Austrian Marshall Plan Foundation
Research Report

Efficient estimation of *p-values* for HMModeller

Author:
Samuel S. Shepard

Supervisor:
Stefan Wegenkittl

January 12, 2011

Department of Information Technology & Systems Management
Salzburg University of Applied Sciences, Austria

Abstract

HMMmodeller leverages the power of profile HMMs for remote homologue identification while being easily customizable by non-technical researchers. We give a method for the efficient estimation of p-values using simulated protein sequences for the profile general distribution and a Pareto distribution fit to the tail of the empirical cumulative distribution function. The biological sequence general distributions are also analyzed and the distribution statistics are correlated with profile HMM properties and other distribution statistics.

1 Introduction

HMMmodeller is a profile HMM tool for remote homologue identification [1–3]. It was created with molecular biologists in mind, such that advanced, customizable query searches on protein databases can be carried out by the simple alteration of the HMM profile with little to no technical expertise. The current version of the software is written as an extension of Chimera [4] in both Java and Python and is the joint effort of the Salzburg University of Applied Sciences with Salzburg University.

Categorizing proteins into families is an ongoing problem [5]. A general assumption is that by accurately assigning protein family membership (or that of superfamily), the function and characteristics of new homologues can be more easily identified. In scanning protein databases for good profile matches, the score value alone may not be sufficient to tell the researcher if the protein is likely to be a new member. It is therefore necessary to put score values in the statistical context of *significance*. In this study, we analyze 77 protein families and provide a method for the efficient estimation of p-values based on simulated protein general distributions.

The p-value, in this context, is the measure of the number sequences that should score at least as extreme as the query protein sequence given that the null hypothesis is true, i.e., that the query sequence does not belong to the protein family or superfamily. Statistical significance is often ascribed to values of $p < 0.05$ for the rejection of the null hypothesis, hence our research focusses on the 95% exceedance tail of the general distributions. We generate simulated protein sequences for the general distribution tail and compare their p-value estimations to general distributions constructed from biological protein sequences. In addition to evaluating the empirical cumulative distribution functions, we follow in the footsteps of [6] by fitting a Pareto distribution to the tail of the general distribution at various sample sizes and testing the power of their p-value estimation. Finally, biological sequence general distributions are analyzed and their statistics correlated with profile HMM properties as well.

2 Methods

HMM profiles for 77 SCOP protein families were provided by the research group of Dr. Peter Lackner[1] and evaluated using *HMM Modeller* version 5 with default parameters [1]. The *R* statistical program was used to evaluate normality, kurtosis, skewness, mean, and the standard deviation of scores for various models [7]. For each family, the "plain score" is for the standard Viterbi algorithm while the "reverse score" is the score of the same profile in reverse order (or, equivalently, the reverse of the input query sequence). The reverse score was first implemented for use in SAM [8], another profile HMM tool. In order to correct for the length differences in the input query sequences, a "simple score" consisting of a one state null model, was used to divide the plain score and produce the "simple corrected score." Similarly, the "reverse corrected score" of the query sequences were produced by dividing the plain score by the reverse score null model.

The ASTRAL SCOP compendium, version 1.73, was obtained[2] and reduced to sequences with less than 40% identity [9–11]. The null or *general distributions* for each of the selected 77 profile HMMs were created by scoring all protein sequences in the reduced version of the ASTRAL database minus proteins within the same SCOP class relative to the profile HMM model used for scoring. For example, suppose we were to create a general distribution for the reverse corrected scores of the SCOP family *a.1.1.2*, then we would score all ASTRAL database proteins not in class *a*, such as proteins in families *b.1.1.2*, *c.1.11.2*, *et cetera*.

The general distribution shows the distribution of scores one would expect to see given that the null hypothesis is true, i.e., that a query protein does not match the profile HMM's family (or super family as the case may be). This means that each profile HMM has a different number of total ASTRAL samples relative to its SCOP class. Table 1 shows the number of samples used for the general distribution of any profile HMM model with respect to its class. We only include classes for the 77 profile HMM models being used in the analysis. Moreover, all general distribution have at least 7,000 protein sequences, which is the down-sample size of the Markov model order analysis as shown in Table 6.

[1]Department of Molecular Biology, University of Salzburg, electronic correspondence

[2]ASTRAL Home Page: `astral.berkeley.edu`

Table 1: The number of ASTRAL database protein sequence samples is given relative to all SCOP classes for each of the 77 studied profile HMMs. All general distribution sample sizes are therefore at maximum limited to the number listed with respect to the profile HMM's SCOP class.

profile HMM SCOP class	ASTRAL samples
a	7753
b	7503
c	7013
d	7248
e	9362
g	8960

2.1 Computation of statistics

The Lilliefors, Anderson-Darling, and Cramer-von Mises normality tests were performed using the *nortest* plug-in (version 1.0) by Juergen Gross for the *R* statistical program (version 2.11.1). Similarly, the kurtosis and skewness were computed using the *moments* plug-in (version 0.11) by Lukasz Komsta within *R*. Meta analysis of the results over all sample sizes and families was performed using a custom MySQL database (version 5.1.46) with PhpMyAdmin (version 3.3.7). Additionally, the slopes from Table 2 for the sequence length analysis were computed using a user-defined function (UDF) from SourceForge's MySQL UDF page[3] (version 0.3) with the patch to make it compatible with MySQL 5 applied as well. The UDF used a linear regression of two MySQL variables to create the slopes and y-intercepts. The discrepancy computations, mentioned in detail below, were carried out in MATLAB Release 2010b[4]. All MATLAB, *R*, unix shell, MySQL query, as well as Perl scripts used to process data are available upon request.

2.1.1 Discrepancy

The discrepancy or Kolmogorov-Smirnov statistic between two distribution functions is the supremum of the distance between their cumulative distribution functions. In Table 7 we take the discrepancies of two empirical cumulative distribution functions as well as the dis-

[3]MySQL extension functions: `sourceforge.net/projects/mysql-udf`

[4]The Mathworks, Inc.: `www.mathworks.com`

crepancies between an e.c.d.f. and a fitted Pareto distribution (using MATLAB's *paretotails* function)—each distribution being compared only with respect to the 95% exceedance values of the reference distribution tail. The discrepancy versus the Pareto tail is easy to compute. Let P be the fitted Pareto distribution and let F be some empirical cumulative distribution function over N samples with respect to distribution X. Moreover, let z_0 be the index for the first exceedance value in X at the 95% threshold. Thus, see that:

$$\text{discrepancy} = \max_{z=z_0}^{N} \{\max\{|F(X_z) - P(X_z)|,\ |F(X_z) - \frac{1}{N} - P(X_z)|\}\}. \tag{1}$$

Our method to compute the discrepancy between two empirical distribution functions is presented in the pseudo-code for Algorithm 1. The *ECDFdiscrepancy* function takes arrays X and Y with dimensions $N \times 2$ and $M \times 2$ respectively. The first column of the array contains the values of the reversed corrected scores or some other score that is being used to build the e.c.d.f. while the second column contains a constant ID for the array, 1 for X and 2 for Y. First the exceedencas of X and Y must be extracted using the pseudo-code subroutine *getExceedences*, which takes a parameter for the array as well as for the exceedance threshold (in this case 95%). The exceedances are concatenated into a third array called Z and sorted by their score values. The second dimension where the array IDs is stored is also concatenated and sorted in these operations, as specified by the pseudocode subroutine *sortRowsByColumn*.

