BIOINFORMATICS ORIGINAL PAPER Vol. 26 no. 14 2010, pages 1708–1713
doi:10.1093/bioinformatics/btq270

Sequence analysis

Advance Access publication May 26, 2010

Threshold Average Precision (TAP-*k*): a measure of retrieval designed for bioinformatics

Hyrum D. Carroll[1], Maricel G. Kann[2], Sergey L. Sheetlin[1] and John L. Spouge[1,*]

[1]National Center for Biotechnology Information, Bethesda, MD 20894 and [2]University of Maryland, Baltimore County, Baltimore, MD 21250, USA

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: Since database retrieval is a fundamental operation, the measurement of retrieval efficacy is critical to progress in bioinformatics. This article points out some issues with current methods of measuring retrieval efficacy and suggests some improvements. In particular, many studies have used the pooled receiver operating characteristic for n irrelevant records (ROC_n) score, the area under the ROC curve (AUC) of a 'pooled' ROC curve, truncated at n irrelevant records. Unfortunately, the pooled ROC_n score does not faithfully reflect actual usage of retrieval algorithms. Additionally, a pooled ROC_n score can be very sensitive to retrieval results from as little as a single query.

Methods: To replace the pooled ROC_n score, we propose the Threshold Average Precision (TAP-k), a measure closely related to the well-known average precision in information retrieval, but reflecting the usage of E-values in bioinformatics. Furthermore, in addition to conditions previously given in the literature, we introduce three new criteria that an ideal measure of retrieval efficacy should satisfy.

Results: PSI-BLAST, GLOBAL, HMMER and RPS-BLAST provided examples of using the TAP-k and pooled ROC_n scores to evaluate sequence retrieval algorithms. In particular, compelling examples using real data highlight the drawbacks of the pooled ROC_n score, showing that it can produce evaluations skewing far from intuitive expectations. In contrast, the TAP-k satisfies most of the criteria desired in an ideal measure of retrieval efficacy.

Availability and Implementation: The TAP-k web server and downloadable Perl script are freely available at http://www.ncbi.nlm.nih.gov/CBBresearch/Spouge/html.ncbi/tap/

Contact: spouge@ncbi.nlm.nih.gov

Supplementary Information: Supplementary data are available at *Bioinformatics* online.

Received on February 19, 2010; revised on April 28, 2010; accepted on May 19, 2010

1 INTRODUCTION

In bioinformatics, retrieval from databases is a fundamental operation. Therefore, progress depends on being able to recognize superior retrieval algorithms, so the measurement of retrieval efficacy is critical in bioinformatics. Swets (1967) stated that an ideal measure of retrieval efficacy (or more simply, a 'retrieval measure') should satisfy four conditions:

(1) It should concern itself solely with the effectiveness of separating the relevant from the non-relevant [records] and not with the efficiency of resource use.

(2) It should not be dependent on a [user] threshold, but should measure the potential output of the method.

(3) It should be a single number.

(4) It should have absolute significance as a measure of a single method and should readily allow comparisons of different methods to decide which is best.

To fix our terms, when a user queries a database of R records, a retrieval algorithm typically lists up to R records, ranked by some score S indicating the probability that the corresponding record has relevance to the query. In text retrieval (e.g. Google or PubMed results), retrieval lists do not indicate the scores producing their list orders. In a significant break with the traditions of information retrieval, however, bioinformatics retrieval often explicitly presents an E-value with the score, so users are free to choose an E-value threshold E_0 and then ignore the retrieval list beyond E_0. For concreteness, we discuss only E-values, but the methods in this article apply to any score S. (Note that E-values and the retrieval ranks increase together.)

In accord with the motivation behind E-values, Wilbur (1992) modified Swets' Condition (2):

(2′) It should be characterized by a [user] threshold, but should reflect the quality of retrieval at every rank down to that threshold.

Wilbur's modification implicitly respects an overarching principle governing retrieval measures, which we call 'the Principle of Fidelity': a retrieval measure should faithfully reflect the actual usage of the retrieval list. If not, the measure might be 'ideal' in some abstract sense, but would lack a practical meaning.

The Principle of Fidelity supports Wilbur's Condition (2′) in bioinformatics, because an E-value threshold E_0 influences the actual usage of a retrieval algorithm. Since a user rarely examines a retrieval list far beyond the E-value threshold E_0, any practical measure of database retrieval in bioinformatics should reflect the user's E_0. The rest of this article, therefore, disregards Swets' Condition (2) in favor of Wilbur's Condition (2′), referring to the result as the 'Swets–Wilbur Conditions'.

*To whom correspondence should be addressed.

In addition to the Swets–Wilbur Conditions, the Principle of Fidelity suggests that an ideal retrieval measure should satisfy additional conditions. Accordingly, we introduce the following conditions:

(5) It should be robust against results representing a small proportion of possible user queries.

