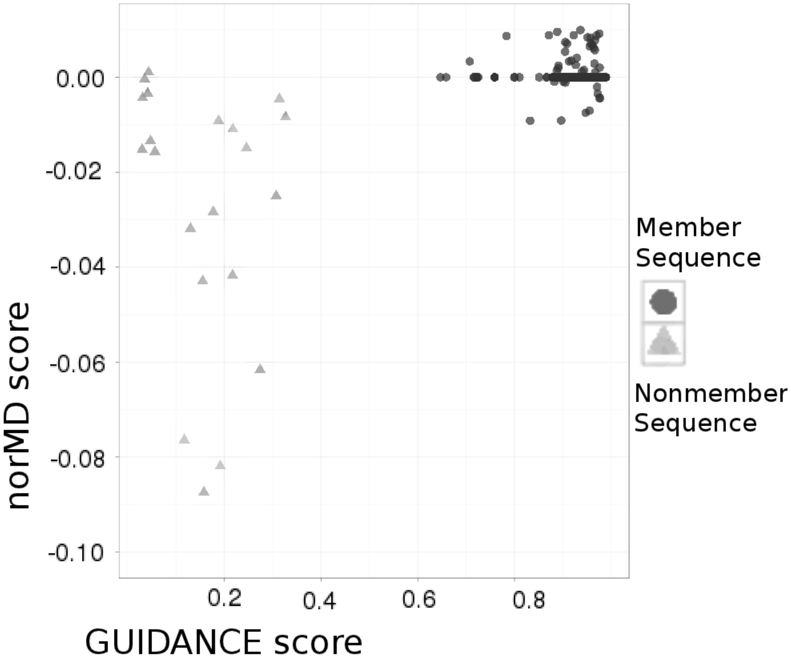


Figure 7: *GUIDANCE and norMD sequence-level scores of MSAs with highly related member sequences and 5% nonmembers.* In sample 2 (please refer to section 3.5) when highly related member sequences are retained in the MSA, the separation of member and nonmember sequences is greater than the separation observed for complete groups of member sequences (please refer to Figure 6).

4.4 Static Threshold Selection

Amount of nonmembers	**GUIDANCE**	**norMD**	**BLAST**
5 %	0.399	-0.000571	4.5
10 %	0.224	-0.000395	8.5
15 %	0.357	-0.000184	6.5
20 %	0.184	-4.99e-05	8.5
25 %	0.145	-1.75e-05	10.5

Table 4: *Static Thresholds* Threshold values for sample 1 (section 3.5) are plotted. "Amount of nonmembers" is the added amount of nonmember sequences. Listed are the thresholds that were obtained from training on the GUIDANCE, norMD, and BLAST sequence-level scoring methods were used as described in sections 3.6, 3.7, and 3.8 respectively. The method used for static threshold selection is described in section 3.11.1.

To explore the performance of static thresholds selection to varying amounts of nonmember sequences in MSAs I used a ground-truth aware method for selecting thresholds (Table 4). I placed all sequences of one sample into one pool and calculated MCC for all unique thresholds (Please refer to section 3.11.1). For all three methods the thresholds has a consistent upward or downward trend along the increasing added amount of nonmember sequences (Figure 8).

Thresholds that result in a maximum MCC in sample 1 are applied to sample 2 (please refer to section 3.5) and the true positive / negative rates are evaluated for correctness of sequence membership identification. It is observed that true positive / negative rates remain similar across samples. In Sample 1, the highest number of MSAs in the top 1% TPR and top 1% TNR was observed for the BLAST methods.

The MCC based method chooses threshold values that increase for norMD at larger amounts of nonmember sequences, but for GUIDANCE the threshold decreases at larger amounts of nonmembers (Table 4). From this result one can conclude that the optimum separation threshold of scores for related and nonmember sequences moves for norMD toward higher scores at larger amounts of nonmembers while for GUIDANCE the optimum separation value shows a trend toward lower scores. This conclusion is also supported by the score separation density plots (Figure 5) and the box plots for MSA-level scores (Figure 3). This explains why at a given static threshold the count of false positives will increase for norMD for larger amounts of nonmembers but it shows the opposite trend for GUIDANCE where the count of false negatives increases at larger amounts of added nonmembers (Figure 9).

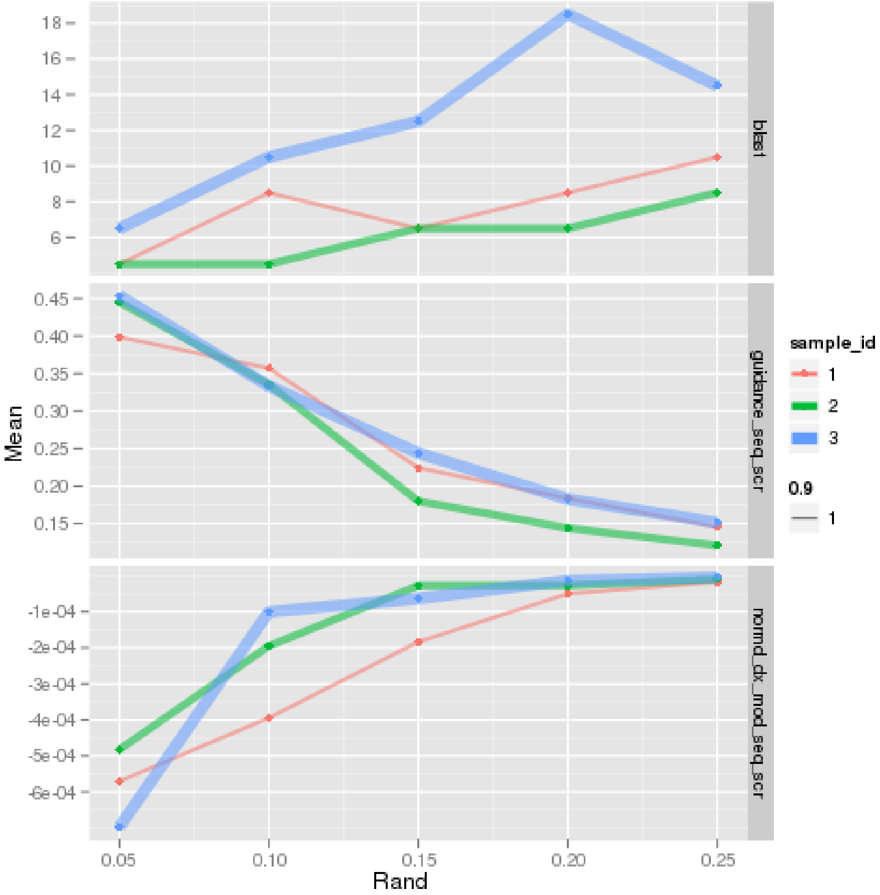


Figure 8: *Static Thresholds* Added amounts of nonmember sequences are show in percent on the horizontal axis. Threshold values for sample 1, 2, and 3 are shown on the vertical axis. The color and width of the lines indicates which sample the value was obtained from. Details are given in legend of Table 4.