The algorithm then proceeds as follows: (1) the current x and y values are computed with respect to the starting exceedance values for X and Y; (2) the next value in Z is tested for an x or y step; (3) distances are calculated based on the current step and the x or y updated; (4) a new discrepancy D is tested based on the computed distances; and (5) the algorithm repeats (2-4) until all values in Z have been analyzed. Our particular implementation was in MATLAB and is available upon request.

Algorithm 1 ECDFdiscrepancy(X,X, N, M)

Require: Given arrays X & Y of lengths $N \times 2$ & $M \times 2$ respectively, return discrepancy.

$Z \leftarrow concatenateArrays(getExceedences(X, 0.95),\ getExceedences(Y, 0.95))$

{Take the exceedances (95% threshold) from X & Y concatenate them into Z.}

$Z \leftarrow sortRowsByColumn(Z, 1)$ {Sort Z by the score values.}

$T \leftarrow lengthOfRows(Z)$ {The combined length of the exceedances.}

$D \leftarrow 0$ {Initial discrepancy is zero.}

$x_{current} \leftarrow floor(N * 0.95)/N - 1/N$ {Initialize the current x.}

$y_{current} = floor(M * 0.95)/M - 1/M$ {Initialize the current y.}

{Loop over all values in Z, stepping each x and y as appropriate.}

{Determine which distribution to step and take distances.}

for $j = 1$ to T **do**

  if Z(j,2) = 1 **then**

    $d_1 \leftarrow |x_{current} - y_{current}|$

    $d_2 \leftarrow |x_{current} + 1/N - y_{current}|$

    $x_{current} = x_{current} + 1/N$ {Update x with respect to the observed step in $Z(j, 2)$.}

  else

    $d_1 = |y_{current} - x_{current}|$

    $d_2 \leftarrow |y_{current} + 1/M - x_{current}|$

    $y_{current} = y_{current} + 1/M$

  end if

  {Take the maximum of the distances.}

  if $d_1 > D$ OR $d_2 > D$ **then**

    $D \leftarrow max(d_1, d_2)$

  end if

end for

return D

2.2 Grid computing for model evaluation

For the high performance computing of profile HMM families, the researcher built a Rocks Cluster version 5.3 using 14 HP compaq dc7100 compute nodes. The compute nodes utilized 2GB of RAM, 3.0Ghz Pentium 4 processors and 80GB scratch disks for processing. The root node itself utilized an Intel Core 2 Quad at 2.83Ghz with 3.5 GB of RAM. All compute nodes were connected via a local switch, and large files were cached prior to running large jobs.

Using a single compute node on our cluster, it took on average roughly 33 minutes per profile HMM family to compute and post-process the ASTRAL database (9536 sequences). In order to score and process the data for 7000 simulated protein sequences, it took on average about 24 minutes per family. With a dataset of 2000 simulated protein sequences the time drops to about 6.5 minutes on average. Using a least-squares linear regression on these data points and a y-intercept of 0, the number of minutes should on average be $t(x) = 0.0034 \cdot x$ (R^2 value of 0.9995). Hence, on average, 1000 sequences would take about 3.4 minutes to process, 4000 sequences about 14 minutes, and 500 sequences less than two minutes. The implication is that if one uses sequence scores to calibrate the profile HMM—estimating the p-value for query sequences using a general distribution of non-family (or superfamily) sequences—then for any given profile HMM, the time to wait becomes quite prohibitive past 4000 sequences (> 15 minutes). This is in fact the largest simulated protein dataset we include for our discrepancy analysis (see Table 7).

2.3 Generation of simulated protein sequences

The amino acid composition of our reduced ASTRAL SCOP database (with no more than 40% identity in the sequences) was analyzed using a custom Perl program. Precisely 300 of the 9536 ASTRAL protein sequences were excluded from the analysis for containing ambiguous amino acid residues (e.g., "x"). Using a Markov chain algorithm, 7000 simulated protein sequences were generated for each Markov model of orders 0 through 4. The model simulates the distribution of amino acid residues (order 0), pairs of acids (order 1), 3-mers (order 2), and so forth—each model training their variables using the frequency information

from our reduced version of the ASTRAL database. A total of 79 untrained transition variable *instances* ($\sim 0.007\%$) were encountered within the Markov model order 4 sequence generation process. In these cases, for simplicity, the next residue was picked at random with uniform probability. The Perl source code for the Markov chain algorithm is available upon request.

3 Results and Discussion

It is well known that scoring sequences with profile HMMs admits a length bias and requires length correction [12]. We examine the simple corrected and reverse corrected methods for removing this bias in the general distribution of the profile HMM. From there, the study examines the statistical properties of the profile HMM general distributions themselves and some first attempts at predicting their shape. Finally, we show how using simulated protein sequences can, perhaps with less computational power, be used to effectively estimate p-values using a simulated general distribution and Pareto tail fit.

3.1 Picking the best length correcting null model

Simple corrected scores require only one state for their null model while the reverse corrected score requires the same number of states as the HMM profile being corrected. Therefore, from a computational standpoint, using the simple corrected score is advantageous. However, in practice the resulting distributions given by the simple corrected score do not perform nearly as well as the reverse corrected score in terms of actual length correction. Moreover, we shall see in Section 3.2 that although the general distribution for the reverse corrected score is not normal, it is more normal-like than the simple corrected score in its skewness and mean, making it easier to study.

As discussed in Section 2.3, simulated protein sequences were generated using both published PAM (Q_{as}) files as well as using the protein composition found in ASTRAL SCOP (Q_{astral}). Therefore, in order to judge the dependence on sequence length we generated simulated proteins with fixed lengths and observed the shift in mean score in particular. Table 2 shows the general shift in mean score due to length for the plain score (without correction)

as well as for the reverse and simple corrected scores. All 77 profile HMMs were evaluated for simulated protein data (based on Q_{as}) at fixed lengths of 50, 100, 150, and 200 residues. Each fixed length dataset contained $\approx$ 7000 samples. For each protein family, the slope of the line for mean score vs. fixed length was evaluated and the average, minimum, and maximum absolute slope taken over all families within each scoring group.