(6) When two disjoint sets of queries are considered, its value for the union of the two sets should lie between its values for the two sets of queries.

(7) It should reflect the choice of threshold; in particular, it should eventually decrease as the threshold increases to include the entire retrieval list.

Condition (5) reflects the fact that not many users are likely to query with the proportionally small subset, so the subset should not greatly influence conclusions about retrieval efficacy. Condition (6) says that when combined, two disjoint sets of retrieval results should not suggest better (or worse) efficacy than either set individually. Condition (7) reflects the fact that presumably, an appropriate E-value threshold E_0 has practical utility: if users prefer to examine the entire retrieval list, they have no use for an E-value threshold.

The receiver operating characteristic for n irrelevant records (ROC_n) score (Gribskov and Robinson, 1996) described in Section 2 is often used as a retrieval measure in bioinformatics. In fact, the 'pooled ROC_n score' (Schaffer *et al.*, 1999) (also described in Section 2) is probably the most popular summary retrieval measure over several different queries. However, the Principle of Fidelity casts immediate suspicion on the pooled ROC_n score as a retrieval measure. Users do not examine the pooled retrieval list aggregated from lists for individual queries: users see the individual retrieval lists one at a time.

Section 3 shows that the pooled ROC_n score does not always satisfy Condition (5) or (6). Moreover, it always fails to satisfy Condition (7). To replace the pooled ROC_n score, we therefore propose as a measure of retrieval the Threshold Average Precision at a median of k errors per query (EPQ) (abbreviated and pronounced 'TAP-k'). The TAP-k summarizes features of the precision–recall (PR) curve (described in Section 2). PR curves are popular in general information retrieval and already have found some favor in bioinformatics (Chen, 2003; Jones *et al.*, 2005; Krishnamurthy *et al.*, 2007; Raychaudhuri *et al.*, 2002; Wass and Sternberg, 2008).

To exemplify the PR curve and TAP-k, Section 3 presents several examples of actual database retrieval, using the programs PSI-BLAST (Schaffer *et al.*, 2001), GLOBAL (Kann *et al.*, 2007), HMMER (Eddy, 1998) and RPS-BLAST (Schaffer *et al.*, 1999). The section shows that unlike the TAP-k, the pooled ROC_n score can produce evaluations so misleading as to be completely contrary to common sense (Chen, 2003; Hand, 2009; Sierk and Pearson, 2004). Finally, Section 4 summarizes the implications of our results.

2 METHODS

2.1 Databases and query sets

We used two distinct databases in this work (see Supplementary Material). First, Gonzalez and Pearson (2010) constructed DB_344_Pfam, 344 protein families from the Pfam database (Finn *et al.*, 2008). As sample queries for DB_344_Pfam, they provided 50 randomly selected families, each with a 'query A', from a deserted part of each family's phylogenetic tree; and a 'query B', from a heavily populated part. Gonzalez and Pearson considered as 'relevant' only sequences in the same domain family or clan as the query.

Second, Kann *et al.* (2007) provided DB_331_CDD, the position-specific scoring matrices (PSSMs) corresponding to 331 multiple sequence alignments from the NCBI Conserved Domain Database (CDD; Marchler-Bauer *et al.*, 2007). As sample queries for DB_331_CDD, they provided DB_8920_PDB [which Kann *et al.* (2007) named 'DB_10185', for the 10 185 PDB sequences it contained before additional filtering]. DB_8920_PDB contains 8920 non-redundant sequences from the RCSB Protein Data Bank (PDB; Berman *et al.*, 2007). Kann *et al*. considered as 'relevant' only to those sequences in DB_8920_PDB that had at least 80% overlap with a representative in DB_331_CDD.

2.2 Retrieval programs

Retrieval with PSI-BLAST (version 2.2.21) provided our anecdotal examples. We performed five PSI-BLAST iterations on NCBI's NR database with an E-value threshold of 0.005, using the final PSSM to retrieve sequences from DB_344_Pfam. Estimates of retrieval efficacy reflected solely the final retrieval from DB_344_Pfam, not the previous iterations on the NR database.

Additionally, we calculated retrieval results for GLOBAL, HMMER and RPS-BLAST with the DB_8920_PDB queries searching in the DB_331_CDD. We utilized two variants of HMMER: HMMER_semi-global and HMMER_local (HMMER in 'global' and 'local' modes, respectively). The settings for HMMER, along with their rationale, have been specified elsewhere (Kann *et al.*, 2007).

