Classifying proteins by family using the product of correlated p-values

Timothy L. Bailey
NPACI/SDSC, MC 0505
9500 Gilman Drive, Bldg 109 (619) 534-8350
La Jolla, California 92093-0505 Fax: 534-5113
tbailey@sdsc.edu

William Noble Grundy
Department of Computer Science
University of California, Santa Cruz (831) 459-2078
Santa Cruz, California 95060 FAX 459-4829
bgrundy@cse.ucsc.edu

Abstract

An important goal in bioinformatics is determining the homology and function of proteins from their sequences. Pairwise sequence similarity algorithms are often employed for this purpose. This paper describes a method for improving the accuracy of such algorithms using knowledge about families of proteins. The method requires a library of protein families against which to compare query sequences. A standard pairwise similarity search algorithm is used to search the library with the query, and a new variant of the Family Pairwise Search (FPS) algorithm converts the results into a list sorted by the E-values of the matches between the query and the families. The E-value of each query-family match is calculated using a statistical distribution introduced here that describes the behavior of the product of the p-values of *correlated* random variables. We also describe an algorithm (ESIZE) for estimating the single parameter of this distribution. This parameter summarizes the amount of correlation among the p-values being multiplied, which corresponds, in this application, to the divergence among the sequences in a family. We show empirically that the E-values reported by this variant of FPS are accurate and that the method has significantly superior classification accuracy than using the best pairwise p-value as the query-family match score. The new algorithm is closely related to an earlier version of FPS that combines similarity scores by averaging "bit scores", which has been shown to have superior classification performance compared with several model-based methods (motifs, HMMs), but lacks E-values and their concomitant advantages.

1 Introduction

Protein sequences and protein domains can be grouped into families according to function, structure or homology. Recent work [Grundy, 1998b; Grundy, 1998a] describes the Family Pairwise Search (FPS) algorithm for searching a database of sequences using a set of family member sequences as the query. The FPS algorithm computes the match between a family and a sequence by combining the pairwise match scores (computed using a sequence comparison method such as the Smith-Waterman algorithm [Smith and Waterman, 1981]) of the sequence and each member of the family. The FPS algorithm was shown to be better at classifying a database of sequences than searching with a single family member or with a statistical model of the family.

In this paper we study the inverse problem—searching a library of protein families using a single sequence as the query—using a variant of the FPS algorithm that uses a scoring function for which accurate p-values can be estimated. Throughout this paper, we will refer to this version of FPS as the POP (product of p-values) algorithm because it combines the pairwise scores by taking the product of their p-values. The input to the POP algorithm consists of a single query sequence and a sequence family library. The library is comprised of a database of sequences (or sequence fragments), a dictionary that lists the family (or families) to which each sequence belongs, and, for each family, the value of the parameter of the distribution of the product of p-values. The output of POP is the list of families sorted by the E-value of the match of the query to the family.

The POP algorithm can be implemented using any pairwise search algorithm that returns accurate p-values. The query is searched against the database using the pairwise search algorithm, and the product of p-values of the matches between the query and each member of a family is converted to a p-value for the overall match to the family using the distribution described below. Converting the combined score (product of p-values) to a p-value allows matches between the query and families with different numbers of members to be directly compared, since it puts all scores on a common scale. This p-value is then multiplied by the number of families in the library to give the E-value of the match, which can be used to estimate the expected number of false positives at or above that match in the list of matching families. The single parameter for the distribution of the product of p-values is computed for each family using the ESIZE algorithm (described below) when the library is built.

We study two aspects of the POP algorithm to determine its reliability and accuracy. First, we show that it returns reliable E-values that accurately predict the number of false positives. This verifies that the distribution function we propose for the product of correlated p-values is appropriate in this application. Second, we study the sensitivity versus selectivity tradeoff of POP. We show that POP has uniformly better coverage (sensitivity) at any level of selectivity (false positive rate) when compared with using the

From: Third International Conference on Computational Molecular Biology (RECOMB99), Lyon, France, April 11–14, 1999

best match (minimum p-value match) between the query and any member of a family (the MINP algorithm, described below). This demonstrates that POP effectively combines the information contained in the matches between the query and the family members into a single score that more accurately classifies the query than just using the best match. We substantiate this result by showing that the improvement in classification quality, as measured by the Receiver Operating Characteristic (ROC) [Gribskov and Robinson, 1996]), is highly statistically significant.

