Improving Retrieval Efficacy of Homology Searches using the False Discovery Rate

Hyrum D. Carroll, Alex C. Williams, Anthony G. Davis, and John L. Spouge

Abstract—Over the past few decades, discovery based on sequence homology has become a widely accepted practice. Consequently, comparative accuracy of retrieval algorithms (*e.g.*, BLAST) has been rigorously studied for improvement. Unlike most components of retrieval algorithms, the E-value threshold criterion has yet to be thoroughly investigated. An investigation of the threshold is important as it exclusively dictates which sequences are declared relevant and irrelevant. In this paper, we introduce the false discovery rate (FDR) statistic as a replacement for the uniform threshold criterion in order to improve efficacy in retrieval systems. Using NCBI's BLAST and PSI-BLAST software packages, we demonstrate the applicability of such a replacement in both non-iterative (BLAST$_{FDR}$) and iterative (PSI-BLAST$_{FDR}$) homology searches. For each application, we performed an evaluation of retrieval efficacy with five different multiple testing methods on a large training database. For each algorithm, we choose the best performing method, Benjamini-Hochberg, as the default statistic. As measured by the Threshold Average Precision, BLAST$_{FDR}$ yielded 14.1% better retrieval performance than BLAST on a large (5,161 queries) test database and PSI-BLAST$_{FDR}$ attained 11.8% better retrieval performance than PSI-BLAST. The C++ source code specific to BLAST$_{FDR}$ and PSI-BLAST$_{FDR}$ and instructions are available at http://www.cs.mtsu.edu/∼hcarroll/blast_fdr/.

Index Terms—Homology search, false discovery rate, retrieval efficacy, uniform E-value thresholding

1 Introduction

In response to a query, many database search algorithms (*e.g.*, BLAST and PSI-BLAST [1]) return a retrieval list of sequences sorted by the E-values assigned to each sequence. Typically, each E-value is calculated from a statistical model of irrelevant ("false positive") database sequences and approximates the expected number of irrelevant sequences with a score equal to or better than the one calculated. Many algorithms truncate their retrieval lists at a uniform E-value threshold. We call this truncation procedure "uniform E-value thresholding". While many different aspects of BLAST have undergone rigorous examination, uniform E-value thresholding has not had the same scrutiny.

This article studies thresholding procedures in two programs for protein sequence retrieval: BLAST and PSI-BLAST. BLAST accepts a sequence as a query to search for relevant ("true positive") matches in a specified database. Additionally, an E-value threshold may be supplied to BLAST. BLAST looks for all relevant matches between that query and the sequences in a database and then applies uniform E-value thresholding by ignoring all matches with an E-value above the specified value.

PSI-BLAST is an iterative version of BLAST, which takes a single protein sequence query and database as inputs. Its first iteration is the same as a BLAST search. At the end of that iteration and each subsequent one, it performs uniform E-value thresholding on the retrieval list (at a stringent default E-value threshold of 0.002). Furthermore, it aligns the truncated list against the original query, and generates a position-specific scoring matrix (PSSM) from the alignment to search the database in the next iteration. The default E-value threshold for entry into the PSI-BLAST alignment is stringent, to prevent an excess of irrelevant sequences ("false positives") from overwhelming the query sequence and "corrupting" the search [2].

As computing potential and the sophistication of computer algorithms increase, so has the need to account for multiple testing. For both non-iterative and iterative homology searches, the query is compared against each sequence in the database independently, resulting in multiple tests. Performing multiple tests can give the perception of a more significant result than the data can support. False discovery rate (FDR) methods aim to control the proportion of irrelevant matches to address the issues introduced by multiple testing. They are widely used in microarray studies and virtually in all facets of genomic studies. Unfortunately, few have adopted their use for sequence analysis. A recent exception to this is the use of a FDR approach to aid in generating the DFam database [3].

Early efforts for managing the false positive rate aimed to control the Family-wise Error Rate (FWER), the likelihood of making one or more false discoveries. Due to the intrinsic nature of how the FWER is computed, FWER methods also provide control over the FDR. Four modern and traditionally-accepted FWER methods are

- *H.D. Carroll, A.C. Williams and A.G. Davis are with the Department of Computer Science, Middle Tennessee State University, Murfreesboro, TN, 37128.*
E-mail: Hyrum.Carroll@mtsu.edu, acw4a@mtmail.mtsu.edu, agd2q@mtmail.mtsu.edu
- *J.L. Spouge is at the National Center for Biotechnology Information, Bethesda, MD 20894.*

the Bonferroni correction [4], the Holm step-up procedure [5], the Hochberg step-down procedure [6], and the Hommel single-wise procedure [7]. The Bonferroni correction uses a uniform P-value threshold determined by a user-specified α (or P-value threshold) divided by the total number of performed tests. The Holm step-up procedure extends the Bonferroni correction by adding the rank of the ordered P-values to the total number of performed tests in the thresholding method. Like the Holm procedure, the Hochberg step-down process utilizes the rank in the thresholding method by looking for the P-value that is less than a user-specified α divided by the total number of performed tests in addition to the current P-value's rank. The Hommel single-wise procedure is similar in that it looks for the P-value for which all P-values with a higher rank are greater than a number proportional to α. In comparison with FWER methods, procedures designed to control only the FDR, such as the Benjamini-Hochberg step-up procedure [8], offer a less conservative form of measurement in exchange for greater control over the number of relevant and irrelevant sequences. The Benjamini-Hochberg method computes a threshold by multiplying the current P-value's rank by a user-specified α and dividing the result by the total number of performed tests.