In other words, for each fixed family f, we have three means (μ_P, μ_S, and μ_R) for the plain, simple corrected and reverse corrected scores respectively. Furthermore, we denote the mean of the plain score for the fixed length dataset of size 50 to be μ_{P50} and so forth. Let the set $L = \{50, 100, 150, 200\}$ be the lengths for fixed length datasets. We can then take the slope of the line estimated for the function $F(i) = \mu_{Pi}, \forall i \in L$ to be m_P, m_S for the simple corrected score, and m_R for the reverse corrected score. Now we can take the average, maximum, and minimum absolute slope values over all families ($N = 77$). For example:

$$AvgAbsSlope_X = \frac{1}{N} \sum_{f=1}^{N} |m_X^{(f)}|, \qquad X = \text{P, S, or R} \tag{2}$$

For Table 2, observe that the uncorrected score column, the plain score, has the highest average, maximum, or minimum slope for each distribution statistic (mean, kurtosis, standard deviation, or skewness) over all 77 profile HMMs. The average absolute slope over all families for the mean statistic shows the greatest shift due to length for all statistics measured—demonstrating the need for length correction. The general formula is $F(x) = -9.1 - 2.83x$. Let us examine the SCOP family *a.1.1.2* for an example. For $L = 50$, we expect $\mu_{P50} = -150.6 = -9.1 - 2.83 \cdot 50$. The actual μ_{P50} is -150.2, which is about 0.27% relative error. For $L = 200$ we see a drastic shift in the mean as well as spreading of the distribution: $F(200) = -575.1$, with the measured $\mu_{P200} = -574.5$ and relative error at 0.10%. Figure 1 shows the shift in the general (null) distribution for this family due to the fixed length of the dataset.

The simple corrected score uses a one-state null model to correct the plain score's length bias. As seen in Table 2, the correction using this method is very dramatic, but in general, under Q_{as} emission probabilities, the null distributions tend to fluctuate quite a bit from family to family in terms of their distribution statistics (the standard deviation of all 77

Table 2: The impact of fixed sequence length on the HMM models for each scoring column. The average, minimum, and maximum absolute slope (over all families) of the score distribution statistics (mean, kurtosis, standard deviation or skewness) versus the fixed length of simulated protein sequences are given for each score column (plain, simple corrected, and reverse corrected score).

Function	Plain score	Simple corrected score	Reverse corrected score
Means			
Avg. abs. slope	2.83	0.04	0.00
Min. abs. slope	2.82	0.03	0.00
Max. abs. slope	2.83	0.04	0.00
Kurtosis			
Avg. abs. slope	0.61	0.05	0.00
Min. abs. slope	0.55	0.00	0.00
Max. abs. slope	0.67	0.02	0.02
St. Dev.			
Avg. abs. slope	0.03	0.00	0.00
Min. abs. slope	0.03	0.00	0.00
Max. abs. slope	0.03	0.00	0.00
Skewness			
Avg. abs. slope	0.02	0.00	0.00
Min. abs. slope	0.02	0.00	0.00
Max. abs. slope	0.02	0.01	0.00

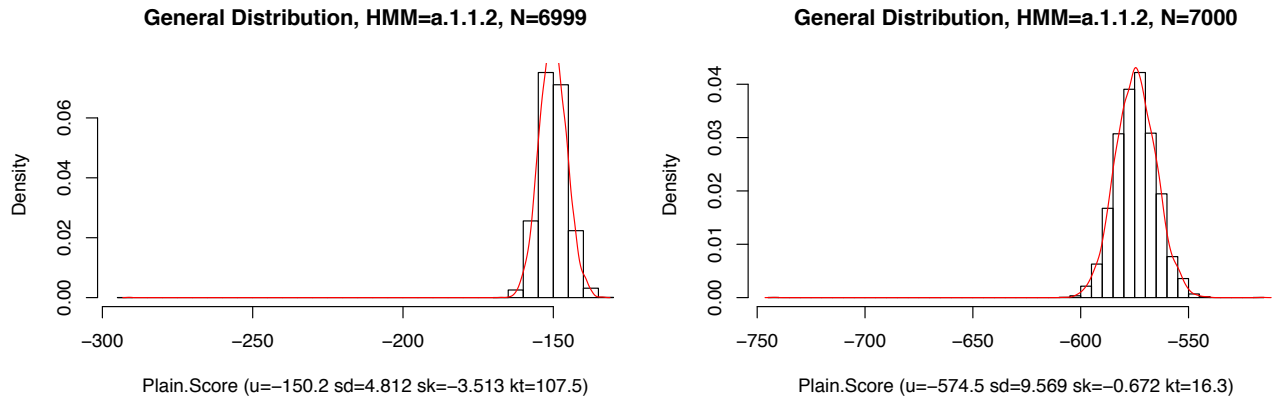

Figure 1: The impact of the simulated protein length on plain score. The general (null) distributions for SCOP family *a.1.1.2* are shown for both the simulated protein dataset at a fixed length of 50 residues (left panel) and at 200 residues (right panel).

null distribution means is 3.51 for the simple corrected score versus 0.0978 for the reverse corrected score). When it comes to length correction, the average absolute slope (see Equation 2) of the mean statistic drops from 2.83 for the plain score to 0.040 for the simple corrected score. The kurtosis, skewness, and standard deviation also drop at least an order of magnitude due to the simple score correction.

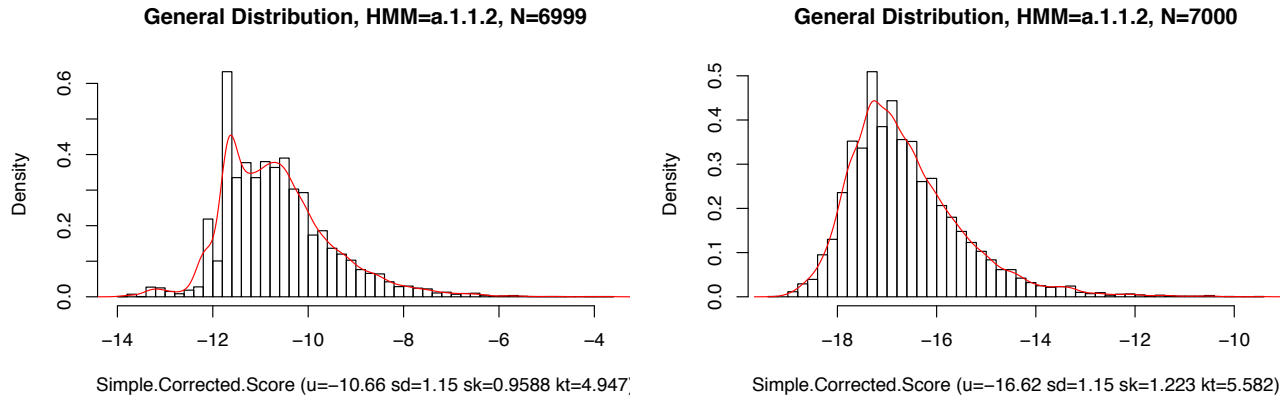

Figure 2: The impact of the simple score correction to alleviate length-dependent bias. The general (null) distributions for SCOP family *a.1.1.2* are shown for both the simulated protein dataset at a fixed length of 50 residues (left panel) and at 200 residues (right panel).