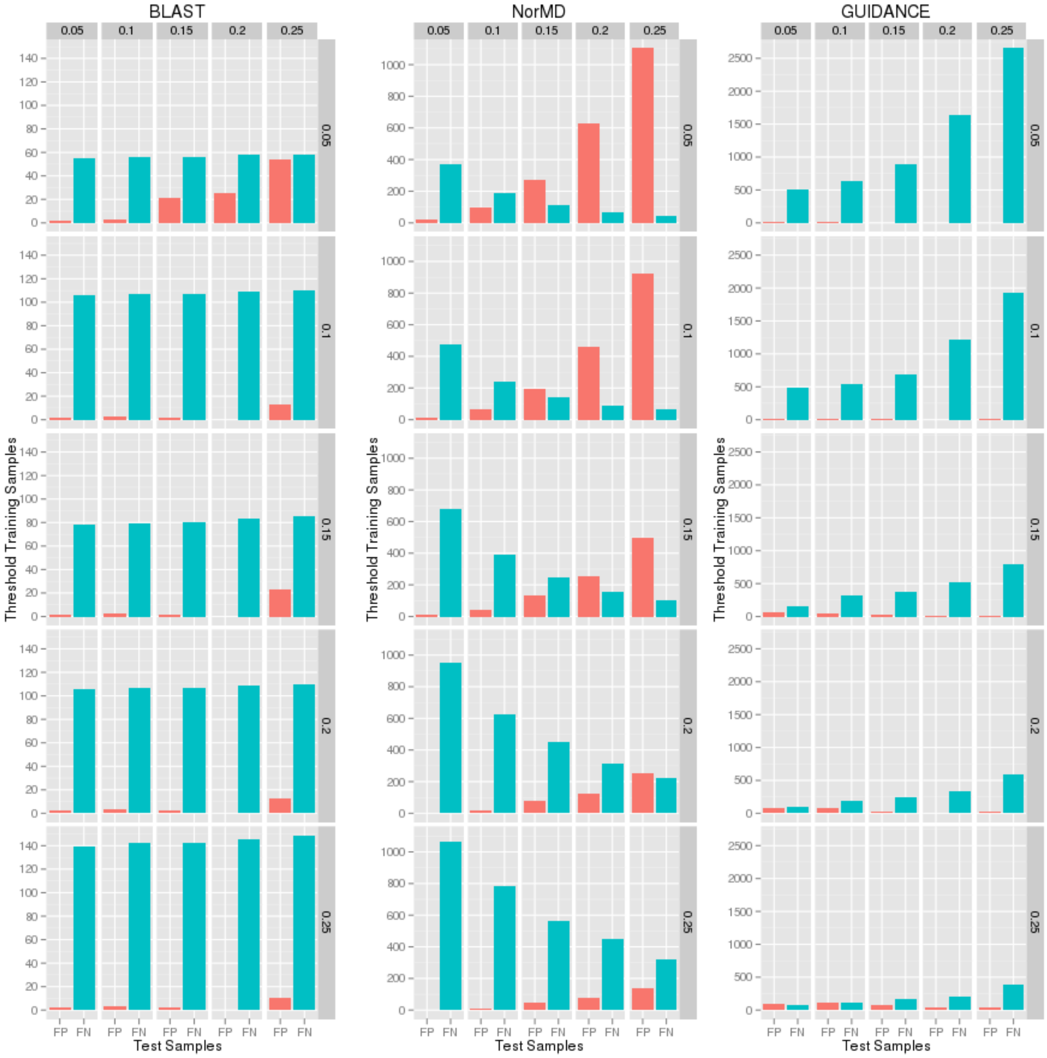


Figure 9: *Counts of mistakes in static MCC threshold selection* Static thresholds from Table 4 are applied to different amounts of added nonmember sequences in sample 2 (please refer to section 3.5). Red columns (labeled FP) show counts of false negatives. Blue columns (labeled FN) show counts of false positives. The outer horizontal axis shows three sequence-level scoring methods BLAST, norMD, and GUIDANCE. The outer vertical axis (labels on the right of each bar plot) the nonmember amount used to lookup the threshold in Table 4. The inner horizontal axises (labels on top) show the nonmember amount in the test random samples. The values on the horizontal axis are counts of mistakes (FP or FN counts).

4.5 Dynamic Threshold by Calibration of Two Scoring Methods

When studying the behavior of scoring methods, I sorted by value the results of the norMD (section 3.7) and GUIDANCE (section 3.6) sequence-level scores for visual inspection. I generated one list for each sample (please refer to section 3.5). I organized the lists by the scoring method and the amount of added nonmembers (please refer to section 3.2). By having the member and nonmember sequences labeled alongside the sorted lists of score values it was clear from visual inspection of the lists that, with few mistakes, the scoring methods are able to separate the member and nonmember sequences. This ability is also evident from ROC (e.g. Figure 14) and density plots (e.g. Figure 5).

However, the point where the separation occurred was different for different sequence families. To find the point of separation automatically, I developed several approaches. One of the approaches was a heuristic that can select thresholds without the knowledge of the ground-truth. It uses norMD and GUIDANCE and finds two thresholds that provide the highest MCC between the two sets of scores (please refer to section 3.11.5). I evaluated the performance of my heuristic and observed that it makes less mistakes than the ground-truth aware static threshold selection (section 3.11.1) as shown in Figure 10 and Figure 11.