In this paper, we explore the retrieval efficacy (how well a method identifies relevant records) of two applications: BLAST$_{FDR}$ and PSI-BLAST$_{FDR}$, each of which is a BLAST variant that uses E-values to calculate the FDR. We demonstrate that both applications perform better than their predecessors (BLAST and PSI-BLAST), in part by drastically decreasing the number of irrelevant sequences. The Methods section presents the implementation details of each application; the Results section describes our testing procedures and their results. We conclude with a discussion of BLAST$_{FDR}$ and PSI-BLAST$_{FDR}$'s applicability. The C++ source code specific to BLAST$_{FDR}$ and PSI-BLAST$_{FDR}$, instructions and supplementary material are available at http://www.cs.mtsu.edu/~hcarroll/blast_fdr/.

2 Methods

BLAST$_{FDR}$ extends version 2.2.27 of NCBI's BLAST algorithm by replacing uniform E-value thresholding with one of the following algorithms: Bonferroni, Holm's step-down process, Hochberg's step-down process, Hommel's single-wise process, and Benjamini and Hochberg's method. The Bonferroni method calculates a threshold value for each sequence retrieved and considers the first k ranked sequences as significant that satisfy the following criterion: $P_k \leq \frac{\alpha}{m}$, where P_k is the P-value of the k^{th} sequence and m is the size of the database searched. Because BLAST relies heavily on E-values instead of P-values, and given that E-value = P-value * m [9], we implemented the Bonferroni method as: $E_k \leq \alpha$ with E_k being the E-value of the k^{th} sequence. Furthermore, the Holm method considers matches significant that meet the following criterion: $E_k \leq \frac{m\alpha}{m+1-k}$. Similarly, the Hochberg method takes a different approach by starting at the least likely match and working toward the best statistical score to consider the following matches as significant: $E_k \leq \frac{m\alpha}{m+1-k}$. The Hommel method also iterates from the least significant match to find the index k such that: $E_{m-k+j} > \frac{j\alpha}{k}$ for $j = 1, \ldots, k$, then uses k to consider the following matches significant: $E_k \leq \frac{m\alpha}{k}$. Finally, the Benjamini-Hochberg method iterates from the match with the best statistical score and uses the following criterion for significant matches: $E_k \leq k\alpha$.

Each match in BLAST is called a high scoring pair (HSP). A database sequence can have multiple HSPs. BLAST organizes all of the HSPs according to the database sequence to which they belong and maintains its internal data structures sorted by the best HSP per database sequence. This is problematic for applying the methods above. Consequently, BLAST$_{FDR}$ and PSI-BLAST$_{FDR}$ restructure the HSPs from being sorted by sequence to being sorted by individual scores before applying the threshold. The new list stores pointers to the original data structures, minimizing the amount of memory required.

To determine retrieval efficacy for BLAST$_{FDR}$ and PSI-BLAST$_{FDR}$, we leveraged the query sequences in the ASTRAL40 database [10]. Each sequence in the ASTRAL40 database has less than 40% sequence identity to the other sequences. More importantly, each sequence has been classified into a "superfamily". We only considered the queries that have at least one other superfamily member in the database. Matches with the sequences in the same superfamily are considered relevant matches. To avoid making erroneous assignments, we ignore matches that are not in the same superfamily as the query sequence. For irrelevant matches, we augmented this database 100-fold with random sequences drawn from the distribution of amino acids residues and length of sequences found in the original ASTRAL40 database. We partitioned the augmented database into Training and Test databases. We sorted the queries by name, and assigned the 5,162 sequences with an odd rank to the Training database and the 5,161 sequences with an even rank to the Test database [11]. Additionally, we randomly selected 103 queries (2%) from the training dataset to use to evaluate which method to use. We refer to this subset as "Training-subset".

For PSI-BLAST and PSI-BLAST$_{FDR}$, each query is first searched against a large non-redundant database. For this study, we clustered NCBI's NR database to 90% sequence identity (NR90) by selecting a representative sequence for each cluster with nrdb90 [12]. After at most five iterations of searching on the NR90 database, the resulting PSSM was used to search against the augmented ASTRAL40 databases.

Traditionally, the Receiver Operating Characteristic (ROC$_n$) method [13] has served as an evaluation criterion for retrieval efficacy. The ROC$_n$ method ignores the threshold implied by a homology search algorithm and truncates a list of matches after the n^{th} irrelevant match. The resulting list of matches is plotted with the number of irrelevant matches on the x-axis and the proportion of relevant matches on the y-axis. A ROC$_n$ score is then the normalized area under the curve. Typically, $n = 50$. The ROC$_n$ method was not suitable for this study as it generally requires the threshold imposed by the algorithm to be artificially modified to allow for n irrelevant matches, thus erasing the effect of the threshold method.