The remainder of this paper is organized as follows. First, we describe our method for estimating the distribution of the product of correlated p-values. Second we describe the methods and databases we use for testing the POP algorithm, as well as the results of those tests. Finally, we discuss the POP algorithm in relation to other methods of family classification.

2 Distribution of the product of correlated p-values

The Family Pairwise Search algorithm computes, for a given query sequence and family, a set of p-values for the matches between the query and each sequence in a family. We would like to combine the information contained in these p-values into a single score that accurately reflects the likelihood that the query belongs to the family and for which we can calculate the statistical distribution. Previous work on scoring the match of a query to a set of motifs [Bailey and Gribskov, 1998] shows that the product of the p-values satisfies both criteria as long as the p-values are independent. Unfortunately, the p-values for matches between a query and members of a sequence family are clearly correlated, so the formula given by Bailey and Gribskov [1998] for the distribution of the product, Z_n, of n *independent* pairwise p-values,

$$Pr(Z_n \leq p) \approx p \sum_{i=0}^{n-1} \frac{(-\ln p)^i}{i!}, \quad (1)$$

does not apply. On the other hand, when all of the family member sequences are completely dependent (i.e., identical), the p-value of the product is just the value of the identical pairwise p-values:

$$Pr(Z_n \leq p) = p^{1/n}. \quad (2)$$

In that case, it is as though the family only contains one sequence, so $n-1$ of the pairwise p-values can be ignored. Thus, Eqn. 1 and 2 apply to two extreme cases: when the family consists of n independent sequences and when it effectively consists of a single sequence. Real sequence families will lie between these two extremes, displaying a partial dependence among the sequences in the family. For these intermediate cases, interpolating between Eqn. 1 and 2 suggests the following equation for the distribution of $correlated$ p-values:

$$Pr(Z_n \leq p) \approx p^y \sum_{i=0}^{\lfloor m \rfloor - 1} \frac{(-\ln p^y)^i}{i!} + p^y (m - \lfloor m \rfloor) \frac{(-\ln p^y)^{\lfloor m \rfloor}}{\lfloor m \rfloor !} \quad (3)$$

where n is the number of members in the family, Z_n is the product of the n pairwise p-values, m is the "effective size" of the family, and $y = m/n$. The parameter m can range between 1 and the actual family size, n. Note that Eqn. 3 converges to Eqn. 2 as $m \to 1$ and to Eqn. 1 as $m \to n$.

The following algorithm, which we call "ESIZE", estimates the effective family size, m, in Eqn. 3 using random sequences as queries. It makes use of the fact that the expected value of the ith smallest of n p-values is $i/(n+1)$. The ESIZE algorithm estimates m by minimizing an RMS (root mean squared) error function that gives equal weight to equal proportional errors in the observed and predicted p-values:

$$E(m) = \sqrt{\sum_{i=1}^{n} [\log(p_i(m)) - \log(i/(n+1))]^2}, \quad (4)$$

where $p_i(m)$ is the ith largest p-value (of the product of p-values) among matches between the given family and n random query sequences.

The ESIZE algorithm searches for the value of m that minimizes the error function (Eqn. 4) using bracketing search [Press *et al.*, 1986]. This works well because the error function generally has a single minimum (data not shown). We have found that approximately 1000 random scores for each family member are sufficient for good estimates of m. To generate these scores, the same pairwise search algorithm as will be employed with POP is used with 1000 shuffled, randomly selected SWISSPROT sequences as the queries.

Fig. 1 shows a typical example of the fit of the distribution to observed data. The family contains 12 sequences, but the effective family size is estimated by the ESIZE algorithm to be 7.15. The average RMS error is 0.14 for this value of m. The data consists of POP p-values for 1000 shuffled Swissprot sequence queries against a single family in the SCOP [Murzin *et al.*, 1995] database. In the next section, we give further evidence that the distribution in Eqn. 3 holds well in practice.

