doi:10.1093/nar/gkp301

SIB-BLAST: a web server for improved delineation of true and false positives in PSI-BLAST searches

Marianne M. Lee[1], Michael K. Chan[1,2,3,*] and Ralf Bundschuh[1,2,4,*]

[1]The Ohio State Biophysics Program, [2]Department of Biochemistry, [3]Department of Chemistry and [4]Department of Physics, Ohio State University, Columbus, OH 43210-1117, USA

Received January 28, 2009; Revised April 6, 2009; Accepted April 15, 2009

ABSTRACT

A SIB-BLAST web server (http://sib-blast.osc.edu) has been established for investigators to use the SimpleIsBeautiful (SIB) algorithm for sequence-based homology detection. SIB was developed to overcome the model corruption frequently observed in the later iterations of PSI-BLAST searches. The algorithm compares resultant hits from the second iteration to the final iteration of a PSI-BLAST search, calculates the figure of merit for each 'overlapped' hit and re-ranks the hits according to their figure of merit. By validating hits generated from the last profile against hits from the first profile when the model is least corrupted, the true and false positives are better delineated, which in turn, improves the accuracy of iterative PSI-BLAST searches. Notably, this improvement to PSI-BLAST comes at minimal computational cost as SIB-BLAST utilizes existing results already produced in a PSI-BLAST search.

INTRODUCTION

Bioinformatics tools, in particular, those utilizing sequence comparison methods to search for homologs in order to obtain clues regarding functional and evolutionary relationships of uncharacterized proteins, have become an integral part of today's laboratory research. BLAST (1) is arguably the most ubiquitous sequence alignment tool for uncovering close homologs. PSI-BLAST (2,3), an extended version of BLAST, is similarly popular, and also more sensitive in detecting the harder-to-find distant homologs due to its iterative and profile-based search approach (4,5). However, in PSI-BLAST searches it is commonly observed that non-homologous proteins are incorporated into the profiles of the later iterations, leading to model corruption and meaningless results. For this reason, PSI-BLAST developers recommend users to confine their iterative search to no more than five or six rounds (3).

Numerous homology detection algorithms with different strategies to improve the discrimination of true and false positives have been developed. These include the state-of-the-art programs: SAM (6–10) and HMMER (7,11), which perform better than PSI-BLAST in remote homology detection (12), but are far less popular than PSI-BLAST due to their expensive computational costs. Thus, there is a trade-off between improved performance and computational efficiency. In light of these considerations, we have developed a novel algorithm, SimpleIsBeautiful (SIB) (13), that overcomes the model corruption problem in PSI-BLAST searches while at the same time, preserving PSI-BLAST's computational efficiency. By benchmarking resultant hits from the last iteration, where the algorithm has identified the most distant homologs, against resultant hits from the second iteration, when the profile is the least corrupted (since it is comprised mostly of close homologs), our SIB algorithm showed improved discrimination between true and false positives over standard PSI-BLAST searches. A direct performance comparison between the SIB algorithm and PSI-BLAST based on the same test set (Aravind dataset) (14) confirmed that SIB outperforms PSI-BLAST in terms of both specificity and sensitivity. Further performance comparison of SIB against another state-of-the-art sequence alignment algorithm SAM-T2K (8,10) using the same Aravind dataset revealed that SIB exhibits a comparable performance, but at a much lower computational costs.

The SIB algorithm has previously been made available as a downloadable awk script at http://bioserv.mps.ohio-state.edu/SimpleIsBeautiful. The script takes the outputs from a PSI-BLAST search (second iteration and final iteration), compares the two lists of hits and re-ranks the merged list of hits based on each hit's corresponding figure of merit (FOM) (13)—a numeric representation analogous to PSI-BLAST's *E*-value. Several requests have since been made for an implementation of a web interface of the SIB algorithm. In this article, we present the SIB-BLAST web server that performs PSI-BLAST searches, runs the latest version of the SIB algorithm,

*To whom correspondence should be addressed. Tel: +1 614 688 3978; Fax: +1 614 292 7557; Email: bundschuh@mps.ohio-state.edu
Correspondence may also be addressed to Michael K. Chan. Tel: +1 614 292 8375; Fax: +1 614 292 6773; Email: chan@chemistry.ohio-state.edu.

which features an improved FOM calculation, and returns a list of hits rank-ordered by their FOM to the users interactively as well as through email.

METHODS

Overview of the SIB algorithm

PSI-BLAST achieves its high sensitivity for remote homologs by performing multiple rounds of searches. At each iteration, PSI-BLAST attempts to build a better model of its original query sequence using homologous sequences found in previous rounds. Thus, the earlier iterations (namely iteration one and two) are highly specific, finding mostly close homologs—but not very sensitive. In later iterations, more weakly related homologs are found, which are then used to build models for the even more remote homologs, imparting PSI-BLAST with the sensitivity needed for detecting the more remote homologs. However, these weakened relationships and diminishing similarities between sequences increase the probability of incorporating a non-homologous sequence into the model, leading to model corruption and the false identification of non-homologous sequences as putative homologs.

As described in the initial report of this algorithm, the SIB algorithm is built on the rationale that the benefits of low model corruption from early rounds of PSI-BLAST and the high sensitivity from later rounds can both be reaped by combining results from these iterations. Thus, the hits from the final round can be validated against those of earlier rounds, which in turn, would lead to improved discrimination of true and false positives. Toward this end, SIB uses a mathematical formulation (13) that combines the reported *E*-values of each ‘overlapped’ hit from the two PSI-BLAST iterations to calculate the FOM. This FOM serves the same role as the *E*-value reported by PSI-BLAST in the sense that it indicates the statistical significance of the hits relative to one another. For the FOM to be a true *E*-value, however, the underlying statistical independence of the two iterations assumption of the FOM formulation has to be fulfilled.