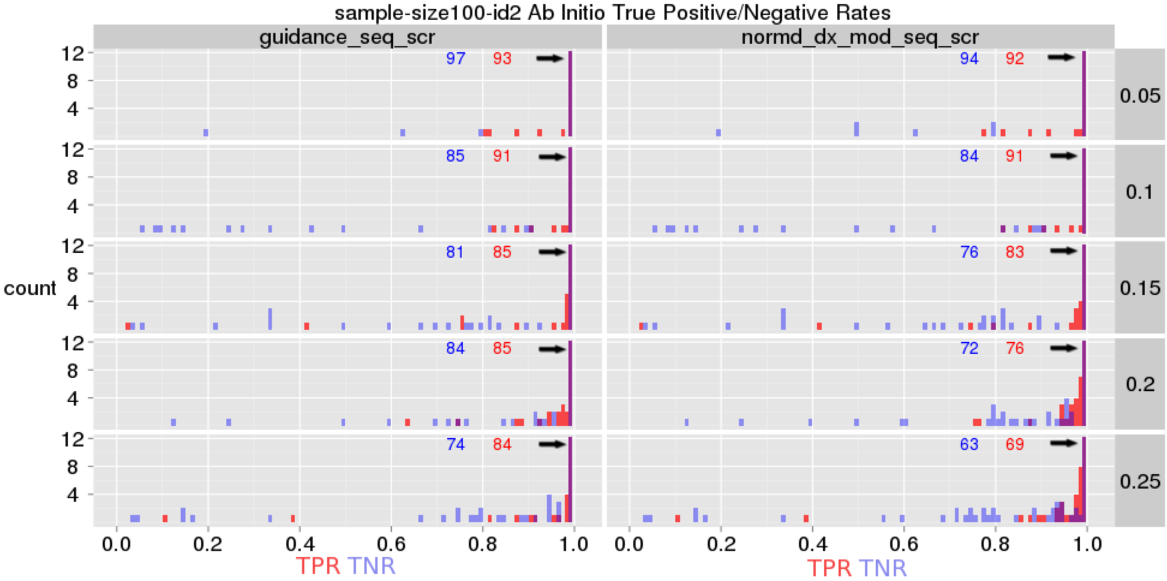


Figure 10: *Dynamic threshold selection.* Using both GUIDANCE and norMD sequence-level scores, our algorithm predicts thresholds for each MSA . The correctness of the threshold is evaluated using the sequences scores of both methods. The vertical axis shows the number of MSAs of Sample 2 for each TPR/TNR bin of size 0.01.

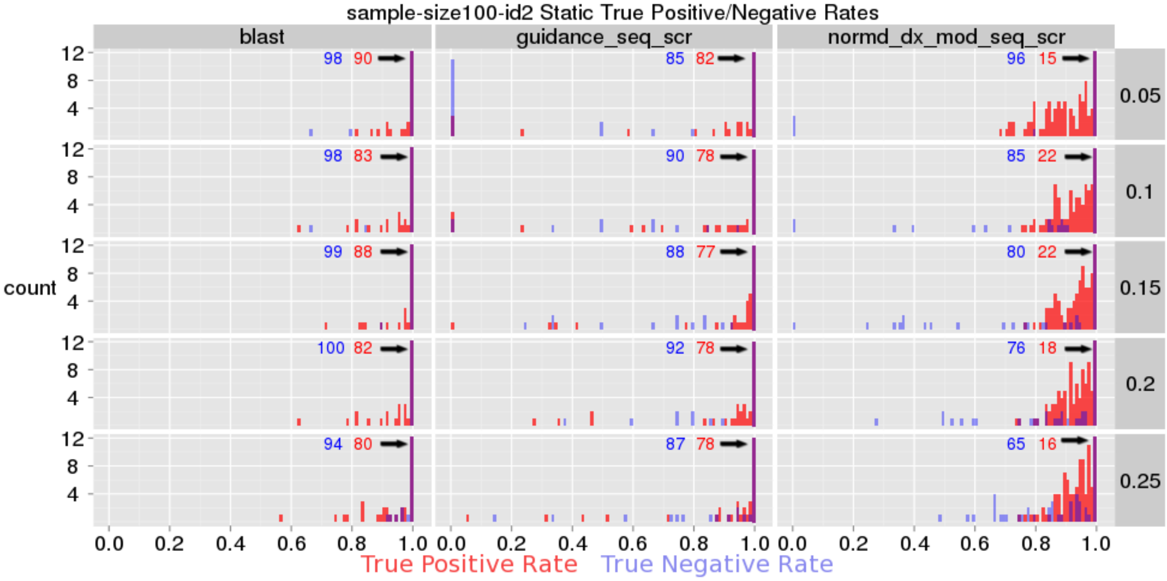


Figure 11: *Performance of the static threshold selection method.* Static thresholds that maximized MCC for Sample 1 were applied to Sample 2 per-sequence scores. On the vertical axis count of MSAs of Sample 2 are shown for each TPR/TNR bin of size 0.01.

4.6 Dynamic Threshold Selection by Min/Max Scaling

In previous section I discussed evaluating the dynamic threshold selection heuristic. In addition, I also found a much simpler method which works surprisingly well. This method is to linearly scale the sequence scores so a 0-1 range as defined in section 3.11.4. For 5% added nonmembers, dynamic threshold selection by scaling achieves a surprisingly good separation between the scores of member and nonmember sequences. The separation of scores before and after scaling is shown in Figure 12. In other random samples (not shown) and for 10%, 15%, 20%, 25% added nonmembers (not shown) I observed similar improvements in score separation. By design this method will produce false negatives when there are no nonmember sequences in the dataset because the scaling will always place at least one sequence bellow the threshold.