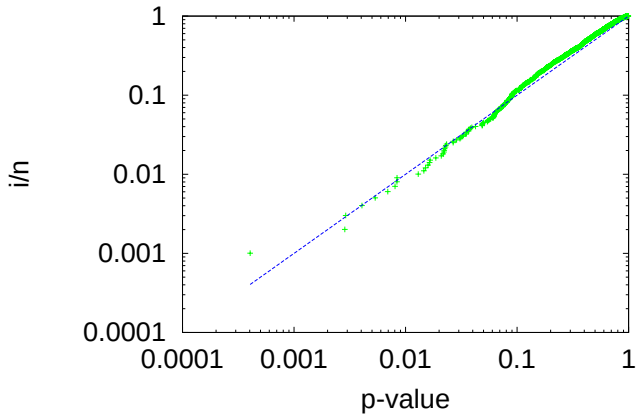

Figure 1: **Accuracy of the distribution of the product of correlated p-values.** The plot shows a typical example of the distribution of p-values computed by the ESIZE algorithm for a single family containing 12 sequences. See text for details.

3 Results

To evaluate the POP algorithm, we test it for the accuracy of the E-values it reports and for its sensitivity versus selectivity tradeoff. For the selectivity-sensitivity test, we compare the performance of POP with another variant of FPS which we will refer to as the MINP (minimum p-value) algorithm. The MINP algorithm sets the match score between a query and a family as the minimum p-value among all the pairwise p-values between the query and members of the family. As the underlying pairwise search algorithm, we use the Smith-Waterman (S-W) algorithm implemented on a Bioccelerator BioXL/P processor [Compugen, Ltd., 1998] with the default

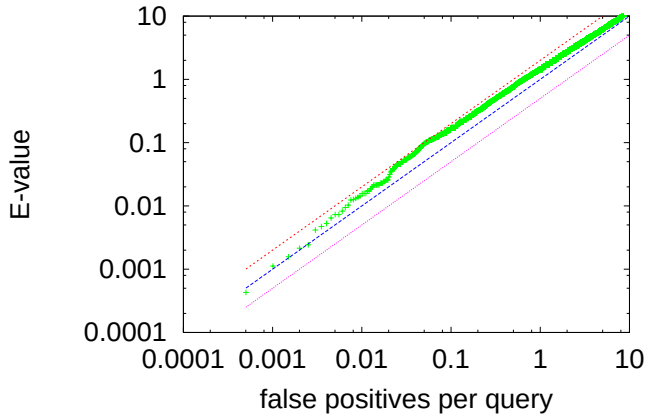

Figure 2: **Accurate prediction of false positive rates by POP E-values.** The plot shows the relationship between E-value and false positives per query for 1000 random sequence queries against all SCOP families with two or more members. Central line is the ideal, upper and lower lines are factors of two away.

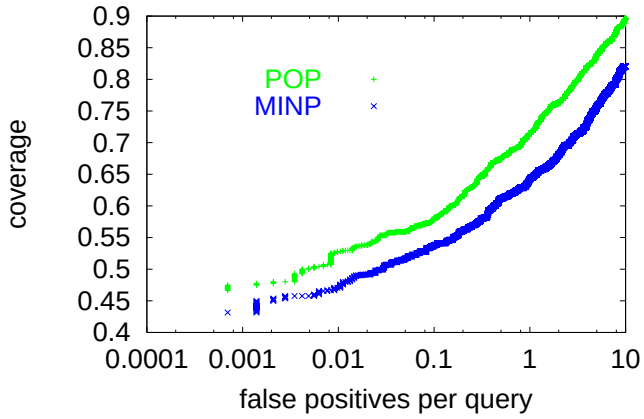

Figure 3: **Improvement in classification accuracy using the POP algorithm.** The plot shows the cross-validated fraction of correct query-family relationships found at different false positive rates per query for all 1434 queries against the split SCOP database.

gap opening penalty and extension penalties of 10 and 0.05, respectively. We built the sequence family library from the "superfamilies" in the SCOP [Murzin *et al.*, 1995] database (version 1.37) of structurally classified proteins. The database is purged to contain only sequences with less than 40% sequence identity as described in Brenner *et al.* [1998].