In this study, we utilize the Threshold Average Precision (TAP) [14] method as the evaluation criterion for retrieval efficacy. The TAP method calculates the median Average Precision-Recall with a moderate adjustment for irrelevant sequences just before the threshold. TAP values range from 0.0 for a retrieval with no relevant sequences to 1.0 for a search that retrieves all of the relevant sequences and only relevant sequences.

Here, we use a slightly simplified calculation of the TAP value because each program uses its own retrieval threshold. We calculate TAP values according to equation 1:

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

where q is a query, $T(q)$ is the total number of relevant records for query q, $p(x)$ is the precision at record x, and j is the last record retrieved.

We choose the TAP measure because it fulfills the conditions for an ideal measure of retrieval efficacy proposed by Swets [15] and Wilbur [16]:

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

Other retrieval measures, such as the tuple of precision and recall, fail to met the criterion of using a single number. While the average precision is a single number, it fails the second criterion in that irrelevant records at the very end of the retrieval do not affect the score.

To determine the best performing threshold method to use, we examined the retrieval performance for each one of them with $\alpha = \{0.0005,\ 0.005,\ 0.05,\ 0.5\}$ using the Training-subset database. From these methods, we adopted the best performing one as the default threshold method in BLAST$_{FDR}$ and PSI-BLAST$_{FDR}$. We then evaluated that method with $\alpha = \{0.0005, 0.005, 0.05, 0.5\}$ using the entire Training database. Finally, the best performing method with the best performing value of α was compared against BLAST and PSI-BLAST using the Test database.

TABLE 1
Average BLAST$_{FDR}$ TAP values using the Training-subset database

	α			
Method	0.0005	0.005	0.05	0.5
Bonferroni	0.163	0.170	0.198	0.199
Holm	0.163	0.170	0.198	0.199
Hochberg	0.081	0.088	0.102	0.150
Hommel	0.163	0.170	0.198	0.199
Benjamini-Hochberg	0.168	0.180	0.203	0.184

TABLE 2
Average BLAST$_{FDR}$ TAP values using the Training database

	α			
Method	0.0005	0.005	0.05	0.5
Benjamini-Hochberg	0.199	0.215	0.229	0.220

3 Results

To evaluate the performance of BLAST$_{FDR}$ and PSI-BLAST$_{FDR}$, we performed several experiments involving five different threshold methods to account for multiple testing. We utilized an augmented version of the ASTRAL40 database (see the Methods section). We measured the performance in terms of the Threshold Average Precision (TAP) value.

First, we evaluated BLAST$_{FDR}$ with the following methods for determining the threshold for matches: Bonferroni correction, Holm step-up procedure, Hochberg step-down procedure, Hommel single-wise procedure and Benjamini-Hochberg. For each method, we set $\alpha = \{0.0005, 0.005, 0.05, 0.5\}$ on the Training-subset database (see Table 1). Of these methods, BLAST$_{FDR}$ with the Benjamini-Hochberg method received the best average TAP value of 0.203 and generally performed better than the other methods. Consequently, we adopted this method as the default for BLAST$_{FDR}$. For comparison purposes, BLAST received an average TAP value of 0.171 on the same database using the default E-value threshold of 10.

On the (full) Training database, we evaluated the same four α values for BLAST$_{FDR}$ using the Benjamini-Hochberg method (see Table 2). Of these parameters, BLAST$_{FDR}$ with $\alpha = 0.05$ received the best average TAP of 0.229 while BLAST received 0.203. Consequently, we adopted this α level as the default for BLAST$_{FDR}$.

We evaluated the efficacy of BLAST and BLAST$_{FDR}$ using the 5,161 query sequences in the Test database.