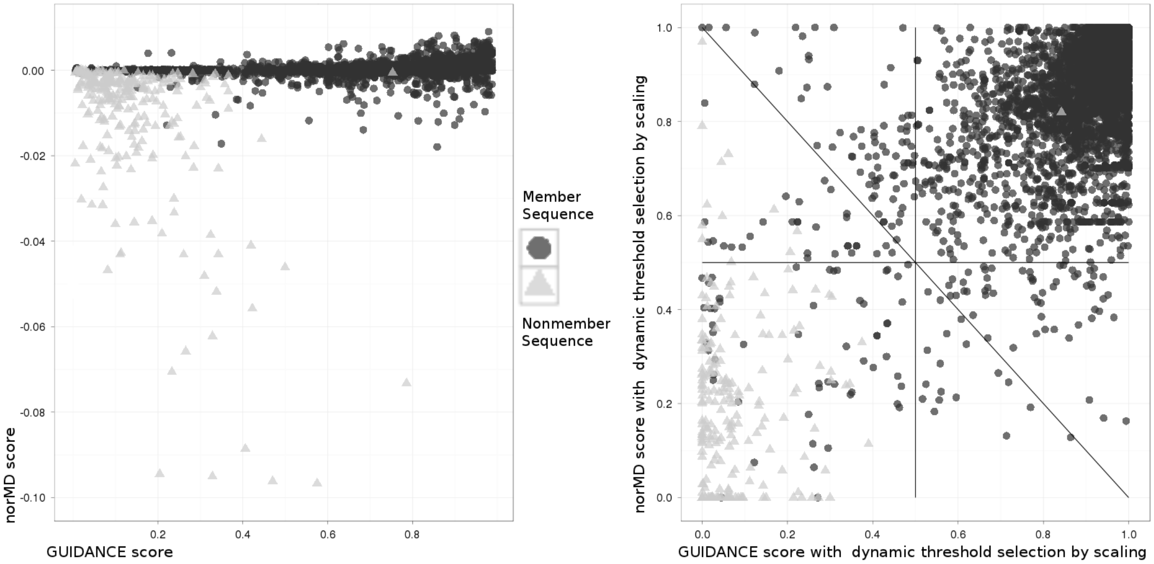


Figure 12: *Dynamic threshold selection by scaling.* The GUIDANCE and norMD sequence-level scores are plotted on the same plane for Sample 2 with 5% added nonmember sequences. The raw scores on the left show only limited separation. The scaled scores on the right show a much better separation with a defined threshold for separating members from nonmembers (the diagonal from [1,0] to [0,1]).

4.7 Scoring with BLAST

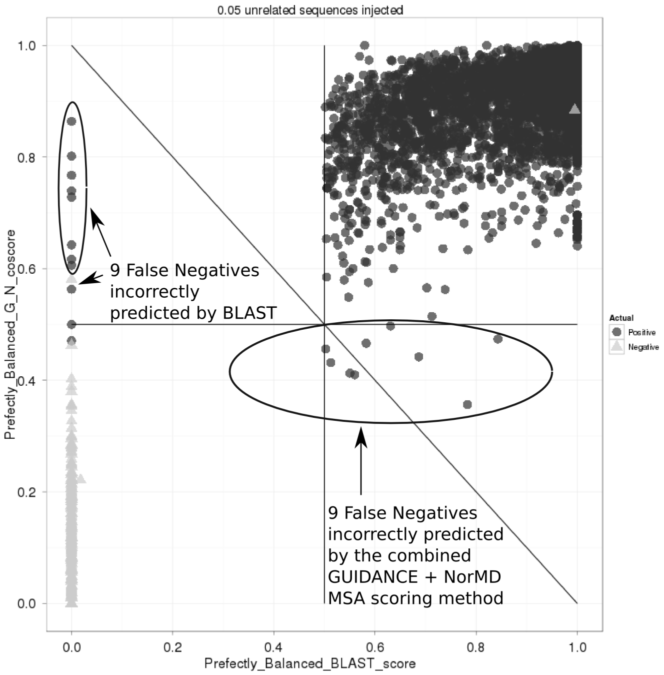


Figure 13: *Comparison of BLAST and combined GUIDANCE+norMD scores.* The scores were calibrated using the Perfect-Balance approach for dynamic threshold selection on MSA sample 1 (please refer to section 3.5) with 5% of nonmembers. The BLAST-based sequence level scores are on the horizontal axis. The combined norMD and GUIDANCE calibrated scores are on the vertical axis. Each darker circle is a member sequences. Each lighter triangle is a nonmember.

I compare the all-to-all pairwise BLAST scoring method with the combined GUIDANCE+norMD scoring method (please see Figure 13). Scores for both methods were normalized with a ground-truth aware named Perfect-Balance dynamic threshold selection. For all samples, BLAST (Altschul et al., 1990) and GUIDANCE+norMD scoring make only a small number of prediction mistakes. For sample 1, out of 6,041 predictions both methods had 12 false assignments (11 false negatives and 1 false positive). Interestingly, with the exception of single cases, all mistakes are unique to BLAST or the MSA scoring methods. Only 1 common false positive and 2 common false negatives were observed.

4.8 Comparing Scoring Methods with ROCs

In order to compare the performances of classifiers at different sensitivity / specificity levels, I superimpose ROC curves on the same plot. I then compare the AUC (area under the curve) values for a number of approaches to cleanup sequences families from nonmembers. Tests with raw, scaled, and Perfect-Balance norMD variants showed lower AUC values than the corresponding variations of the BLAST and GUIDANCE methods. All-to-all pairwise alignment BLAST method has same or better performance at detecting nonmember sequences than the performance of all tested approaches based on raw MSA sequence-level scores (shown in Figure 14).

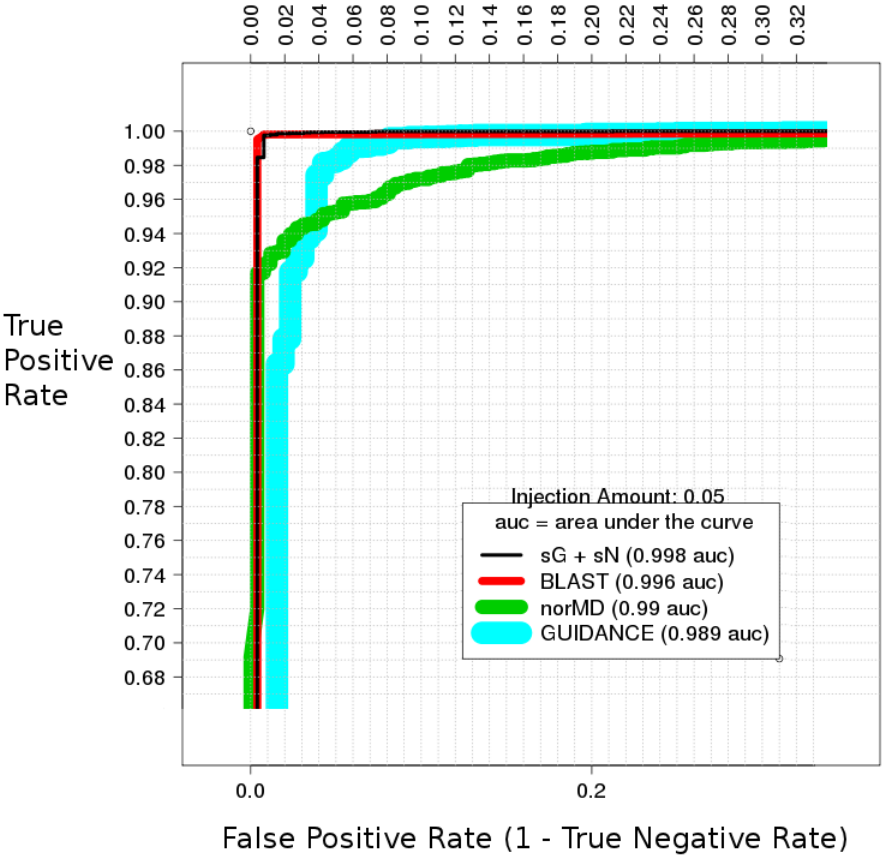


Figure 14: *ROC for norMD, GUIDANCE, and BLAST without threshold selection.* All sequence-level scores of Sample 1 with 5% nonmember sequences were combinded in one pool. All unique thresholds in this pool were evaluated to generate the ROC.