We measure the reliability of the POP E-values by assessing the number of false positives per query observed at a given E-value. Each of 1000 shuffled SWISSPROT sequences is used as a query to POP to search the family library. The combined results are then sorted by E-value.[1] Fig. 2 shows E-value as a function of false positives per query (i/n, where i is the rank and n is the number of queries). Ideally, all points should lie near the line $x = y$ (middle line). In Fig. 2, all of the over 200,000 E-values correspond to false positive rates within a factor of 2 of the theoretical rate (outer lines). This shows that the E-values reported by POP are accurate.

Fig. 3 shows that the POP gives consistently better sensitivity than the MINP algorithm. In this cross-validated test, two family libraries are built by splitting each family in half. Families with only one member are removed because POP and MINP behave identically when there is only one sequence in the family definition. This leaves approximately 120 families in in each half-library. The effective family sizes are then estimated separately for each library using the ESIZE algorithm and a set of random queries. Sequences in one half-library are then used as queries against the other half-library. Following the methodology of Brenner *et al.* [1998], all the search results are sorted together by E-value. The figure shows, for each point in the sorted list, the fraction of queries (above that point) as a function of the number of false positives (above that point) divided by the number of queries. In this test, the POP algorithm consistently gives better sensitivity, or coverage, at a given error rate compared to the MINP algorithm. For example, for an error rate of one false positive per query, the coverage using POP is about 9% higher than using MINP. Since E-value is such a good predictor of false positive rate, as was shown in Fig. 2, the coverage at an E-value of 1.0 (or any other value) can be read directly from the plot in Fig. 3.

[1]The random queries are distinct from the ones used in computing the effective family sizes. For both sets of random sequences, the SEG algorithm [Wootton and Federhen, 1996] is used to remove regions with low information content after shuffling the letters in the sequences.

The superior classification performance of POP is further verified using ROC analysis. The area under the ROC curve ("ROC number") gives a picture of the complete tradeoff between sensitivity and selectivity, since it integrates the sensitivity of a search over the complete range of possible selectivity values. The ROC number is computed for each query using the same data as was used to construct Fig. 3. Pairwise comparisons of these ROC numbers using a two-tailed signed rank test [Snedecor and Cochran, 1980; Henikoff and Henikoff, 1997; Salzberg, 1997] rejects the null hypothesis that POP and MINP give statistically similar classification accuracy. The rank sum for POP is larger, and the test is significant at the 0.01 level. In these ROC comparisons, POP has higher ROC number for 140 queries and MINP for 92 queries.

The POP algorithm also has better classification performance compared to the MINP algorithm using PROSITE families to build the family library. We determined this by repeating the two cross-validated classification accuracy analyses described above using all 1338 families in the PROSITE [Bairoch, 1995] Release 15 database (data not shown). Using the same signed rank comparison of ROC numbers, classification performance of the POP algorithm was significantly better than that of MINP.

Using the p-value of the combined score in MINP effectively normalizes for the different number of members in each family. This is not done in the MINP algorithm, so one would expect erroneous matches to large families to contribute strongly to the error rate. We also explored adding the calculation of p-values to MINP to see if this would improve its classification accuracy. A modified MINP algorithm, PMINP (p-value of the minimum p-value), calculates the p-value of the minimum p-value, M_n, of the match between the query and the n family members using the equation

$$Pr(M_n \leq p) \quad = \quad 1 - (1 - p)^n. \tag{5}$$

Fig. 4, which was constructed analogously to Fig. 2, verifies that this gives accurate p-values in practice. The sensitivity-selectivity tradeoff improves relative to MINP as measured by plotting coverage as a function of false-positives per query (data not shown). ROC analysis of PMINP shows that POP is still significantly better, this time at a significance level of better than 10^{-7} according to the signed rank test. It is clear, therefore, that the differences between POP and MINP are not merely due to the effect of normalizing for family size.