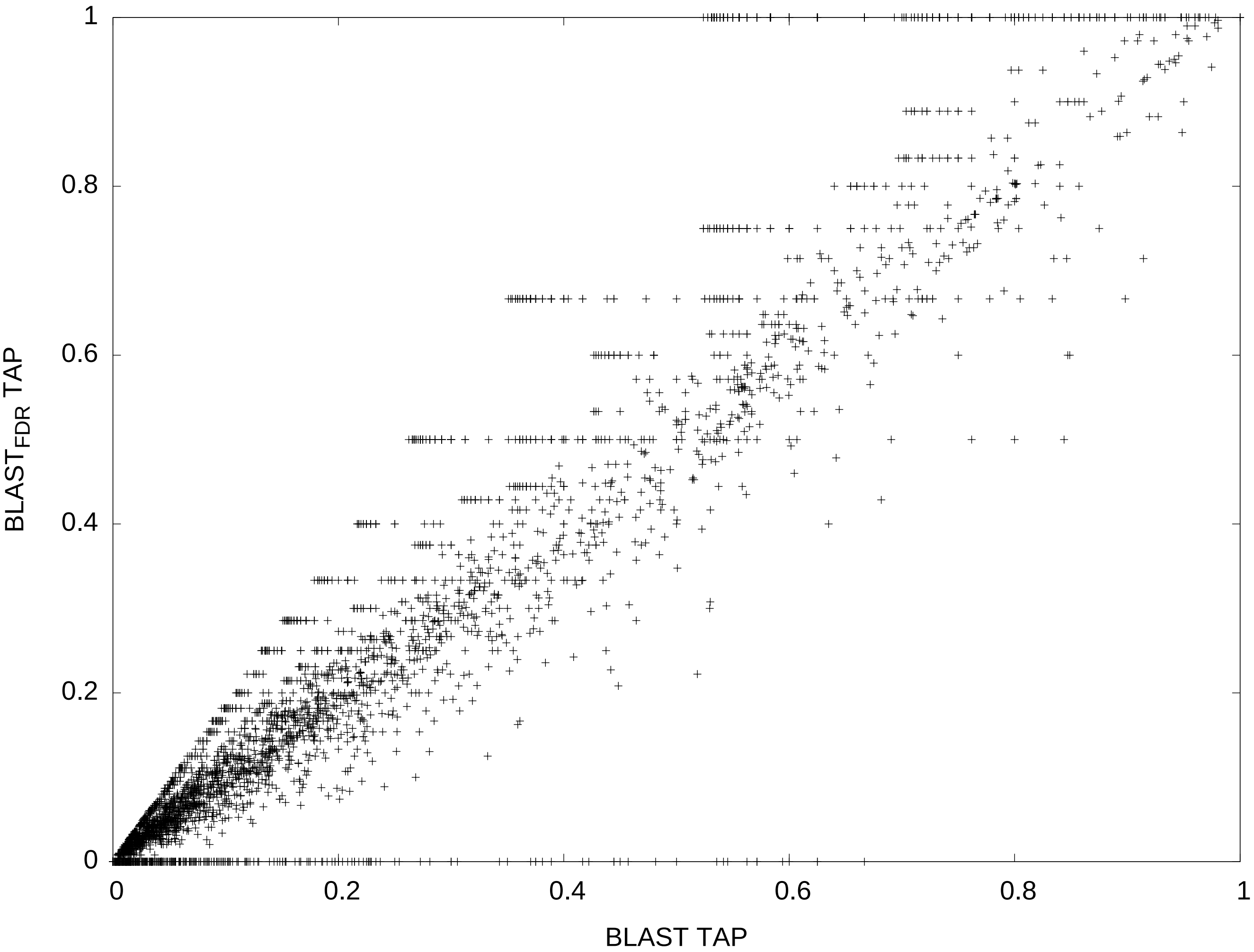

Fig. 1. TAP results for every query in the Test database for BLAST and BLAST$_{FDR}$

TABLE 3
Average TAP values for BLAST and BLAST$_{FDR}$

Database	BLAST	BLAST$_{FDR}$
Training-subset	0.171	0.203
Training	0.203	0.229
Test	0.198	0.226

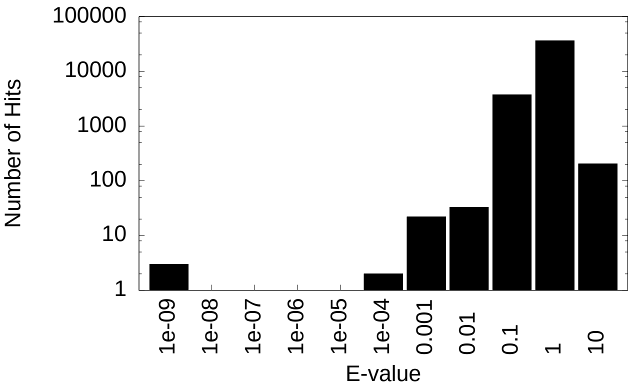

Fig. 2. Histogram of the E-values of sequences in the Test database retrieved by BLAST but not by BLAST$_{FDR}$

Table 3 summarizes the results and Figure 1 details the TAP values for BLAST plotted against the TAP values for BLAST$_{FDR}$ for each of the queries. While BLAST received an average TAP value of 0.198, BLAST$_{FDR}$ earned an average TAP value of 0.226. In terms of irrelevant sequences, BLAST$_{FDR}$ retrieves an average of only 0.27 irrelevant sequences per query whereas BLAST retrieves 2,780% more with 7.44 per query. For every dataset in the Test database, the retrieval list for BLAST$_{FDR}$ was shorter than the respective list for BLAST. This is noticeable in Figure 1 as "lines" for BLAST$_{FDR}$ TAP values. For BLAST, it retrieves more, and in particular a more varied number of irrelevant sequences (typically at the end of the retrieval), resulting in a wider distribution of TAP values. Finally, Figure 2 is a histogram of the E-values of sequences retrieved by BLAST that were not retrieved by BLAST$_{FDR}$.