Name	**Acronym in Figures**	Section
Raw GUIDANCE	**rG**	3.6
Raw norMD	**rN**	3.7
Raw BLAST	**rB**	3.8
Scaled GUIDANACE	**sG**	3.11.4
Scaled norMD	**sN**	3.11.4
Scaled BLAST	**sB**	3.11.4
Perfect-Balance GUIDANACE	**pG**	3.11.3
Perfect-Balance" norMD	**pN**	3.11.3
Perfect-Balance" BLAST	**pB**	3.11.3

Table 5: *Acronyms of Test Approaches.* A list of all approaches that were studied in this project. Acronyms are provided as they are used in Figures 15 and 16. In addition, sections describing the methods are provided.

All-to-all pairwise alignment BLAST method also performed better at detecting nonmember sequences than the combined raw MSA sequence-level scores (shown in Figure 15). This trend is persistent across all tested amounts of added nonmember sequences (5%, 10%, 15%, 20%, and 25%). This a surprising finding because BLAST is only aligning pairs of sequences so it is able to use the rich information that is available to the MSA scoring methods when families of multiple sequences are aligned (please refer to section 1.1).

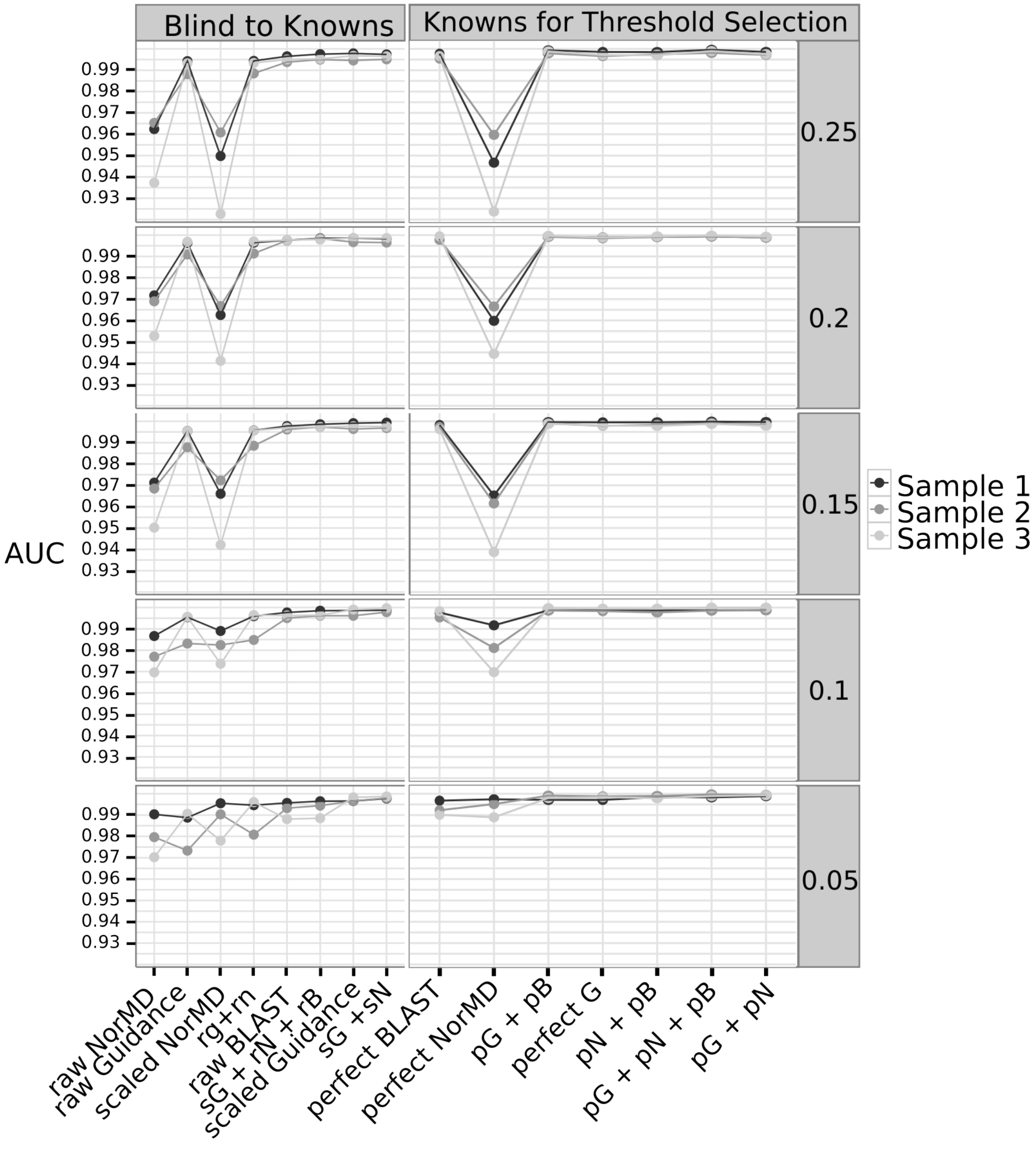


Figure 15: *Comparison of Area Under the ROC curve (AUC) for different methods* ROC curves were generated for various scoring methods and combined versions of the scores. Areas under the curves of the ROC are show for all amounts of added nonmember sequences. For abbreviations "r", "s", "p" denote raw, scaled, and perfect respectfully. "G", "N", and "B" denote GUIDANCE, norMD, and BLAST respectfully. "+" indicates combined classification. Details are provided in Table 4.8

4.9 BLAST-based and GUIDANCE-based cleanup methods

GUIDANCE sequence-level scores combined with my method of dynamic threshold by scaling (please refer to section 3.11.4) had significantly better performance of identifying nonmember sequences than the BLAST-based method (please refer to section 3.8). As can be seen in Figure 16, all three samples that were tested (please refer to section 3.5) had higher AUC values for my improved method (shown as *scaled GUIDANCE*) then for the all-to-all pairwise alignment BLAST method (shown as *raw BLAST*). This trend is persistent across all tested amounts of added nonmember sequences (5%, 10%, 15%, 20%, and 25%) as can be seen in Figure 15.