Continuing our example, Figure 2 shows the impact of the simple score correction for the general (null) distributions of SCOP family *a.1.1.2*—using just the fixed length datasets of 50 and 200 residues on the left and right respectively. Comparing Figure 2 to Figure 1 we notice immediately that the general distribution is not very symmetric; however, the change in mean is greatly subdued: only about 6 score points difference between the 50 and 200 fixed length datasets when corrected versus the uncorrected difference of roughly 420 score points.

The reverse corrected score uses the reverse HMM profile (or sequence) to correct the length bias in the plain score. Table 2 shows negligible change in mean score over all families due to the fixed length of the simulated protein datasets. All other changes in null distribution statistics (kurtosis, standard deviation, and skewness) due to the fixed length of the dataset are, in general, similar to the simple corrected score. Figure 3 reveals the length correction impact of the reverse score on the general distribution with respect to

our example family *a.1.1.2*.

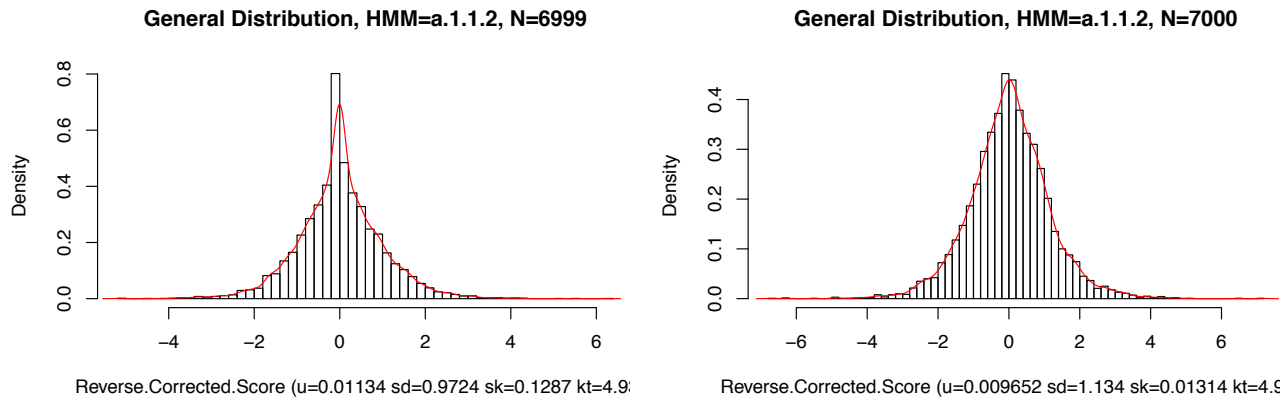

Figure 3: The impact of reverse score correction to alleviate length-dependent bias. The general (null) distributions for SCOP family *a.1.1.2* are shown for both the simulated protein dataset at a fixed length of 50 residues (left panel) and at 200 residues (right panel).

Other notable features of the reverse corrected score compared to the simple score correction is that the reverse score correction produces a much more symmetric general distribution as exampled in Figure 3. Indeed, the absolute skewness over all families using biological protein sequences from the ASTRAL database ranges from a minimum of 0.000542 to a maximum of 0.164 score points while the simple corrected score has a range of 0.0482 to 2.07 absolute skewness. The absolute mean score over all families has a minimum of 0.000965 to 0.428 while simple score correction has a range from 0.00294 to 12.4 in terms of absolute mean. Hence the reverse corrected score, in general, is much closer to 0 in terms of its mean score and skewness than the simple corrected score.

The purpose of the reverse and simple scores are to correct for biases due to sequence length within the plain score, therefore, we chose to use the reverse corrected score throughout the rest of this manuscript after observing how well it corrects the mean score of the general distribution. Moreover, the more symmetrical and near-zero mean with near-zero skewness properties of the reverse corrected score general distribution make it more simple to study. We believe, however, that the choice of emission probabilities for the simple score correction may be improved with the selection of Q_{astral} instead of Q_{as}—thus we reserve such exploration for future research.

3.2 Analyzing the general distributions of the reverse corrected score

A primary goal of statisticians is the characterization of observed distributions. Indeed, many tests exist to classify them, especially normality tests. We performed normality tests on the general distributions of the reverse corrected scoring column, in particular, the Lilliefors, Cramer-von Mises, and the Anderson-Darling tests. The summary results are given in Table 3. "Control data" was computed using 500 random samples taken from a normal distribution with the mean and standard deviation set to the values for the particular profile HMM family's general distribution. The general distribution was also sampled at 500 and 1000 samples for each family and the tests applied. Using the full 7000+ samples was not practical since even very small (and practically irrelevant) deviations between the distributions will lead to a lot of false values reported. All general distribution were, of course, of the reverse corrected score. The 0.05 significance threshold was used for all of the normality tests.

Table 3: The number of profile HMM family general distributions that do not reject the null hypothesis for normality. All numbers are out of a total of 77 families and with respect to the reverse corrected score general distribution. The significance level of the normality tests is reckoned at the 0.05 threshold. The normality control use 500 samples while the profile HMM general distribution uses both 500 and 1000 samples.

Normality test	Control 500	Gen. distr. 500	Gen. distr. 1000
Anderson-Darling	76	20	5
Cramer-von Mises	75	21	4
Lilliefors (Kolmogorov-Smirnov)	73	25	12

One may observe from Table 3 that the control samples properly fail to reject the null hypothesis for normality the majority of the time (76, 75, and 73 failed rejections for the Anderon-Darling, Cramer-von Mises and Lilliefors tests respectively). Switching to the general distribution 500 sequence sample yields roughly one third of the families remaining classified as normal under the null hypothesis (or failing to reject normality, more precisely). As the sample size is increased to 1000 sequences the number of families not rejecting normality drops to roughly one-fourth of the previous numbers under the Anderon-Darling and

Cramer-von Mises tests (5 and 4 respectively). When one examines the 5 general distributions that did not reject the Anderson-Darling normality test (for 1000 samples) in detail, the kurtosis of each distribution using the full 7000+ samples turns out to range from 3.42 to 3.69. Moreover, the smallest kurtosis of all 77 studied families is 3.23, which is still more peaked than the normal distribution. Hence, we believe that once the sample size is sufficiently large each profile HMM family general distribution should fail the normality hypothesis, provided the test is able to accurately function for larger samples of data.