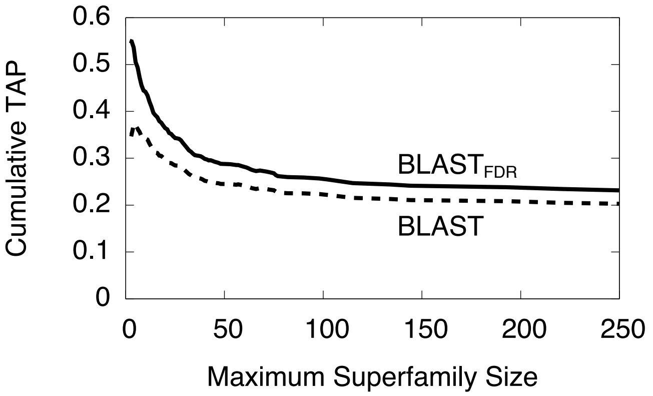

Fig. 3. Cumulative BLAST$_{FDR}$ TAP and BLAST TAP versus aggregate superfamily size for the Test database

TABLE 4
Average TAP values for BLAST using the Test database

	E-value Threshold						
	1e-5	1e-4	0.001	0.01	0.1	1	10
BLAST	0.170	0.184	0.198	0.210	0.223	0.223	0.198

Furthermore, BLAST$_{FDR}$ performs notably better on datasets that belong to small superfamilies. Figure 3 illustrates this with the cumulative average TAP for both BLAST$_{FDR}$ and BLAST for ascending superfamily sizes. For example, for superfamilies with a size of twelve or fewer members, BLAST$_{FDR}$ has a TAP of 0.421 and BLAST a TAP of 0.332.

We also evaluated the retrieval performance of BLAST using the Test database for the following E-value thresholds: 1e-5, 1e-4, 0.001, 0.01, 0.1, 1 and 10 (the default). Table 4 reports the average TAP value for each of these thresholds. BLAST performed the best with an E-value threshold of 0.1 with an average TAP value of 0.223.

We also evaluated PSI-BLAST$_{FDR}$ in the same manner as above using the Training-subset database by using the same five methods and the same values of α (see Table 5). Iterating on the NR90 database first and then searching on the ASTRAL40 database noticeably increases the TAP value for each algorithm. Again we observe that Bonferroni, Holm and Hommel reported identical values due to their similar algorithms. The ordering of the methods is the same as with BLAST$_{FDR}$, consequently, we again adopted the Benjamini-Hochberg method as the default for PSI-BLAST$_{FDR}$ and set the default for the α parameter to 0.05 as well. With these parameters, PSI-BLAST$_{FDR}$ has a TAP value of 0.332. For comparison, PSI-BLAST received a TAP value of 0.296 on the same databases using the default E-value thresholds.

We also evaluated the efficacy of PSI-BLAST and PSI-BLAST$_{FDR}$ using the Training database (5,162 query sequences) and Test database (5,161 query sequences) (see Table 6). While PSI-BLAST received an average TAP value of 0.346 on the Training database, PSI-BLAST$_{FDR}$ earned an average TAP value of 0.385. Additionally, for the Test database, PSI-BLAST received an average TAP of 0.338 and PSI-BLAST$_{FDR}$ and average TAP value of 0.378. Furthermore, to visualize the results of each query in the Test database, each TAP value for PSI-BLAST is plotted against the respective PSI-BLAST$_{FDR}$ TAP value in Figure 4. In terms of irrelevant sequences, PSI-BLAST$_{FDR}$ retrieves an average of only 1.07 irrelevant sequences per query whereas PSI-BLAST retrieves 12.62 per query (1183% more). Finally, Figure 5 is a histogram of the E-values of sequences retrieved by PSI-BLAST that were not retrieved by PSI-BLAST$_{FDR}$. For every dataset in the Test database, the retrieval list for PSI-BLAST$_{FDR}$ was shorter than the respective list for PSI-BLAST.

TABLE 5
Average PSI-BLAST$_{FDR}$ TAP values using the Training-subset database

	α			
Method	0.0005	0.005	0.05	0.5
Bonferroni	0.302	0.319	0.327	0.323
Holm	0.302	0.319	0.327	0.323
Hochberg	0.215	0.225	0.257	0.318
Hommel	0.302	0.319	0.327	0.323
Benjamini-Hochberg	0.309	0.329	0.332	0.303

TABLE 6
Average TAP values for PSI-BLAST and PSI-BLAST$_{FDR}$

Database	PSI-BLAST	PSI-BLAST$_{FDR}$
Training-subset	0.296	0.329
Training	0.346	0.385
Test	0.338	0.378

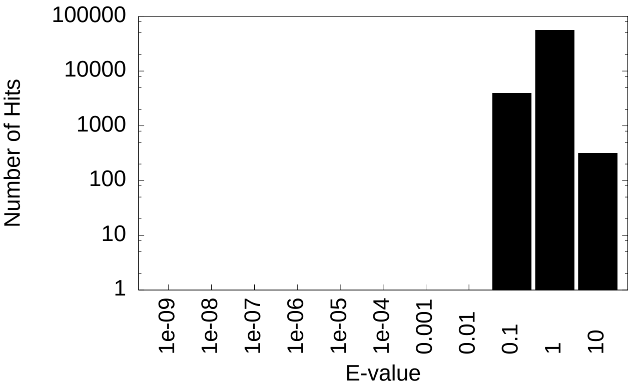

Fig. 5. Histogram of the E-values of sequences in the Test database retrieved by PSI-BLAST but not by PSI-BLAST$_{FDR}$