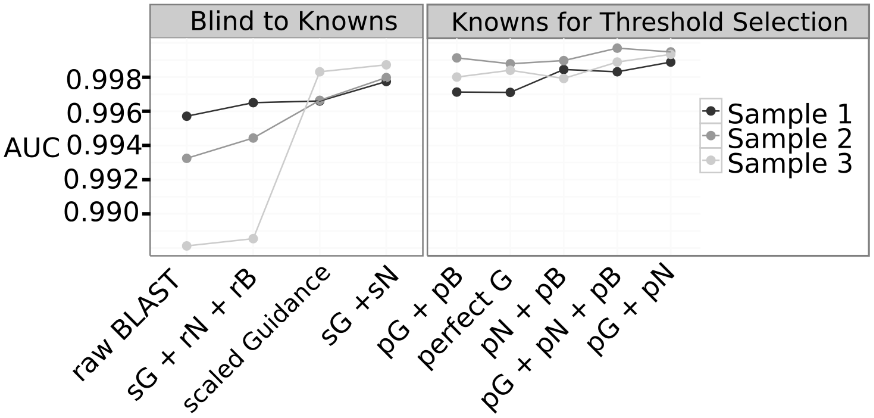


Figure 16: *AUC Zoom-ins for best results from Figure 15.* Areas under the ROC curve are shown for 9 methods that gave the best AUC performance for sequence families with 5% of nonmember sequences. Details are given in the caption of Figure 15.

5 Conclusions

An important finding of this study is that the MSA scoring methods considered here - norMD and GUIDANCE - show only moderate to poor performance in identifying nonmember contaminations in protein families. They are outperformed by simple all-against-all BLAST comparisons. This could be due to the fact that it is easier to generate pairwise alignments resulting in robust scores, while MSAs are much more complex. BLAST is highly optimized for homology detection between pairs of sequences and has a robust threshold systems (the e-values) that can be set statically to separate two non-homologous sequences. In contrast to this, the MSA scoring methods depend on the quality of the MSAs. If an MSA is of poor quality for both member and nonmember sequences then it is expected to misclassify them more often than a pairwise rescoring approach with BLAST. This aspect greatly reduces the effectiveness of static MSA scoring methods for removing nonmember sequences.

While the BLAST-based cleanup method consistently outperformed the other methods when using static thresholding on raw and combined scores, I was able to improve the GUIDANCE method by implemening a dynamic thresholding approach that exceeded the performance of the BLAST method. This improved GUIDANCE approach is now my tool of choice for revising MSAs in applied research projects where the identifiction of nonmembers is an important step in modeling protein families.

6 Contributions

This project makes the following contributions to the field of sequence family modeling through MSA quality assessment: (1) I developed and applied a framework for automated evaluation and comparison of sequence-level-capable MSA scoring methods. (2) I discovered that sequence cleanup techniques based on GUIDANCE and norMD are outperformed a BLAST-based method. (3) I developed a modified sequence family cleanup approach that outperforms the BLAST-based approach. The classifier that achieving this performance is a dynamic threshold selection technique applied to the sequence-level scores from GUIDANCE.

7 Future work

It would be interesting to try to adapt a norMD variant assisted with a neighbor-joining generated phylogeny as recommended by Thompson et al. (2001). The use of the NJ assisted norMD method is outside of the scope of this project because this method is unable to provide per-sequence scores. I was not able to include the NJ assisted norMD method in the comparisons because sequence-level scores are required to perform the score separation, threshold selection, and ROC analysis that were done in this project. The second limitation of the NJ assisted norMD method is the requirement of a cutoff value to be specified by the user. Automatic threshold is difficult because no information is provided by the NJ assisted norMD method without an *a priory* specification of the threshold. To overcome these limitations I propose extracting the sequence-level scores from the NJ assisted norMD method via the following procedure:

1. The NJ tree assisted norMD sequence removal method will be applied at various thresholds. Some number X will be added to the whole MSA NorMD score as the next threshold to be tested. I do not yet know how to assign values to X but the solution should be possible.

2. Run the NJ tree assisted sequence removal method. As the cutoff parameter supply the modified norMD score (Equation 8) of entire MSA plus X.

$$norMD = \frac{MD - GAPCOST}{MaxMD} \tag{8}$$

3. Increment X so that only one sequence will be removed. Assign the used cutoff to the removed sequence. Repeat the incremental and removal until all sequences are scored.

8 Implementation

Software was designed, tested, debugged, and refined to run all experiments and analysis in this project. In addition, housekeeping procedures were developed to: preprocess data into usable datasets, randomly sample data, apply MSA scoring methods to the datasets, and collect scores of all runs into uniform tables. "Trials" isolated data of particular set of upstream Uniprot and CDD versions.

All code was organized into individual steps which are executed on the command line. These utilities automatically determine what is the most up to date trial so no configuration is required. Each step is preconfigured for the correct set of local upstreams (other steps that are executed prior to the current step). On the command line the utilities will ask for the sample size as that is the only parameter that connects the whole run into a single pipeline. Smaller sample sizes can be used to test the software.

Three programming languages were used as best fits for various types of analysis: GNU Bourne-Again SHell (Bash) 70%, R 29%, and Ruby 1%.

Requirements	**Included Packages**
GNU Linux environment	MAFFT v6.857 (Katoh et al., 2002)
R (R Development Core Team, 2011)	GUIDANCE v1.1 (Penn et al., 2010)
Ruby	norMD 1_3 (Thompson et al., 2001)
Bioperl (needed by GUIDANCE)	GNU Parallel v20110822 (Tange, 2011)
dos2unix (needed by GUIDANCE)	bio3d 1.1-3 (Grant et al., 2006)
ROCR (Sing et al., 2005) (R package)	
ggplot2 (Wickham, 2009) (R package)	

Table 6: External and Internal dependencies

Tested on Debian GNU/Linux 6.0 (Lenny). One command does the complete build of all necessary components. Simply type: make. Should take about 4 minutes. Installs everything in <project>/opt - the local project directory.