2.3 Retrieval measures

2.3.1 The ROC_n curve and score Given a particular query, assume every database record is either relevant or irrelevant to the query. [The standard ROC terminology refers to 'true positives' and 'false positives', but in information retrieval, the terms 'relevant' and 'irrelevant' are pertinent. Unlike some authors (Hand, 2009), we view information retrieval as a problem in ranking, not a problem in classification.] Let the total number of irrelevant records be F. In response to a query, a retrieval algorithm produces a ranked retrieval list of all records in the database. Number each irrelevant record in the database $1, 2, \ldots, f, \ldots, F$, according to its order in the retrieval ranking. The 'ROC curve' plots the fraction of relevant records preceding the f-th irrelevant record against the fraction f/F. The 'ROC score' is the area under the ROC curve, abbreviated 'AUC' (Swets, 1988). The ROC score is the probability that a random relevant record is ranked before a random irrelevant record (Bamber, 1975). By analogy to the ROC curve, the 'ROC_n curve' is the ROC curve truncated after the first n irrelevant records, with the ROC_n score being the area under the ROC_n curve divided by n/F. An 'ideal retrieval' ranks all relevant records before any irrelevant record. The normalization by n/F ensures that ideal retrieval receives the maximum ROC_n score of 1.0. For the ROC_n, a threshold of $n = 50$ irrelevant records seems accepted practice (Gribskov and Robinson, 1996).

2.3.2 The pooled ROC_n score To calculate the pooled ROC_n score, merge the retrieval lists for all sample queries into a 'pooled retrieval list', and sort the pooled list on the E-value. Then, calculate the ROC_n score for the pooled list, as though it were a single retrieval list.

2.3.3 The PR curve and the average precision PR curves and average precision (AP) often quantify retrieval efficacy in general information retrieval. To calculate the AP (see Supplementary Material), fix the retrieval algorithm A and consider a particular query q to a fixed database. Let the database contain $T(q)$ records relevant to the query$q$, and let them be ranked $t_1 < \ldots < t_{T(q)}$ in the retrieval list for algorithm A. (Thus, for ideal retrieval, $t_i = i$ for $i = 1, \ldots, T(q)$.) Let $p(j)$ denote 'precision', defined as the fraction j/t_j of relevant records in the retrieval list up to and including the j-th relevant record. (The precision is, therefore, one minus the false discovery rate, i.e.

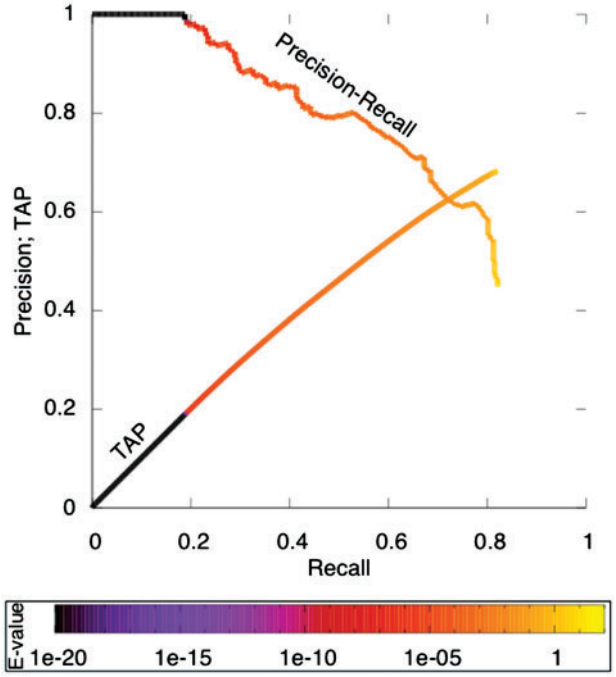

Fig. 1. An example of a PR graph and TAP curve. The E-values at each point are represented by the colors on the bar beneath.

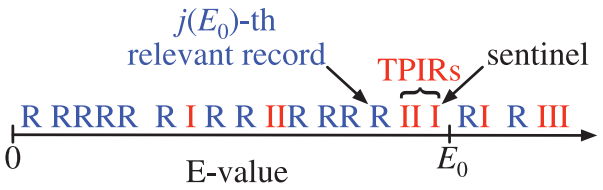

Fig. 2. Example retrieval list with relevant (blue 'R's) and irrelevant (red 'I's) records illustrating the $j(E_0)$-th relevant record, TPIRs and the sentinel record.

it is a true positive rate.) Also, let $r(j)$ denote 'recall', the fraction $j/T(q)$ of relevant records up to and including the j-th relevant record. The PR curve plots $p(j)$ against $r(j)$ (Fig. 1).