One approach to estimating the p-values of profile HMM is to characterize the general distribution with a known parameterized distribution. We have seen that the reverse corrected score general distribution is near-zero in its mean and skewness but most likely not normally distributed due to its peakedness. In order to calculate the standard deviation and kurtosis of the general distribution based *a priori* on the profile HMM properties alone, a regression model would have to be fitted to the available 77 profile HMM family properties versus the statistics of their respective general distributions. First, we computed a correlation matrix in R using the Spearman correlation (for non-normal distributions) with normal and Holm adjusted p-values. The results are given in Table 4. Profile HMM properties included the average and total entropy of the match column emission probabilities, the average and total Jensen-Shannon divergence of the match columns emission probabilities from Q_{astral}, the number of match columns as well as the total columns in the profile, and the number of members included in the profile alignment. The number of general distribution samples, as observed in Table 1 was also included.

Table 4 shows that the number of match columns in the HMM profile significantly correlates with both the kurtosis and standard deviation of the general distributions ($\rho = 0.74$ & 0.58, $p < 0.01$ respectively). One can also take the Jensen-Shanon divergence of the emission probabilities for each match column or the Shanon entropy and then average them, but this will not correlate well with the general distribution statistics because the number of match columns is so important. If one sums the entropies of the match columns a smaller Spearman correlation is observed than for the match columns alone ($\rho = 0.72$ & 0.54, $p < 0.01$ for the kurtosis and standard deviation respectively). Similarly, if one uses the total number of profile columns the correlation is reduced even more. The reason for this is the same for both

Table 4: Spearman correlation of profile HMM properties with general distribution statistics (standard deviation & kurtosis). The average and total entropy or JSD (Jensen-Shannon divergence) values were with respect to the emission probabilities of the model match columns. The adjusted Holm p-value is also shown.

profile HMM property	Kurtosis			Standard Deviaion		
	Spearman ρ	p-value	adj. p	Spearman ρ	p-value	adj. p
Avg. entropy	-0.100	0.368	1.000	-0.320	0.005	0.099
Avg. JSD (Q_{astral})	0.090	0.450	1.000	0.290	0.011	0.181
Total entropy	0.720	0.000	0.000	0.540	0.000	0.000
Total JSD (Q_{astral})	0.790	0.000	0.000	0.760	0.000	0.000
# match columns	0.740	0.000	0.000	0.580	0.000	0.000
# total columns	0.330	0.003	0.067	0.460	0.000	0.001
# profile members	-0.310	0.007	0.122	0.370	0.001	0.026
# gen. distr. samples	-0.380	0.001	0.016	-0.330	0.003	0.067

profile HMM properties: noise in the contributing factor of the match columns. Consider the peakedness (kurtosis) and spread (standard deviation) of the general distribution scores, it is obvious that counting non-match columns in addition to the match ones will reduce the positive correlations with these statistics since non-match columns do not affect what counts as a "correct" protein nearly as much as the match columns do. In other words, more match columns are correlated with a higher frequency of near-zero general distribution scores—most likely via increasing the stringency of the HMM profile definition. What may be less obvious is how summing the entropy obscures the number of match columns. Let $p_i(x_j)$ be the jth emission probability of match column i in some profile HMM with M match columns, then:

$$\text{total entropy} = -\sum_{i=1}^{M}\sum_{j=1}^{20} p_i(x_j)\log_2(p_i(x_j)). \tag{3}$$

When a uniform distribution is considered for the emission probabilities, the match column is indifferent to the next amino acid and so entropy is high and information content is low. On other hand, when a particular amino acid is very important we will get a low entropy value. For example, under the simplified case where only two emission states are possible, see that $-[0.9 \cdot \log(0.9) + 0.1 \cdot \log(0.1)] = 0.14 < 0.30 = -2 \cdot [0.5 \cdot \log(0.5)]$. Hence, since match column count correlates directly with the general distribution kurtosis, most

likely by specifying a more stringent profile for a good score, then summing the entropies of the match columns will mask this correlation effect because higher information content columns (more stringent for a good score) will have smaller entropies and take away from column count effect being correlated.

The Jensen-Shannon divergence has been shown to be useful in biological sequence analysis [13], is symmetric, and produces non-negative reals. In our correlation analysis, summing the JSD does not suffer problems as summing entropies did, since the more different the two distributions being measured are, the larger their divergence value will be. The data here shown is for the Q_{astral} reference distribution compared to the match column emission probability distribution. The total or summed JSD has the highest observed correlation with both the kurtosis and standard deviation of the general distributions ($\rho = 0.79$ & 0.76, $p < 0.01$ respectively). The choice of the reference distribution is indeed also important. When one uses a uniform distribution in place of Q_{astral} the values are reduced to $\rho = 0.77$ & 0.75, $p < 0.01$ for the kurtosis and standard deviation respectively. Significant correlations are also observed for the number of members in the profile HMM alignment as well as with the general distribution sample size, albeit they are much smaller and inverse for the kurtosis. In particular, more members in the profile alignment tends to give rise to a more flat general distribution.

Figure 4 shows the linear regression of the total JSD with both the standard deviation and kurtosis. Since a Spearman correlation was used in Table 4 (ranked values), it is not clear until one examines the scatterplots that a linear model need not best fit the data. Indeed, we have observed that a logarithmic function gives a better least-squares fit than the linear model, especially with respect to the standard deviation (R^2 of 0.58 vs. 0.42). While the correlations we have observed in this section are very promising for using a parameterized distribution for our general distribution p-value estimations, finding the correct distribution and deriving those parameters *a priori* with great certainty will still require much future work. Hence, we turn to a simulation approach for estimating p-values in the next section.