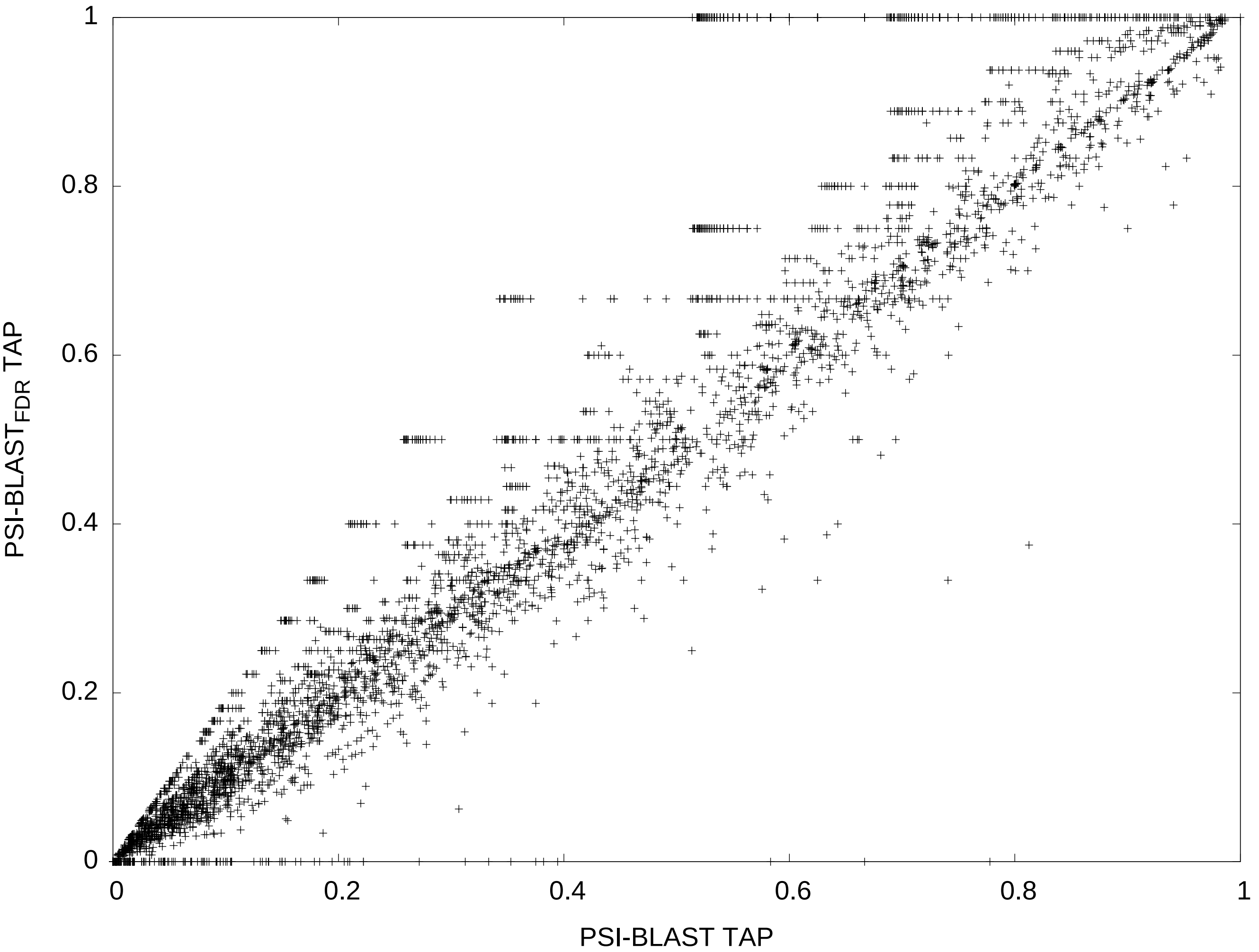

Fig. 4. TAP results for every query in the Test database for PSI-BLAST and PSI-BLAST$_{FDR}$

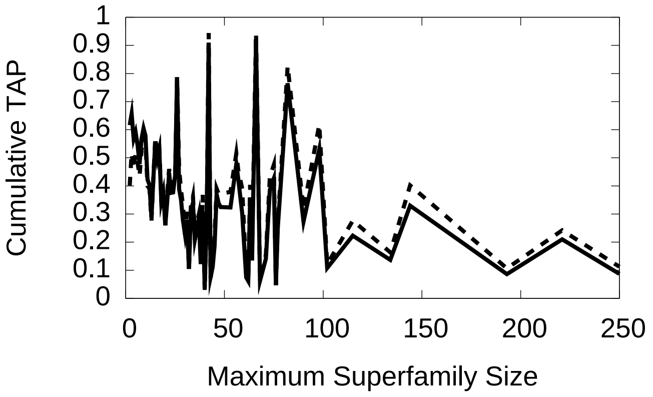

Fig. 6. Cumulative PSI-BLAST$_{FDR}$ TAP and PSI-BLAST TAP versus aggregate superfamily size for the Test database. PSI-BLAST$_{FDR}$ is the solid line and PSI-BLAST the dashed line