2.3.4 The TAP-k We now design a retrieval measure reflecting the usage of E-values in bioinformatics. Let E_0 be an arbitrary E-value threshold. For the query q, define $j(E_0)$ to be the number of relevant records in the retrieval list with an E-value less than or equal to the threshold E_0. Consider the 'terminal pre-threshold irrelevant records' (TPIRs), the irrelevant records retrieved after the $j(E_0)$-th relevant record but having an E-value less than or equal to E_0 (Fig. 2). Call the last record with an E-value less than or equal to E_0 the 'sentinel' record. Regardless of whether or not the sentinel is relevant, it is associated with a precision $p(E_0)$, where $p(E_0)$ is the fraction of records preceding or including the sentinel that are relevant. (If there are no records before the E-value threshold E_0, define $p(E_0)=0$.) The following measure captures the effect of both post-threshold relevant records and TPIRs:

$$\overline{p}(E_0;q)=\frac{1}{T(q)+1}\left[\sum_{m=1}^{j(E_0)}p(m)+p(E_0)\right] \quad (1)$$

To summarize Equation (1), it assigns the post-threshold relevant records a precision of 0, considers the precision at the sentinel record, and then averages the precision of the pre-threshold relevant, sentinel and post-threshold relevant records. (If $j(E_0)=0$, there are no relevant records before the threshold, and we adopt the standard convention that empty sums equal 0, so Equation (1) yields the value 0.)

To measure the overall retrieval efficacy for several sample queries, the simplest and most intuitive aggregate measure is $\overline{p}(E_0)$, the average of the TAP, $\overline{p}(E_0;q)$, over all queries (Chen, 2003). Query averages are easy to interpret, and if usage favors certain types of queries, the average can be weighted, e.g. linearly or quadratically with the number of proteins in a family (Green and Brenner, 2002).

Now, we determine an E-value threshold E_0 mirroring a user's tolerance for retrieval errors. Assume (as the ROC_n score does) that a user tolerates about k EPQ, k being some arbitrary integer. Section 3, e.g. gives $k=20$ as an arbitrary but not unreasonable estimate of a (maximum) tolerable EPQ. Determine the smallest E-value $E_k(A)$ corresponding to a median number of k EPQ over all queries q for a given algorithm A. Thus, for any E-value threshold larger than $E_k(A)$, at least 50% of the queries have at least k errors. (Section 3 explains why the median EPQ is preferable to the mean EPQ.) Each algorithm's E-value predicts the actual number of EPQ with varying accuracy, so the threshold $E_k(A)$ depends on the algorithm A. With the same median k EPQ, all algorithms have the same specificity. With their specificities fixed at the same value, their sensitivities are on an equal footing, and therefore comparable.

In summary, our measure of overall retrieval efficacy is $\overline{p}_k=\overline{p}_k(A)$, the (query-averaged) TAP-k for a median k EPQ (the 'TAP-k'), i.e. it is the average over all queries of Equation (1) with $E_0=E_k(A)$.

2.4 Software availability

We implemented Equation (1) in a Perl script and provided a web interface to calculate the TAP-k for a set of retrieval lists. Both are easy to use, return results quickly, and are freely available at http://www.ncbi.nlm.nih.gov/CBBresearch/Spouge/html.ncbi/tap/.

3 RESULTS

In this section, we compare the TAP-k to the pooled ROC_n score. First, we examine the effect of using the median EPQ versus using the mean EPQ. Next, we show how a single query can skew the pooled ROC_n score. Then, we present an example of calculating the E-value threshold for k median EPQ. Finally, we show how varying the E-value threshold affects the TAP-k.

3.1 Median EPQ versus mean EPQ

To illustrate the compelling reasons why the mean EPQ is inferior to the median EPQ when analyzing database retrieval, consider a PSI-BLAST retrieval from DB_344_Pfam. Figure 3 displays a box-and-whisker plot of the distribution over queries of the EPQ against E-value. At an E-value of 0.01, only about 10% of queries produce any retrieval errors at all (data not shown), although the mean EPQ is already about 2. Thus, the mean EPQ of 2 reflects a definite minority of the queries that users encounter. As an extreme hypothetical example, if a single query out of 1×10^6 possible queries produced 2×10^6 false positives, and all other queries had perfect retrieval, the mean EPQ would still be 2.0, although few users indeed would encounter any retrieval errors. In contrast, the median EPQ is 0.0, accurately representing 99.9999% of the queries. The Principle of Fidelity, therefore, favors the median EPQ over the mean EPQ, because it reflects a user's typical experience more closely.

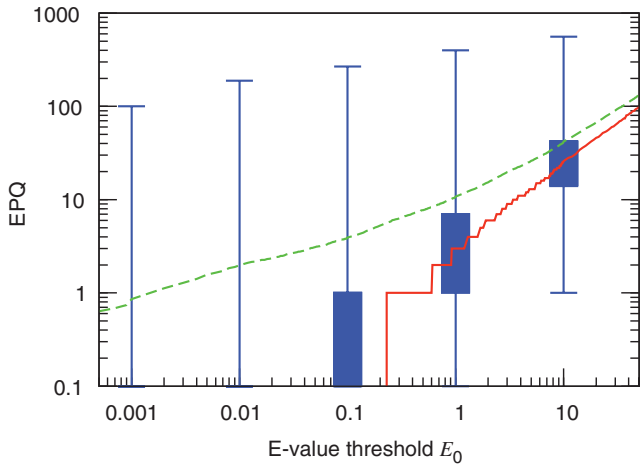

Fig. 3. The distribution of EPQ versus E-value for PSI-BLAST retrieval over all queries in DB_344_Pfam. The dashed green line indicates the mean EPQ; the solid red line, the median EPQ; the top and bottom of the blue boxes, the first and third quartiles of the EPQ distribution; and the top and bottom whiskers, the maximum and minimum EPQ over all queries.