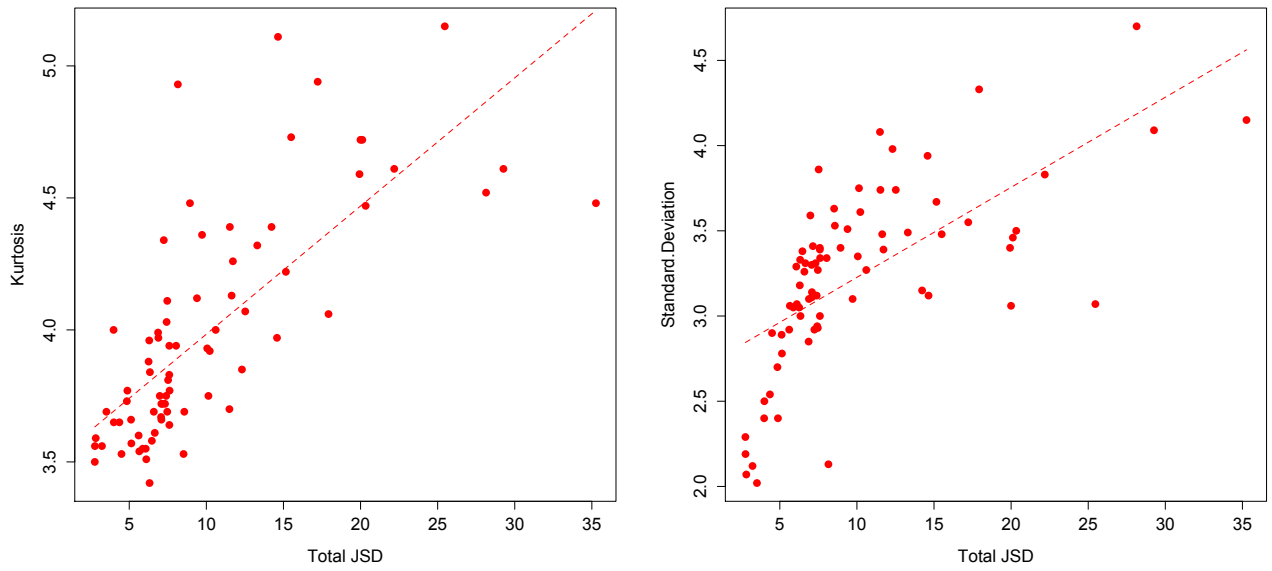

Figure 4: Scatterplot of the total Jensen-Shannon divergence of the match columns versus the kurtosis (left) and standard deviation (right) of the general distributions (over all 77 profile HMM families). The least-squares linear regression line is shown on each scatterplot.

3.3 Markov order effect on the simulated general distribution

Markov chains can be used to generate simulated sequences based on trained frequency tables. The larger the Markov model order k, the greater the memory of the chain—allowing for a more sophisticated simulation of the modeled proteins observed in the ASTRAL SCOP database. Unfortunately, the training requirements for the algorithm also go up with respect to the Markov model order. Table 5 gives the effect of the Markov model order on the transition variables. Since the size of the ASTRAL SCOP database is only about 9000 sequences, one cannot pick very large sizes of k or risk running out of statistical support. Indeed, the number of transition variables goes up by 20^{k+1}. Observe that the coverage, or training, of all possible transitions goes down at order 3 and 4. While the number of observation in general is fairly consistent, the number of observations per trained transition variable goes down exponentially.

A question open to argument is the minimum number of biological sequences needed in order to have good statistical support for one's classification or simulation methods. Since orders 0 and 1 have a minimum of 21943 and 273 observations per variable, one may conclude

these are quite safe choices for use with this database. Moreover, if one includes the untrained variables as well, the average number of observations per variable for orders 3 & 4 goes down to 9.89 and 0.492 respectively. Therefore, while increasing the order of the Markov model increases the sophistication of the simulation, it also increase the uncertainty of the estimated probabilities for the transition (and initiation) variables. By observing the trend in Table 5, the reader can see why going beyond order 4 is unwarranted for the purposes of our simulation analysis.

Table 5: The effect of the Markov model order on transition variables. From left to right: the number of trained variables, their percent coverage of the possible set of transitions, the number of transition observations taken from ASTRAL in the training step; the average, minimum, maximum and median number of observations per trained transition variable.

Order	#Variables	Coverage	Observations	Obs. per Variable	Min. Obs.	Max. Obs.	Median
0	20	100.0%	1610012	80500.60	21943	149823	85116.5
1	400	100.0%	1600776	4001.94	273	13144	3524.0
2	8000	100.0%	1591540	198.94	1	1434	149.0
3	149672	93.5%	1582304	10.57	1	406	7.0
4	986176	30.8%	1573068	1.60	1	198	1.0

Let us next compare the properties of the general distributions (reverse corrected score) created using protein sequences from ASTRAL versus the simulated protein sequences at various Markov orders. First, the sequences selected for the general distribution of each profile HMM family is sampled down to 7000 random samples to be consistent with the number of simulated protein sequences generated by the Markov model. Next, the kurtosis and standard deviation is computed for each family's general distribution and the biological and simulated protein datasets compared. Since the mean and skewness of the distribution for the reverse corrected score is expected to be near zero, we exclude them from the analysis. The relative error is calculated with respect to the simulated versus ASTRAL general distributions with the average and maximum values being obtained for all 77 profile HMMs. Relative errors are computed with the absolute value before the aggregate functions are assessed. The results are given in Table 6.

One may observe that the relative errors tend to be fairly stable across all Markov model

Table 6: Relative error of general distribution statistics for simulated protein datasets at different Markov orders. The mean and standard deviation relative and maximum error is with respect to general distributions calculated from the ASTRAL protein dataset versus simulated proteins generated by Markov models at various orders. All distributions use the reverse corrected score.

Markov order	Statistic	Avg. relative error	Max. relative error
0	Kurtosis	4.9%	16.2%
0	St. Dev.	3.3%	9.9%
1	Kurtosis	5.2%	16.8%
1	St. Dev.	3.5%	11.2%
2	Kurtosis	4.6%	19.0%
2	St. Dev.	3.2%	12.4%
3	Kurtosis	4.6%	18.0%
3	St. Dev.	3.5%	12.2%
4	Kurtosis	4.6%	16.7%
4	St. Dev.	3.7%	10.6%