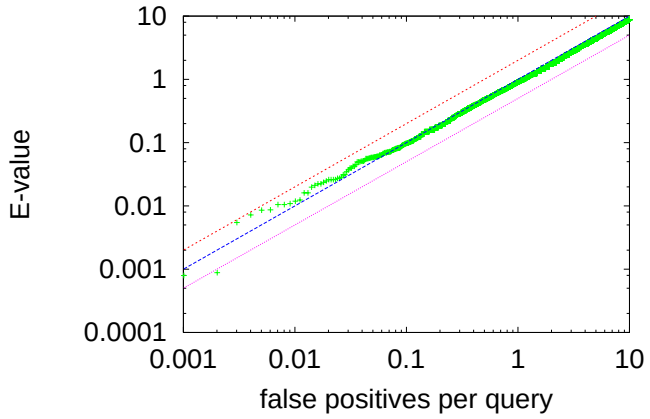

Figure 4: **Accurate prediction of false positive rates by PMINP E-values.** The plot shows the relationship between E-value and false positives per query for 1000 random sequence queries against all SCOP families with two or more members. Central line is the ideal, upper and lower lines are factors of two away.

4 Discussion

The principal result of this work is an algorithm for searching a library of proteins using a single sequence as a query. This algorithm, POP, combines the information in each of the matches between known members of a family and the query by taking the product of the p-values of those matches. This method takes into account the similarity of the query and each family member listed in the library entry for the family. The intuition is that distant, true homologs should have smaller p-value products than queries that have merely chance similarities to a single family member. This is borne out by the improvement in sensitivity versus selectivity of POP compared with MINP, which uses the p-value of the single best match as the match score. In addition, POP is likely to be far less susceptible to falsely classified family members in the family library than "best match" algorithms like MINP.

Previous work [Grundy, 1998b] studied several variants of family pairwise search algorithms in the mode of searching using a single family as the query against a database of sequences. The POP algorithm can also be used in this mode. It should be possible to modify the ESIZE algorithm to compute the effective size of the query family "on the fly" from the low-scoring sequences in the database being searched.[2] The results of the pairwise similarity searches of the database using each sequence in the query would be combined and input to ESIZE to determine the parameter of the product of p-values distribution. This parameter and the pairwise results would then be input to the POP algorithm.

The same previous work showed that family pairwise search algorithms in the family versus sequence mode (rather than the sequence versus family library mode discussed here) have better classification performance than several common model-based methods such as hidden Markov models (HMMs) [Eddy, 1995] and motifs [Bailey and Gribskov, 1998]. We expect these results to hold for POP as well, since it combines scores similarly to one of these versions of FPS that averages the "bit scores" [Altschul *et al.*, 1997] of the individual matches. Bit scores (normalized for the lengths of the query and target sequences) are related to p-values by the equation

$$Pr(S \geq s) \quad \approx \quad 2^{-S}, \tag{6}$$

where S is a length-normalized bit score. The relationship between POP p-values and average bit scores can then be seen from

$$\frac{1}{n}\sum_{i=1}^{n} s_i \quad \approx \quad -\log_2 \left[\left(\prod_{i=1}^{n} p_i \right)^{1/n} \right],$$

where s_i is the bit score for the match between the query and the ith member of the n-member family, and p_i is the p-value of that score. Thus, the FPS algorithm using average bit scores sorts matches to the query (approximately) by the geometric mean of the pairwise match p-values. This is the same as POP when all the members of the family are identical (Eqn. 2) and gives excessively large p-values in all other cases. FPS using average bit scores should therefore be less effective at utilizing all of the independent information present in the members of most families.

It should not be inferred from the better classification performance of the POP algorithm that model-based methods are not useful. Classification is only one benefit of building and using statistical models of sequence families. Other benefits include the localization and illumination of highly conserved, functionally and/or structurally important patterns in in the sequences. Furthermore, motif and HMM search algorithms identify the presence or absence as well as the location and spacing of these features of interest in the query sequence. HMM methods can also be used to multiply align the query and the sequences in the family, giving further insight into the relationship between the query and the family.