3.2 The pooled ROC$_n$

To illustrate the counter-intuitive behavior of the pooled ROC$_n$ score when measuring retrieval efficacy over several queries, consider Figure 4a. It displays ROC$_{50}$ curves corresponding to three retrieval lists from the Pfam homoserine dehydrogenase family, two for the single queries A and B of the homoserine dehydrogenase family in Pfam, and one for the corresponding pooled retrieval list for A and B together. The retrieval list for query A has a ROC$_{50}$ score of 0.971, close to ideal retrieval, ranking all but 14 out of the 481 relevant records before any irrelevant record. On the other hand, the retrieval list for query B has a ROC$_{50}$ score of 0.195, because it ranks only 94 of the 481 relevant records ahead of the corresponding irrelevant records. All initial $n = 50$ irrelevant records in the retrieval list of query B have lower E-values than any relevant records in the retrieval list for query A, with the 50-th record for query B having an E-value of 1×10^{-134}. Since query B has small E-values that appear early in its retrieval, it dominates the values of the pooled ROC$_{50}$. Because pooling the two queries doubles the number of relevant records, the pooled ROC$_n$ score is only 0.098, half the minimum ROC$_n$ of the two queries.

We attempted to remedy the counter-intuitive behavior of the ROC$_n$ score by truncating the retrieval lists for queries A and B at the E-value threshold E_{20}(PSI-BLAST) (data not shown). For every $n \leq 203$, however, the pooled ROC$_n$ curve still places an exaggerated emphasis on query B and its ineffective retrieval.

3.3 The calculation of the threshold $E_k(A)$

To illustrate the mechanics of determining the threshold $E_k(A)$ for different algorithms, consider the median EPQ at the E-value threshold $E_0 = E_k(A)$ corresponding to $k = 20$ median EPQ. The threshold $E_k(A)$ depends on the algorithm A, e.g. E_{20}(GLOBAL) = 66.5, E_{20}(HMMER_semi-global) = 82.7, E_{20}(HMMER_local) = 39.7 and E_{20}(RPS-BLAST) = 79.4. In actual usage, if users tolerate k EPQ and limit the EPQ by learning the E-value threshold $E_k(A)$, the Principle of Fidelity indicates that different algorithms A should be compared at different E-values $E_k(A)$.

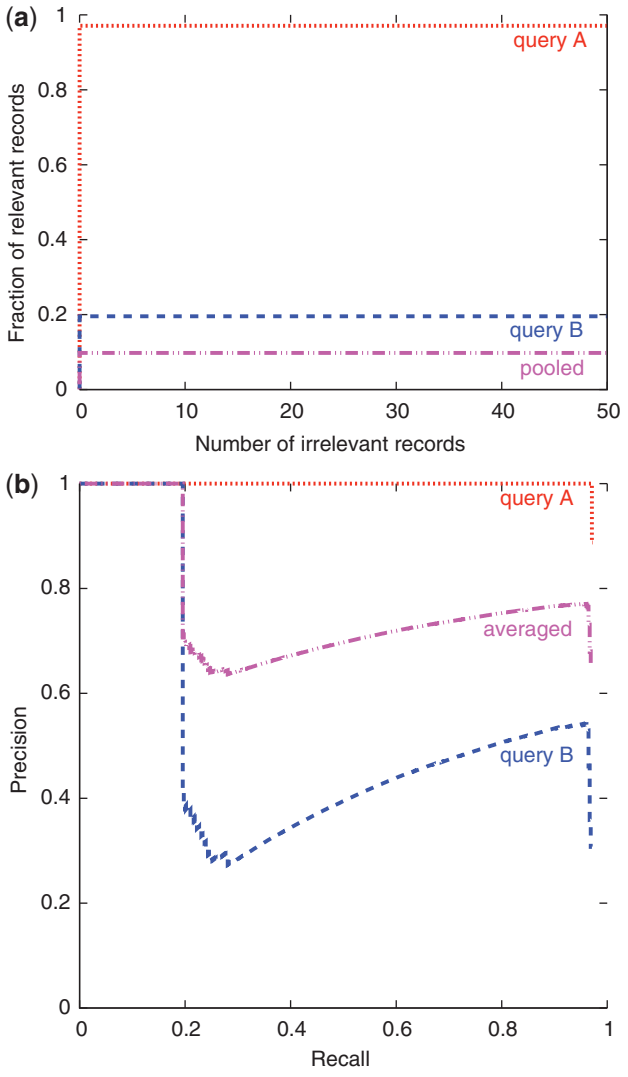

Fig. 4. PSI-BLAST retrieval results for the homoserine dehydrogenase Pfam family searching in DB_344_Pfam. (**a**) Individual ROC$_{50}$ curves, along with the corresponding pooled ROC$_{50}$. Note that the pooled ROC curve is lower than both of the queries. This same condition continues until 203 irrelevant records. (**b**) PR curves (and their average) for the same retrieval results. The TAP for each is the AP (with the precision of last record repeated).