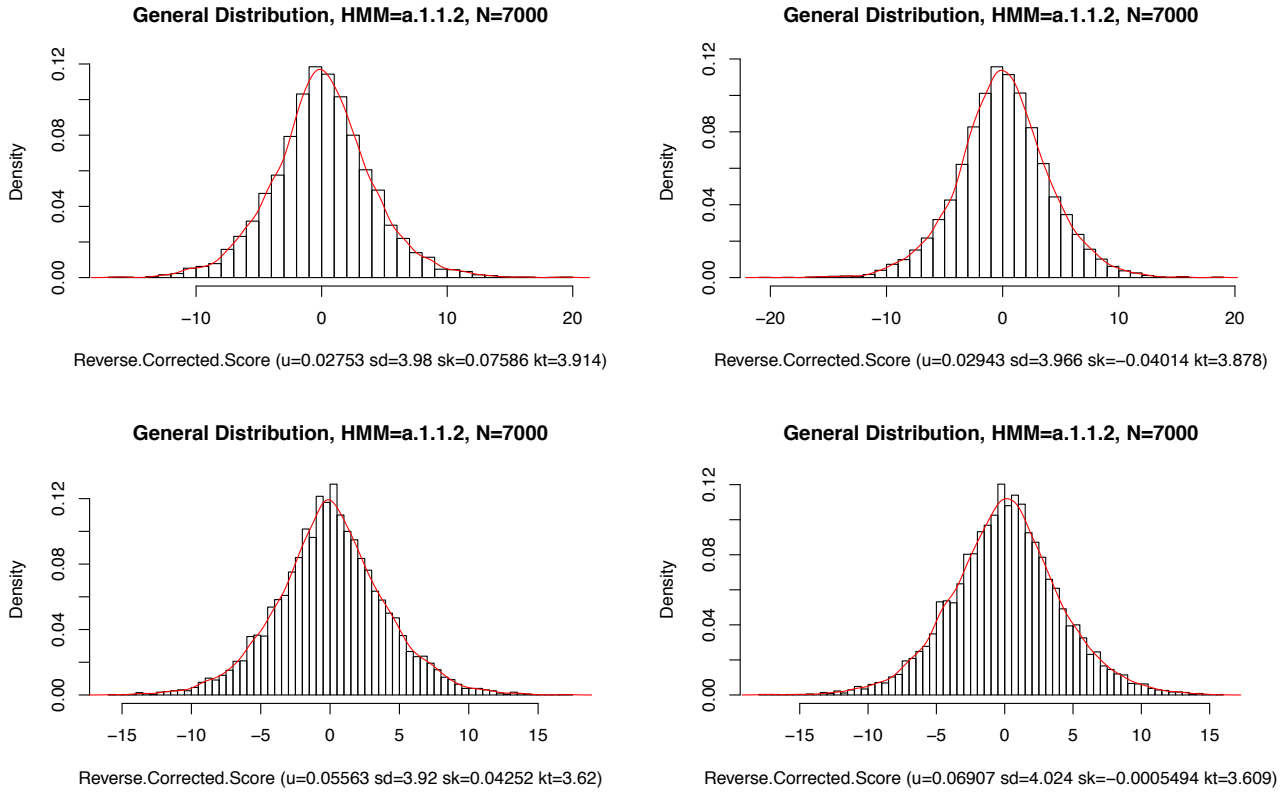

Figure 5: General distributions for the reverse corrected score of SCOP family *a.1.1.2* based on simulated and biological datasets. The top left density plot is the general distribution based on the ASTRAL database, the top right is constructed from 7000 Markov order 2 simulated protein sequences, the bottom left uses Markov order 0 while the bottom right uses Markov order 4 simulated protein sequences.

orders whether at maximum or on average. Moreover, the relative errors tend to be on average only about 5% which informs us that the general distributions created from the simulated proteins differs only a very little bit from the ones created by biological sequences when one considers the standard deviation and kurtosis statistics. Markov model order 0 has the smallest maximum relative error at 16.2% for the kurtosis and 9.9% for the standard deviation. Hence, for the purposes of this research we will primarily consider Markov order 0 for simulated protein sequences based on the Q_{astral} emission probability matrix.

As an example, Figure 5 shows the density plots of general distributions for the reverse corrected score of SCOP family *a.1.1.2* based on simulated and biological datasets. All four general distributions utilized 7000 samples. The plot of the Markov model order 2 simulated sequence general distribution (top right) most resembles the biological one (top left) for this particular profile HMM (kurtosis of 3.87 versus 3.91), while plots for Markov model order 0 (bottom left, kurtosis of 3.62) and order 4 (bottom right, kurtosis of 3.60) are also quite similar. Although we have selected Markov models of order 0 for our simulated protein dataset for reasons of simplicity and accuracy, a question for future study is if particular Markov model orders are significantly better than others for simulating the general distributions of particular profile HMMs.

3.4 Using simulated proteins to compute the p-value

We have already seen that simulated protein datasets appear to have similar general distribution statistics compared to the general distributions derived from the ASTRAL dataset. Next, simulated protein datasets based on a Markov model order 0 approximation of the ASTRAL database are explored for their ability to give good p-value estimates compared to general distribution p-value estimates derived from biological sequence scores. In order to test this, we calculate the discrepancy of the distribution tail at the 95% threshold versus the p-value estimate of the simulated sequence general distributions. Both a fitted Pareto distribution (S_{pareto}) and the e.c.d.f. (S_{ecdf}) are used for the simulated data general distribution comparisons. The algorithm for the discrepancy computations is discussed in depth within Section 2.1.1. The number of samples used to compute the cumulative distribution functions for the biological sequence general distribution data (B_{ecdf}) is with respect to the

class of the profile HMM under study and so class-specific sample sizes are shown in Table 1.

When fitting the Pareto distribution or calculating the e.c.d.f. of the general distribution tail, different sample sizes of simulated proteins were employed. One advantage of using simulated proteins is that one does not need to change the dataset with respect to the SCOP class of the profile HMM. More importantly, small datasets may give good p-value estimates. Hence, we used simulated datasets of 250, 500, 1000, 2000, and 4000 sequences. The discrepancy results are given in Table 7.

Table 7: The discrepancy values between a reference tail distribution at the 95% threshold and some alternative distribution. B_{ecdf} is the biological empirical cumulative distribution function, S_{ecdf} is the one based on simualted proteins, and S_{pareto} uses a Parto distribution fit to smooth the S_{ecdf} tail. Sample sizes are given for the alternative distribution. The average, minimum, and maximum discrepancy is given with respect to all 77 profile HMM families. Reverse corrected score general distributions were alone considered in this analysis.

Reference distribution	Alternative distribution	Number samples	Average discrepancy	Minimum discrepancy	Maximum discrepancy
B_{ecdf}	S_{ecdf}	7000	0.0067	0.0020	0.0187
S_{ecdf}	S_{pareto}	250	0.0131	0.0033	0.0321
S_{ecdf}	S_{pareto}	500	0.0097	0.0019	0.0219
S_{ecdf}	S_{pareto}	1000	0.0064	0.0011	0.0143
S_{ecdf}	S_{pareto}	2000	0.0047	0.0018	0.0107
S_{ecdf}	S_{pareto}	4000	0.0030	0.0016	0.0057
B_{ecdf}	S_{pareto}	250	0.0141	0.0032	0.0383
B_{ecdf}	S_{pareto}	500	0.0108	0.0021	0.0383
B_{ecdf}	S_{pareto}	1000	0.0084	0.0020	0.0266
B_{ecdf}	S_{pareto}	2000	0.0079	0.0018	0.0239
B_{ecdf}	S_{pareto}	4000	0.0068	0.0012	0.0189

The first section of Table 7 shows the average, minimum, and maximum discrepancy over all 77 profile HMMs of the biological sequence e.c.d.f. versus the simulated sequence cumulative distribution function. The S_{ecdf} always contains 7000 samples while the B_{ecdf} contains 7000+ samples based on the class of the profile HMM family. If the discrepancy values are high then using simulated protein sequences may be a poor choice for p-value estimates, while if the discrepancies remain small, then the use of simulated proteins will be similar to the use of biological sequences for the purposes of p-value estimates. On average, the discrepancy is about 0.0067 and ranges from 0.0020 to 0.0187 across all studied profile

HMMs. These numbers are acceptably low to warrant the use of simulated protein sequences (less than 1% on average and less than 2% at maximum). Hence, the Pareto distribution can capture the tail properties of the simulated sequence e.c.d.f. quite well and in a sample size dependent manner.