Another significant result of this work is a method for estimating the distribution of the product of non-independent p-values. This method may find use in other applications where it is desired to combine evidence from several correlated sources. The parameter of this distribution is also interesting because of its interpretation as the effective number of independent p-values. We have interpreted it here as the effective size of the family, ranging from 1 to the actual number of sequences in the set, depending on the degree of divergence (independent information) among the sequences. This might be useful in other contexts, such as clustering, since it gives a measure of the homogeneity or diversity of a set of sequences.

The datasets and implementations of the algorithms described in this paper are available at URL:

ftp://ftp.sdsc.edu/pub/sdsc/biology/fps/pop.

[2] This would be analogous to how p-values are computed for some pairwise similarity search algorithms such as FASTA [Pearson, 1998].

Acknowledgments

Timothy L. Bailey is supported by the National Biomedical Computation Resource, an NIH/NCRR funded research resource (P41 RR-08605), NBCR grant (NIH P41 RR-08605) and the NSF through cooperative agreement ASC-02827. William Grundy is supported by a Sloan/DOE postdoctoral fellowship in computational molecular biology.

References

[Altschul *et al.*, 1997] Stephen F. Altschul, Thomas L. Madden Alejandro A. Schäffer, Jinhui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman. Gapped-BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.

[Bailey and Gribskov, 1998] Timothy L. Bailey and Michael Gribskov. Combining evidence using p-values: application to sequence homology searches. *Bioinformatics*, 14:48–54, 1998.

[Bairoch, 1995] Amos Bairoch. The PROSITE database, its status in 1995. *Nucleic Acids Research*, 24:189–196, 1995.

[Brenner *et al.*, 1998] Steven E. Brenner, Cyrus Chothia, and Tim Hubbard. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proceedings of the National Academy of Sciences, USA*, 95:6073–6078, 1998.

[Compugen, Ltd., 1998] Compugen, Ltd. BIOXL/P Manual. `http://www.compugen-us.com`, 1998.

[Eddy, 1995] Sean R. Eddy. Multiple alignment using hidden Markov models. In C. Rawlings et al., editor, *Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology*, pages 114–120, Menlo Park, California, 1995. AAAI Press.

[Gribskov and Robinson, 1996] Michael Gribskov and Nina L. Robinson. The use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Computers and Chemistry*, 20:25–33, 1996.

[Grundy, 1998a] W. N. Grundy. Family-based homology detection via pairwise sequence comparison. In S. Istrail, P. Pevzner, and M. Waterman, editors, *Proceedings of the Second Annual International Conference on Computational Molecular Biology*, pages 94–100, New York, 1998. ACM.

[Grundy, 1998b] W. N. Grundy. Homology detection via Family Pairwise Search. *Journal of Computational Biology*, to appear, 1998.

[Henikoff and Henikoff, 1997] Steven Henikoff and Jorja G. Henikoff. Embedding strategies for effective use of information from multiple sequence alignments. *Protein Science*, 6:1–8, 1997.

[Murzin *et al.*, 1995] Alexey G. Murzin, Steven E. Brenner, Tim Hubbard, and Cyrus Chothia. SCOP: A structural classification of proteins database for the investigation of sequences and structures. *Journal of Molecular Biology*, 247:536–540, 1995.

[Pearson, 1998] William R. Pearson. Empirical statistical estimates for sequence similarity searches. *Journal of Molecular Biology*, 276:71–84, 1998.

[Press *et al.*, 1986] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes*. Cambridge University Press, Cambridge, England, 1986.

[Salzberg, 1997] S. L. Salzberg. On comparing classifiers: pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, 1:371–328, 1997.

[Smith and Waterman, 1981] Temple Smith and Michael Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[Snedecor and Cochran, 1980] George W. Snedecor and William G. Cochran. *Statistical Methods*. Iowa State University Press, Iowa, 1980.

[Wootton and Federhen, 1996] J. C. Wootton and S. Federhen. Analysis of compositionally biased regions in sequence databases. *Methods in Enzymology*, 266:554–571, 1996.