3.4 The query-averaged TAP-k

Figure 4b illustrates the PR curves for the Pfam homoserine dehydrogenase family, up to the recall corresponding to the threshold E-value E_{20}(PSI-BLAST) = 8.1, where PSI-BLAST yields 20 median EPQ over all queries for DB_344_Pfam. On one hand, query A yields nearly ideal retrieval, and there is little difference between the ROC$_n$ score and the TAP-k. On the other hand, query B yields precision 1.0 until a recall of about 0.2, when the precision drops dramatically. It then rises and falls again before the recall near 1 corresponding to E_{20}(PSI-BLAST) = 8.1. Although the curve for query B indicates that its retrieval is inferior to the retrieval for

Table 1. Retrieval results for DB_331_CD

Algorithm	TAP-20	E_0*(A)	Peak TAP
GLOBAL	0.164	1.034	0.227
HMMER_semi-global	0.185	0.861	0.237
HMMER_local	0.152	0.116	0.221
RPS-BLAST	0.142	0.331	0.218

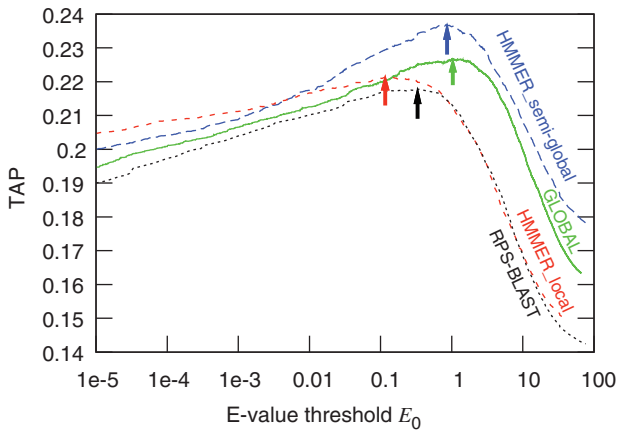

Fig. 5. TAP curves against the E-value threshold E_0, for searching DB_331_CDD with each query from DB_8920_PDB in turn. Retrieval results for GLOBAL are represented with a solid green line; for HMMER_semi-global, with a long-dashed blue line; for HMMER_local, with a medium-dashed red line; and for RPS-BLAST, with a dotted black line. The arrows indicate the maximum TAP and its E-value threshold E_0*(A) for each algorithm A.

query A, the average of the curves for queries A and B lies between the individual curves, as one expects intuitively.

Using the threshold $E_k(A)$ for $A = \{$GLOBAL, HMMER_semi-global, HMMER_local, RPS-BLAST$\}$, we calculated the (query-averaged) TAP-k for each algorithm's retrieval from DB_331_CDD (Table 1). (On one hand, because the number of errors for a single query is unbounded, the median is better than the mean as a summary statistic for determining a threshold E-value; on the other hand, the TAP-k for each query is bounded, so the mean is a suitable summary statistic for the TAP-k over all queries.) Additionally, we looked at the average TAP versus E-value (Fig. 5), because the TAP may peak as the EPQ increases. In contrast, all ROC curves increase (or at least remain constant) with increasing EPQ. In Figure 5, each algorithm A has a peak E-value, E_0*(A), between 0.12 and 1.03 (Table 1).

4 DISCUSSION

This article is not the first to question the pertinence of ROC analysis to information retrieval (Chen, 2003; Hand, 2009; Pearson and Sierk, 2005; Sierk and Pearson, 2004). In fact, many other researchers have pointed out the superiority of PR curves over ROC curves in information retrieval. Fawcett (2006) advises that, 'PR graphs are commonly used where "the number of [irrelevant records] is many orders of magnitude greater than [the number of relevant records]" ', the common case for database retrieval and most of bioinformatics. Likewise, Liu and Shriberg (2007) suggest that for 'an imbalanced dataset, PR curves generally provide better visualization than do ROC curves, for viewing differences among different algorithms.' Similarly, Davis and Goadrich (2006) warn that 'with highly skewed datasets, PR curves give a more informative picture of an algorithm's performance' and that, 'by comparing false positives to true positives rather than true negatives, [precision] captures the effect of the large number of negative examples on the algorithm's performance.' Finally, Landgrebe *et al.* (2006) argue that ROC analysis effectively ignores the 'minority class' of relevant records.