In the second section of Table 7 we consider the Pareto distribution and how well it approximates the S_{ecdf} itself. As previously stated, we test Pareto distributions fitted to down-samples of the simulated sequence data: 250, 500, 1000, 2000, and 4000 samples. The most obvious observation of the results is that increasing the number of total samples used in the general distribution produces better fits with respect to the empirical cumulative distribution function constructed from the full 7000 simulated sequences. The worst discrepancy observed for 250 samples is 0.0321 falling steadily until the maximum discrepancy is 0.0057 at 4000 samples across all 77 studied families.

The last section of Table 7 shows how well the Pareto distribution approximates the tail of the B_{ecdf} itself. The S_{ecdf} has already been shown to be quite similar to the B_{ecdf} and the S_{pareto} has been shown to be quite similar to the S_{ecdf} in a sample size dependent manner. The transitive property provides the intuition that S_{pareto} will also approximate the B_{ecdf} in a sample size dependent manner. This is the case. In fact, the discrepancy between B_{ecdf} and S_{pareto} at 4000 samples is already extremely comparable to the discrepancy between the B_{ecdf} and S_{ecdf} at 7000 samples—motivating the use of the Pareto distribution for the tail p-value estimate instead of the empirical cumulative distribution function. In particular, across all studied profile HMM families, the S_{pareto} performs better at minimum discrepancy (0.0012 vs. 0.0020) and slightly worse at maximum (0.0189 vs. 0.0187) or on average (0.0068 vs. 0.0067).

Depending on the desired level of accuracy, the researcher may choose to estimate the p-value using a Pareto distribution fitted to the tail but constructed from fewer samples. For example, one can construct a general distribution with only 1000 samples and still get an average discrepancy of 0.0084 across all studied families. As previously mentioned, other researchers have employed Pareto distributions within permutation tests to get accurate p-value estimates with fewer permutations [6].

Scoring smaller samples for the general distribution may be quite beneficial if computation

time is a concern (in addition to the simplicity of using one set of sequences for all profile HMMs). As noted in Section 2.2, we estimate that it takes approximately 14 minutes to process 4000 sequences for the average HMM profile using our cluster, while it should take about 3.4 minutes to score 1000 simulated sequences for the general distribution p-value calibration.

4 Conclusion

The reverse corrected score seems to behave more stably with respect to its mean score and slightly better in terms of actually correcting for the length bias inherent in scoring sequences with profile HMMs. Moreover, general distributions constructed from the reverse corrected scores of biological sequences show a near-zero mean as well as near-zero skewness, which is easier to study. Simulated protein sequences can be generated from the observed amino acid frequencies of the ASTRAL database using Markov models of various orders. We found that Markov model order 0 is easy to compute, has plenty of statistical support from the ASTRAL database, and reasonably approximates the biological sequence general distributions in terms of distribution statistic (mean, kurtosis, standard, deviation, and skewness) as well as tail e.c.d.f. discrepancy.

We have shown that one can use linear regression on general distribution parameters (mean, kurtosis, etc.) versus profile HMM properties. However, the distribution type to parameterize is not clear and the regression permits some uncertainty. The Pareto distribution, however, can easily be fitted to simulated protein data and allows for p-value estimates similar to the empirical cumulative distribution functions derived from biological sequences but using 1000s of fewer sequence samples than either biological or simulation-based empirical cumulative distribution functions. Scoring fewer sequences can save computation time, and using simulated sequences provides the simplicity of having a single dataset for all profile HMMs without respect to its SCOP classification.

5 Acknowledgements

This work was funded by the Marshall Plan Foundation of Austria and performed at Fachhochschule Salzburg under the direction of Univ. Doz. Dr. Stefan Wegenkittl. I am grateful to Gabriele Abermann and Larry Hatch for their help in recruiting me to the project. Simon Kranzer, Peter Ott, and Bernadette Himmelbauer were very helpful in providing both technical and non-technical support during the project. Finally, Roland Graf and especially Stefan Wegenkittl have done much to advance *HMMmodeller*—without their work contribution, insightful discussion, and sharing of time I would have been quite lost.

References

[1] F. F. Auer, "Scoring schemes and parameter prediction for profile hmms," Master's thesis, Fachhochschule Salzburg, 2009.

[2] P. Lackner, F. Auer, M. Radlingmaier, and S. Wegenkittl, "Optimierte modelle zur beschreibung von proteinfamilien," in *Drittes Forschungsforum der Österreichischen Fachhochschulen*, (Fachhochschule, Kärnten), April 15–16, 2009.

[3] S. Wegenkittl, F. Auer, D. Bindreither, and P. Lackner, "Expert knowledge enhanced structure based profile hmms for protein sequence families." In Posterpresentation at the 3DSig, 2009.

[4] E. F. Pettersen, T. D. Goddard, C. C. Huang, G. S. Couch, D. M. Greenblatt, E. C. Meng, and T. E. Ferrin, "Ucsf chimera–a visualization system for exploratory research and analysis.," *J Comput Chem*, vol. 25, pp. 1605–1612, Oct 2004.

[5] V. Kunin, I. Cases, A. J. Enright, V. de Lorenzo, and C. A. Ouzounis, "Myriads of protein families, and still counting.," *Genome Biol*, vol. 4, no. 2, p. 401, 2003.

[6] T. A. Knijnenburg, L. F. A. Wessels, M. J. T. Reinders, and I. Shmulevich, "Fewer permutations, more accurate p-values," *Bioinformatics*, vol. 25, pp. i161–8, Jun 2009.

[7] R Development Core Team, *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. ISBN 3-900051-07-0.

[8] K. Karplus, C. Barrett, and R. Hughey, "Hidden markov models for detecting remote protein homologies," *Bioinformatics*, vol. 14, no. 10, pp. 846–856, 1998.

[9] S. E. Brenner, P. Koehl, and M. Levitt, "The astral compendium for protein structure and sequence analysis," *Nucleic Acids Res*, vol. 28, pp. 254–6, Jan 2000.

[10] J.-M. Chandonia, N. S. Walker, L. Lo Conte, P. Koehl, M. Levitt, and S. E. Brenner, "Astral compendium enhancements," *Nucleic Acids Res*, vol. 30, pp. 260–3, Jan 2002.

[11] J.-M. Chandonia, G. Hon, N. S. Walker, L. Lo Conte, P. Koehl, M. Levitt, and S. E. Brenner, "The astral compendium in 2004," *Nucleic Acids Res*, vol. 32, pp. D189–92, Jan 2004.

[12] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological sequence analysis*. Cambridge Univ Press, eleventh ed., 2006.

[13] I. Grosse, P. Bernaola-Galván, P. Carpena, R. Román-Roldán, J. Oliver, and H. E. Stanley, "Analysis of symbolic sequences using the jensen-shannon divergence," *Phys Rev E Stat Nonlin Soft Matter Phys*, vol. 65, p. 041905, Apr 2002.