As with BLAST$_{FDR}$ and BLAST, PSI-BLAST$_{FDR}$ performs notably better than PSI-BLAST on datasets that belong to small superfamilies. Figure 6 illustrates this with the cumulative average TAP for both PSI-BLAST$_{FDR}$ and PSI-BLAST for ascending superfamily sizes. For the iterative methods, the results vary greatly per superfamily size for medium and large sized superfamilies.

TABLE 7
Average TAP values for PSI-BLAST using the Test database

	E-value Threshold						
	1e-5	1e-4	0.001	0.01	0.1	1	10
PSI-BLAST	0.320	0.332	0.346	0.358	0.371	0.363	0.338

Finally, we evaluated the effects of truncating the retrieval of PSI-BLAST using the Test database for the following E-value thresholds: 1e-5, 1e-4, 0.001, 0.01, 0.1, 1 and 10 (the default). Table 7 reports the average TAP value for each of these thresholds. PSI-BLAST performed the best truncating the E-value threshold at 0.1 with an average TAP value of 0.371.

4 Discussion

In this article we discussed an observed deficiency in the control of the proportion of irrelevant records in retrieval algorithms. Including too many irrelevant sequences has been shown to corrupt searches in a genetic database search algorithm [2]. For iterative algorithms like PSI-BLAST, this corruption is propagated and magnified with each iteration. To address this issue, we propose BLAST$_{FDR}$ and PSI-BLAST$_{FDR}$, each of which is an implementation of their predecessor that exercises a false discovery rate method, for finer control over the percentage of irrelevant sequences.

To establish default parameters for BLAST$_{FDR}$ and PSI-BLAST$_{FDR}$, we evaluated the following thresholding methods: Bonferroni correction, Holm step-up procedure, Hochberg step-down procedure, the Hommel single-wise procedure and Benjamini-Hochberg step-up procedure with $\alpha = \{0.0005, 0.005, 0.05, 0.5\}$ for each method. The Benjamini-Hochberg method with $\alpha = 0.05$ performed the best. Interestingly, only the Benjamini-Hochberg method stops improving with relaxed restrictions (see Table 1), suggesting that the FDR provides an appropriate retrieval cut-off.

Using accepted evaluation procedures, BLAST$_{FDR}$ and PSI-BLAST$_{FDR}$ performed better than BLAST and PSI-BLAST respectively. For the ASTRAL40 Test datasets, BLAST$_{FDR}$ had an average TAP value that was 14.1% higher than BLAST and PSI-BLAST$_{FDR}$ had an average TAP value that was 11.2% better than PSI-BLAST. These differences are notable given the extremely wide use that BLAST and PSI-BLAST enjoy. Furthermore, BLAST$_{FDR}$ is particularly appropriate for queries with small superfamily sizes as evidenced by it obtaining an average TAP value 26.8% higher than BLAST for superfamilies with sizes up to and including twelve. The performance of PSI-BLAST$_{FDR}$ on the Test data sets was also best for queries that belong to smaller superfamilies. For queries in larger superfamilies, if the goal is to assign function to a query, then adequately identifying the superfamily is sufficient. For example, retrieving 50% of a large superfamily clearly indicates which superfamily the query belongs. This objective is not currently captured in retrieval evaluation metrics and may make evaluation values misleading for large superfamilies.

We also afforded BLAST and PSI-BLAST with the advantage of evaluating multiple threshold parameters. We truncated the retrieval lists for BLAST and PSI-BLAST for E-value $= \{1\text{e-}5, 1\text{e-}4, 0.001, 0.01, 0.1, 1, 10\}$ (where 10 is the default) (see Tables 4 and 7). For the Test database, BLAST$_{FDR}$ and PSI-BLAST$_{FDR}$ still performed better than BLAST and PSI-BLAST respectively at all threshold levels. BLAST and PSI-BLAST both performed best at an E-value of 0.1, but at this threshold BLAST retrieved 1,214 less relevant sequences than BLAST$_{FDR}$ and PSI-BLAST 3,932 less relevant sequences than PSI-BLAST$_{FDR}$.

While both BLAST$_{FDR}$ and PSI-BLAST$_{FDR}$ show noticeable performance improvements over BLAST and PSI-BLAST, the increases were not seen for all queries. For example, Figures 1 and 4 illustrate that there are several datasets in the Test database that BLAST$_{FDR}$ and PSI-BLAST$_{FDR}$ receive a TAP value of 0.0 but BLAST achieves a non-zero TAP value. Clearly some improvements can be made to these methods to improve their performance.