Several bioinformatics studies have relied on ROC analysis as their figure of merit for automatic improvement of database retrieval algorithms. As a figure of merit, however, the pooled ROC_n score suffers from defects so obvious that other bioinformatics studies (wisely, in our opinion) have gone so far as to defend conclusions drawn from the pooled ROC_n score by checking individual retrieval lists (Sierk and Pearson, 2004). Clearly, if the defects of the pooled ROC_n score require human intervention, it is inadequate to the task of automated improvement of retrieval algorithms. To provide explicit logical foundations for the discussion about the inadequacies of retrieval measures, this article also articulated a Principle of Fidelity: a retrieval measure should faithfully reflect the actual usage of the retrieval list. In harmony with the Principle of Fidelity, we suggested adding Conditions (5)–(7) to the Swets–Wilbur Conditions for an ideal retrieval measure.

The Section 3 demonstrates that the pooled ROC_n can violate Conditions (5) and (6). Condition (6) is common sense, so its failure in Figure 4 is particularly disturbing. On the other hand, since the TAP-k is an average over all queries, Conditions (5) and (6) both follow as rigorous mathematical truths. Moreover, consideration of the geometry of a ROC curve shows that the ROC_n always increases with the EPQ (or equivalently, with the E-value threshold), in violation of Condition (7). On the other hand, the TAP-k sometimes does satisfy Condition (7): Figure 5 shows its eventual decrease against an increasing E-value threshold E_0. Interestingly, the peak values in Figure 5 occur at values of E_0*(A) not entirely outside acceptable ranges of the E-value thresholds for the corresponding algorithms, perhaps leading to the hope that even the selection of threshold E-values E_0 might be automated. Unfortunately, the Supplementary Material gives an example of retrieval lists where the TAP-k increases monotonically with the E-value threshold. Although the TAP-k has many properties desirable to optimizing retrieval algorithms automatically, it is currently unable to serve as a basis for automated determination of a best E-value threshold E_0*(A).

For concreteness, this article has discussed E-values, but in fact it is pertinent to any score S, not just to E-values. The E-value is really just a type of score that retains theoretical meaning across different queries as a surrogate for a record's probability of relevance. Like other scores, however, E-values do not predict the relevancy of records with complete accuracy, and the accuracy depends very much on the application (Brenner *et al.*, 1998). Thus, if a particular algorithm A produces a retrieval list, a user willing to tolerate about a median k EPQ must apparently learn the corresponding E-value threshold $E_0 = E_k(A)$ by empirical experience. Initially, it might appear counter-intuitive that the E-value threshold $E_0 = E_k(A)$ depends on the algorithm A, but the dependency does reflect actual usage of the algorithm. This article approximated actual usage by specifying a median EPQ of $k = 20$, but the measure of tolerated

EPQ can and should be adapted to fit individual needs, e.g. k can be chosen differently, query-averages can be weighted, trimmed means or a different percentile EPQ from the median for k can be used, etc.

The ROC_n also depends implicitly on the algorithm A, because it fixes the total number n of errors across all queries. Thus, where the TAP-k fixes the median EPQ, the ROC_n fixes the mean EPQ. In general (particularly for unbounded random variables), the mean can be much more misleading as a measure of central tendency than the median. In particular, if it produces many errors, even a single query could increase the mean EPQ arbitrarily. Figures 3 and 4 reinforce the superiority of choosing the median EPQ in the TAP-k, by showing that a single query can dominate the pooled ROC_n score. By extension, coverage versus EPQ plots (Brenner *et al.*, 1998) could reflect typical user experience more closely by plotting coverage against median EPQ, rather than mean EPQ.

Figure 4 illustrates the same retrieval results for both the pooled ROC_n score and a TAP-k. The pooled ROC_n score and the TAP-k agree (0.9709 versus 0.9708, respectively) for query A, but differ for query B (0.1954 and 0.5214, respectively). The pooled ROC_n score is 0.098, whereas the TAP-k is 0.745. Thus, besides giving some numerical comparison of the pooled ROC_n and TAP-k, Figure 4 illustrates that the TAP-k faithfully represents the relative contribution of 'ill-behaved' queries to a summary measure of retrieval over all queries.

In general, we expect that studies would usually draw the same conclusions about relative retrieval efficacy of different algorithms, regardless of whether they used the pooled ROC_n score or the TAP-k (although the TAP-k enforces realistic E-value thresholds by exposing its threshold $E_0 = E_k(A)$ explicitly). Davis and Goadrich (2006) noted similar expectations between AP and ROC scores. In cases of striking discordance, however, this article presents compelling arguments that the TAP-k is more likely than the pooled ROC_n score to accord with intuitive expectations, and that its use will make measurements of retrieval efficacy reflect actual user experience more faithfully. Most importantly, unlike the pooled ROC_n, the TAP-k always satisfies Conditions (5) and (6) for an ideal retrieval measure, so it can provide a suitable figure of merit when automating the evaluation of retrieval algorithms.