9 Availability

- Code `https://github.com/alevchuk/ms`

- Data `http://biocluster.ucr.edu/projects/GROUPBALANCER/gbdata-1.0`

10 Supplements

Supplements are available at: `http://biocluster.ucr.edu/~alevchuk/thesis-supplements`

References

Ahola, V., Aittokallio, T., Vihinen, M., Uusipaikka, E., 2006. A statistical score for assessing the quality of multiple sequence alignments. BMC Bioinformatics 7, 484–484.
URL http://www.hubmed.org/display.cgi?uids=17081313

Ahola, V., Aittokallio, T., Vihinen, M., Uusipaikka, E., Oct 2008. Model-based prediction of sequence alignment quality. Bioinformatics 24 (19), 2165–2171.
URL http://www.hubmed.org/display.cgi?uids=18678587

Altschul, S. F., Erickson, B. W., 1986. Optimal sequence alignment using affine gap costs. Bull Math Biol 48 (5-6), 603–616.
URL http://www.hubmed.org/display.cgi?uids=3580642

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., Lipman, D. J., Oct 1990. Basic local alignment search tool. J Mol Biol 215 (3), 403–410.
URL http://www.hubmed.org/display.cgi?uids=2231712

Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E. L., Studholme, D. J., Yeats, C., Eddy, S. R., Jan 2004. The Pfam protein families database. Nucleic Acids Res 32 (Database issue), 138–141.
URL http://www.hubmed.org/display.cgi?uids=14681378

Bauer, M., Jan 2011. cd02338 ZZ PCMF like sequence family. http://goo.gl/JtLcC.

Benner, S. A., Cohen, M. A., Gonnet, G. H., Nov 1994. Amino acid substitution during functionally constrained divergent evolution of protein sequences. Protein Eng 7 (11), 1323–1332.
URL http://www.hubmed.org/display.cgi?uids=7700864

Capella-Gutiérrez, S., Silla-Martínez, J. M., Gabaldón, T., Aug 2009. trimAl: a tool for automated alignment trimming in large-scale phylogenetic analyses. Bioinformatics 25 (15), 1972–1973.
URL http://www.hubmed.org/display.cgi?uids=19505945

Carrillo, H., Lipman, D., October 1988. The multiple sequence alignment problem in biology. SIAM J. Appl. Math. 48, 1073–1082.
URL http://dl.acm.org/citation.cfm?id=53867.53874

Cartwright, R. A., Nov 2005. DNA assembly with gaps (Dawg): simulating sequence evolution. Bioinformatics 21 Suppl 3, 31–38.
URL http://www.hubmed.org/display.cgi?uids=16306390

Castresana, J., Apr 2000. Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. Mol Biol Evol 17 (4), 540–552.
URL http://www.hubmed.org/display.cgi?uids=10742046

Dayhoff, M. O., Schwartz, R. M., 1978. Chapter 22: A model of evolutionary change in proteins. In: in Atlas of Protein Sequence and Structure.

Durbin, R., Eddy, S. R., Krogh, A., Mitchison, G., Jul. 1999a. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, Ch. 5.
URL http://selab.janelia.org/cupbook.html

Durbin, R., Eddy, S. R., Krogh, A., Mitchison, G., Jul. 1999b. Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press.
URL http://selab.janelia.org/cupbook.html

Eddy, S. R., 1998. Profile hidden Markov models. Bioinformatics 14 (9), 755–763.
URL http://www.hubmed.org/display.cgi?uids=9918945

Edgar, R. C., 2004a. Local homology recognition and distance measures in linear time using compressed amino acid alphabets. Nucleic Acids Res 32 (1), 380–385.
URL http://www.hubmed.org/display.cgi?uids=14729922

Edgar, R. C., Aug 2004b. MUSCLE: a multiple sequence alignment method with reduced time and space complexity. BMC Bioinformatics 5, 113–113.
URL http://www.hubmed.org/display.cgi?uids=15318951

Felsenstein, J., 1985. Confidence limits on phylogenies: An approach using the bootstrap. Evolution 39 (4), pp. 783–791.
URL http://www.jstor.org/stable/2408678

Felsenstein, J., Sep. 2003. Inferring Phylogenies, 2nd Edition. Sinauer Associates.
URL http://www.sinauer.com/detail.php?id=1775

Finn, R. D., Mistry, J., Tate, J., Coggill, P., Heger, A., Pollington, J. E., Gavin, O. L., Gunasekaran, P., Ceric, G., Forslund, K., Holm, L., Sonnhammer, E. L., Eddy, S. R., Bateman, A., Jan 2010. The pfam protein families database. Nucleic Acids Res 38 (Database issue), 211–222.
URL http://www.hubmed.org/display.cgi?uids=19920124

Girke, T., Lauricha, J., Tran, H., Keegstra, K., Raikhel, N., Oct 2004. The cell wall navigator database. a systems-based approach to organism-unrestricted mining

of protein families involved in cell wall metabolism. Plant Physiol 136 (2), 3003–3008.
URL http://www.hubmed.org/display.cgi?uids=15489283

Gonnet, G. H., Cohen, M. A., Benner, S. A., Jun 1992. Exhaustive matching of the entire protein sequence database. Science 256 (5062), 1443–1445.
URL http://www.hubmed.org/display.cgi?uids=1604319

Grant, B. J., Rodrigues, A. P. C., ElSawy, K. M., McCammon, J. A., Caves, L. S. D., 2006. Bio3d: an R package for the comparative analysis of protein structures. Bioinformatics 22 (21), 2695–2696.
URL http://bioinformatics.oxfordjournals.org/content/22/21/2695.abstract

Haft, D. H., Selengut, J. D., White, O., Jan 2003. The TIGRFAMs database of protein families. Nucleic Acids Res 31 (1), 371–373.
URL http://www.hubmed.org/display.cgi?uids=12520025

Henikoff, S., Henikoff, J. G., Nov 1992. Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci U S A 89 (22), 10915–10919.
URL http://www.hubmed.org/display.cgi?uids=1438297

Hirosawa, M., Totoki, Y., Hoshida, M., Ishikawa, M., Feb 1995. Comprehensive study on iterative algorithms of multiple sequence alignment. Comput Appl Biosci 11 (1), 13–18.
URL http://www.hubmed.org/display.cgi?uids=7796270