While we used BLAST and PSI-BLAST as examples in this study, other retrieval algorithms that use uniform thresholding could also benefit from the implementation of a FDR controlled threshold. Furthermore, employing more advanced false discovery rate methods, such as the Q-value method [17] could also yield improvements. Implementation of the Q-value, because it requires the entire distribution of statistical scores, is inherently challenging for heuristic algorithms like BLAST and PSI-BLAST.

References

[1] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs," *Nucleic Acids Research*, vol. 25, no. 17, pp. 3389–3402, 1997.

[2] M. Gonzalez and W. Pearson, "Homologous over-extension: a challenge for iterative similarity searches," *Nucleic acids research*, vol. 38, no. 7, pp. 2177–2189, 2010.

[3] T. J. Wheeler, J. Clements, S. R. Eddy, R. Hubley, T. A. Jones, J. Jurka, A. F. Smit, and R. D. Finn, "Dfam: a database of repetitive DNA based on profile hidden Markov models," *Nucleic acids research*, vol. 41 (D1), pp. D70–D82, 2013.

[4] C. E. Bonferroni, *Il calcolo delle assicurazioni su gruppi di teste*. Tipografia del Senato, 1935.

[5] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian journal of statistics*, pp. 65–70, 1979.

[6] Y. Hochberg, "A sharper Bonferroni procedure for multiple tests of significance," *Biometrika*, vol. 75, no. 4, pp. 800–802, 1988.

[7] G. Hommel, "A stagewise rejective multiple test procedure based on a modified Bonferroni test," *Biometrika*, vol. 75, no. 2, pp. 383–386, 1988.

[8] Y. Benjamini and Y. Hochberg, "Controlling the False Discovery Rate: a Practical and Powerful Approach to Multiple Testing," *Journal of the Royal Statistical Society, Series B*, vol. 57, pp. 289–300, 1995.

[9] M. G. Kann, S. L. Sheetlin, Y. Park, S. H. Bryant, and J. L. Spouge, "The identification of complete domains within protein sequences using accurate *E*-values for semi-global alignment," *Nucleic Acids Research*, vol. 35, no. 14, pp. 4678–4685, 2007.

[10] J. Chandonia, G. Hon, N. Walker, L. Lo Conte, P. Koehl, M. Levitt, and S. Brenner, "The ASTRAL Compendium in 2004," *Nucleic Acids Research*, vol. 32, no. Database Issue, pp. D189–D192, 2004.

[11] S. Altschul, E. Gertz, R. Agarwala, A. Schäffer, and Y. Yu, "PSI-BLAST pseudocounts and the minimum description length principle," *Nucleic Acids Research*, vol. 37, no. 3, pp. 815–824, 2009.

[12] L. Holm and C. Sander, "Removing near-neighbour redundancy from large protein sequence collections." *Bioinformatics*, vol. 14, no. 5, pp. 423–429, 1998.

[13] M. Gribskov and N. Robinson, "Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching," *Computers and Chemistry*, vol. 20, no. 1, pp. 25–33, 1996.

[14] H. D. Carroll, M. G. Kann, S. L. Sheetlin, and J. L. Spouge, "Threshold Average Precision (TAP-k): A Measure of Retrieval Efficacy Designed for Bioinformatics," *Bioinformatics*, vol. 26, no. 14, pp. 1708–1713, 2010.

[15] J. A. Swets, "Effectiveness of Information Retrieval Methods," Bolt, Beranek, and Newman, Inc., Cambridge, MA, 1967.

[16] W. J. Wilbur, "An information measure of retrieval performance," *Information Systems*, vol. 17, no. 4, pp. 283–298, 1992.

[17] J. Storey, “A direct approach to false discovery rates,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 64, no. 3, pp. 479–498, 2002.

Hyrum D. Carroll received his BSc degree in Computer Engineering and MSc and PhD degrees in Computer Science from Brigham Young University in 2002, 2004 and 2008 respectively. He was a postdoctoral researcher at the National Center for Biotechnology Information from 2008 to 2011. Since then he has been an Assistant Professor at Middle Tennessee State University. His research interests include homology search algorithms and computational biology algorithms that leverage next-gen sequence data.

Alex C. Williams received the BSc degree in Computer Science from Middle Tennessee State University. He is currently working toward the MSc degree with Dr. Hyrum Carroll. He is a graduate teaching assistant for the Department of Computer Science. He is also a national member of the ACM and chair of his local student chapter.

Anthony G. Davis received the BSc degree in Computer Science and is currently working toward the MSc degree in Computer Science from Middle Tennessee State University. He is currently a graduate teaching assistant for the Department of Computer Science at MTSU.

John L. Spouge received his BSc and MD degrees from the University of British Columbia, and his DPhil degree in Applied Probability at Trinity College, Oxford under John Hammersley in 1983. He was a post-doctoral fellow at Los Alamos National Laboratory and the National Institutes of Health before becoming a founding member of the National Center for Biotechnology Information in 1989. His research interests include sequence and structure statistics, HIV, DNA barcodes, and coalescent theory.