ACKNOWLEDGEMENTS

The authors would like to thank Stephen Altschul, John Wilbur and Anna Panchenko for reviewing the manuscript; the anonymous reviewers for their helpful suggestions; Mileidy Gonzalez for sharing her database of domain families; and Eva Czabarka for initiating discussions on the Principle of Fidelity.

Funding: National Institutes of Health (NIH) (1K22CA143148 to M.G.K.); Intramural Research Program of the NIH, National Library of Medicine (in part).

Conflict of Interest: none declared.

REFERENCES

Bamber,D. (1975) Area above ordinal dominance graph and area below receiver operating characteristic graph. *J. Math. Psychol.*, **12**, 387–415.

Berman,H. *et al*. (2007) The worldwide Protein Data Bank (wwPDB): ensuring a single, uniform archive of PDB data. *Nucleic Acids Res.*, **35**, D301–303.

Brenner,S.E. *et al*. (1998) Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proc. Natl Acad. Sci. USA*, **95**, 6073–6078.

Chen,Z. (2003) Assessing sequence comparison methods with the average precision criterion. *Bioinformatics*, **19**, 2456–2460.

Davis,J. and Goadrich,M. (2006) The Relationship Between Precision-Recall and ROC Curves. In *Proceedings of the 23rd International Conference on Machine learning*. ACM, Madison, Wisconsin, USA, pp. 233–240.

Eddy,S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755–763.

Fawcett,T. (2006) An introduction to ROC analysis. *Pattern Recognit. Lett.*, **27**, 861–874.

Finn,R.D. *et al.* (2008) The Pfam protein families database. *Nucleic Acids Res.*, **36**, D281–D288.

Gonzalez,M. and Pearson,W. (2010) Homologous over-extension: a challenge for iterative similarity searches. *Nucleic Acids Res.*, **38**, 2177–2189.

Green,R.E. and Brenner,S.E. (2002) Bootstrapping and normalization for enhanced evaluations of pairwise sequence comparison. *Proc. IEEE*, **90**, 1834–1847.

Gribskov,M. and Robinson,N.L. (1996) Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput. Chem.*, **20**, 25–33.

Hand,D.J. (2009) Measuring classifier performance: a coherent alternative to the area under the ROC curve. *Mach. Learn.*, **77**, 103–123.

Jones,C.E. *et al.* (2005) Automated methods of predicting the function of biological sequences using GO and BLAST. *BMC Bioinformatics*, **6**, 272.

Kann,M.G. *et al.* (2007) The identification of complete domains within protein sequences using accurate E-values for semi-global alignment. *Nucleic Acids Res.*, **35**, 4678–4685.

Krishnamurthy,N. *et al.* (2007) FlowerPower: clustering proteins into domain architecture classes for phylogenomic inference of protein function, *BMC Evol. Biol.*, **7** (Suppl. 1), S12.

Landgrebe,T.C.W. *et al.* (2006) Precision-recall operating characteristic (P-ROC) curves in imprecise environments. In *Proceedings of 18th International Conference on Pattern Recognition*, IEEE Computer Society, Los Alamitos, California, USA, pp. 123–127.

Liu,Y. and Shriberg,E. (2007) Comparing valuation metrics for sentence boundary detection. *IEEE Int Conf. Acoust. Speech Signal Process.*, 185–188.

Marchler-Bauer,A. *et al.* (2007) CDD: a conserved domain database for interactive domain family analysis. *Nucleic Acids Res.*, **35**, D237–D240.

Pearson,W.R. and Sierk,M.L. (2005) The limits of protein sequence comparison? *Curr. Opin. Struct. Biol.*, **15**, 254–260.

Raychaudhuri,S. *et al.* (2002) Associating genes with gene ontology codes using a maximum entropy analysis of biomedical literature. *Genome Res.*, **12**, 203–214.

Schaffer,A.A. *et al.* (1999) IMPALA: matching a protein sequence against a collection of PSI-BLAST-constructed position-specific score matrices. *Bioinformatics*, **15**, 1000–1011.

Schaffer,A.A. *et al.* (2001) Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Res.*, **29**, 2994–3005.

Sierk,M.L. and Pearson,W.R. (2004) Sensitivity and selectivity in protein structure comparison. *Protein Sci.*, **13**, 773–785.

Swets,J.A. (1967) *Effectiveness of Information Retrieval Methods*. Bolt, Beranek, and Newman, Inc., Cambridge, MA.

Swets,J.A. (1988) Measuring the accuracy of diagnostic systems. *Science*, **240**, 1285–1293.

Wass,M.N. and Sternberg,M.J.E. (2008) ConFunc - functional annotation in the twilight zone. *Bioinformatics*, **24**, 798–806.

Wilbur,W.J. (1992) An information measure of retrieval performance. *Inf. Syst.*, **17**, 283–298.