Katoh, K., Misawa, K., Kuma, K., Miyata, T., Jul 2002. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. Nucleic

Acids Res 30 (14), 3059–3066.
URL `http://www.hubmed.org/display.cgi?uids=12136088`

Kim, J., Ma, J., Aug 2011. PSAR: measuring multiple sequence alignment reliability by probabilistic sampling. Nucleic Acids Res 39 (15), 6359–6368.
URL `http://www.hubmed.org/display.cgi?uids=21576232`

Klimke, W., Agarwala, R., Badretdin, A., Chetvernin, S., Ciufo, S., Fedorov, B., Kiryutin, B., O'Neill, K., Resch, W., Resenchuk, S., Schafer, S., Tolstoy, I., Tatusova, T., Jan 2009. The National Center for Biotechnology Information's Protein Clusters Database. Nucleic Acids Res 37 (Database issue), 216–223.
URL `http://www.hubmed.org/display.cgi?uids=18940865`

Landan, G., Graur, D., Jun 2007. Heads or tails: a simple reliability check for multiple sequence alignments. Mol Biol Evol 24 (6), 1380–1383.
URL `http://www.hubmed.org/display.cgi?uids=17387100`

Letunic, I., Doerks, T., Bork, P., Jan 2009. SMART 6: recent updates and new developments. Nucleic Acids Res 37 (Database issue), 229–232.
URL `http://www.hubmed.org/display.cgi?uids=18978020`

Marchler-Bauer, A., Lu, S., Anderson, J. B., Chitsaz, F., Derbyshire, M. K., DeWeese-Scott, C., Fong, J. H., Geer, L. Y., Geer, R. C., Gonzales, N. R., Gwadz, M., Hurwitz, D. I., Jackson, J. D., Ke, Z., Lanczycki, C. J., Lu, F., Marchler, G. H., Mullokandov, M., Omelchenko, M. V., Robertson, C. L., Song, J. S., Thanki, N., Yamashita, R. A., Zhang, D., Zhang, N., Zheng, C., Bryant, S. H., Jan 2011. CDD: a Conserved Domain Database for the functional annotation of proteins. Nucleic Acids Res 39 (Database issue), 225–229.
URL `http://www.hubmed.org/display.cgi?uids=21109532`

Penn, O., Privman, E., Landan, G., Graur, D., Pupko, T., Aug 2010. An alignment confidence score capturing robustness to guide tree uncertainty. Mol Biol Evol 27 (8), 1759–1767.
URL `http://www.hubmed.org/display.cgi?uids=20207713`

Pirovano, W., Feenstra, K. A., Heringa, J., Feb 2008. PRALINETM: a strategy for improved multiple alignment of transmembrane proteins. Bioinformatics 24 (4), 492–497.
URL `http://www.hubmed.org/display.cgi?uids=18174178`

Privman, O., Jun 2011a. GUIDANCE sequence score. `http://guidance.tau.ac.il/overview.html#SeqScore`.

Privman, O., Jun 2011b. GUIDANCE sequence score. `http://guidance.tau.ac.il/overview.html#Remove_SEQ`.

R Development Core Team, 2011. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.
URL `http://www.R-project.org/`

Raymond, J., Gardiner, E., Willett, P., 2002. RASCAL: Calculation of Graph Similarity using Maximum Common Edge Subgraphs. The Computer Journal 45 (6), 631–644.
URL `http://comjnl.oxfordjournals.org/cgi/content/abstract/45/6/631`

Rost, B., Feb 1999. Twilight zone of protein sequence alignments. Protein Eng 12 (2), 85–94.
URL `http://www.hubmed.org/display.cgi?uids=10195279`

Saitou, N., Nei, M., Jul 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol Biol Evol 4 (4), 406–425.
URL http://www.hubmed.org/display.cgi?uids=3447015

Shameer, K., Jun 2011. What is the largest collection of multiple alignments? http://biostar.stackexchange.com/questions/9052/what-is-the-largest-collection-of-multiple-alignments.

Sing, T., Sander, O., Beerenwinkel, N., Lengauer, T., 2005. ROCR: visualizing classifier performance in R. Bioinformatics 21 (20), 3940–3941.
URL http://bioinformatics.oxfordjournals.org/content/21/20/3940.abstract

Tange, O., Feb 2011. Gnu parallel - the command-line power tool. ;login: The USENIX Magazine 36 (1), 42–47.
URL http://www.gnu.org/s/parallel

Tatusov, R. L., Fedorova, N. D., Jackson, J. D., Jacobs, A. R., Kiryutin, B., Koonin, E. V., Krylov, D. M., Mazumder, R., Mekhedov, S. L., Nikolskaya, A. N., Rao, B. S., Smirnov, S., Sverdlov, A. V., Vasudevan, S., Wolf, Y. I., Yin, J. J., Natale, D. A., Sep 2003. The COG database: an updated version includes eukaryotes. BMC Bioinformatics 4, 41–41.
URL http://www.hubmed.org/display.cgi?uids=12969510

Thompson, J. D., Plewniak, F., Poch, O., Jul 1999. A comprehensive comparison of multiple sequence alignment programs. Nucleic Acids Res 27 (13), 2682–2690.
URL http://www.hubmed.org/display.cgi?uids=10373585

Thompson, J. D., Plewniak, F., Ripp, R., Thierry, J.-C., Poch, O., 2001. Towards a reliable objective function for multiple sequence alignments. Journal of Molecular

Biology 314 (4), 937 – 951.
URL http://www.sciencedirect.com/science/article/pii/S0022283601951873

UniProt Consortium, Jan 2011. Ongoing and future developments at the Universal Protein Resource. Nucleic Acids Res 39 (Database issue), 214–219.
URL http://www.hubmed.org/display.cgi?uids=21051339

Wang, L., Jiang, T., 1994. On the complexity of multiple sequence alignment. J Comput Biol 1 (4), 337–348.
URL http://www.hubmed.org/display.cgi?uids=8790475

Wickham, H., 2009. ggplot2: elegant graphics for data analysis. Springer New York.
URL http://had.co.nz/ggplot2/book

Wilbur, W. J., Lipman, D. J., Feb 1983. Rapid similarity searches of nucleic acid and protein data banks. Proc Natl Acad Sci U S A 80 (3), 726–730.
URL http://www.hubmed.org/display.cgi?uids=6572363