In our original FOM formulation, the *E*-values of the two iterations were converted to *P*-values. Under statistical independence, our assumption to combine *P*-values was to simply multiply the two *P*-values. This combined *P*-value was then converted to a FOM that was mathematically analogous to the ‘*E*’-value reported by PSI-BLAST. Recently, we have become aware that Bailey and Gribskov (15) have determined that the correct equation for calculating the combined *P*-value of two independent values should in fact be

$$P_{\mathrm{tot}} = P_2P_f(1 - \log(P_2P_f))$$

where P_2 and P_f are the *P*-values for iteration 2 and the final iteration, respectively. This new equation for calculating the combined *P*-value has been incorporated into the SIB algorithm running on the SIB-BLAST web server described below.

The performance of the original SIB algorithm was directly verified using the same 103 sequences from the Aravind ‘gold standard’ dataset (3,14) used by the PSI-BLAST developers. The evaluation results indicated that SIB exhibits higher specificity and sensitivity than that of PSI-BLAST for near identical computational time and a comparable performance to SAM-T2K using much less time. We have verified that the error versus coverage plot remains essentially unchanged after incorporating the new method of combining two *P*-values as expected since the new formula is still a monotonous function of P_2P_f.

SIB-BLAST Web server

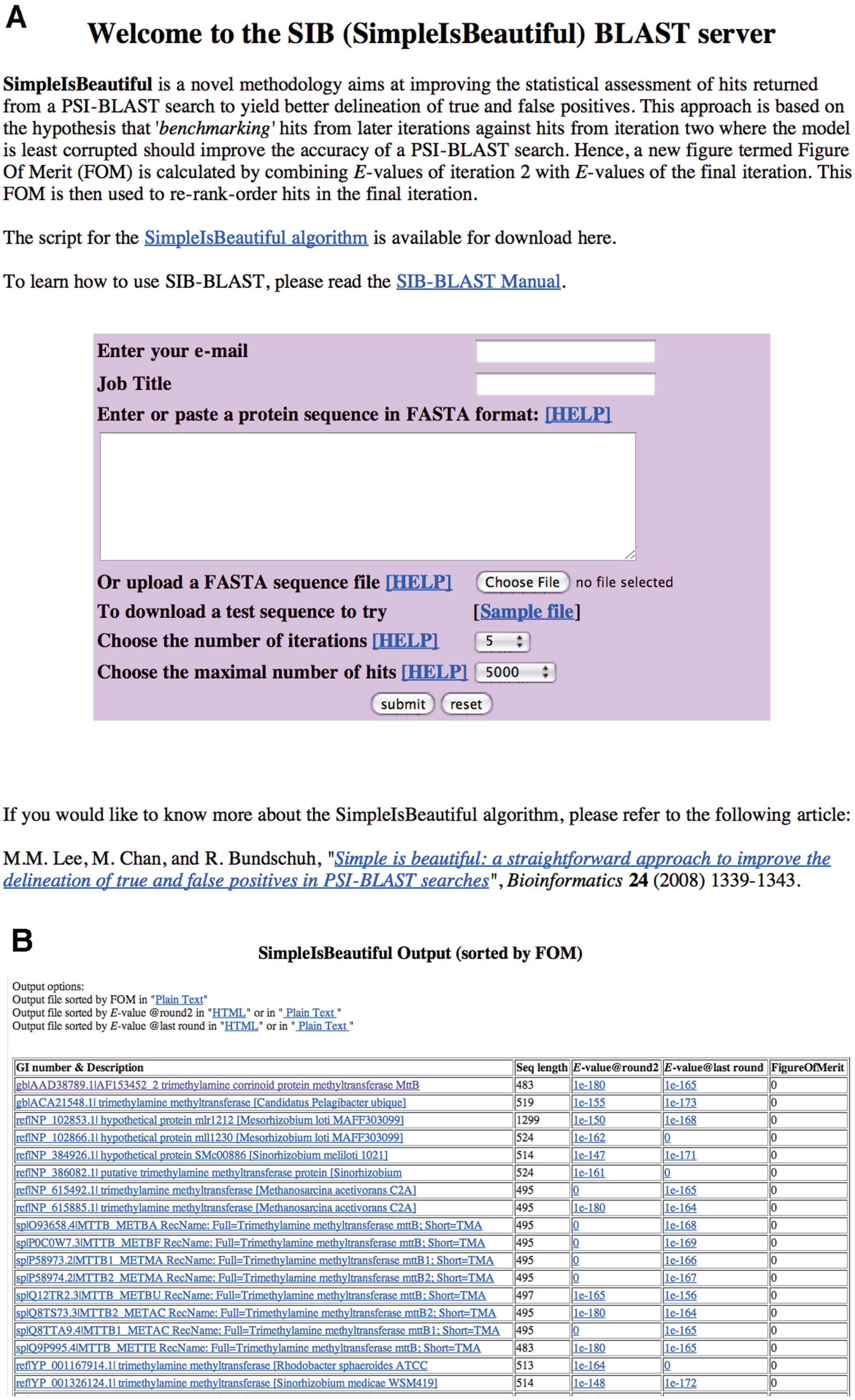
SIB-BLAST is comprised of the following steps: (i) performing a PSI-BLAST search of a query protein sequence against the non-redundant (NR) database; (ii) comparing resultant hits found in iteration 2 and the last iteration; (iii) calculating a FOM for each hit by combining its corresponding *E*-values at iteration 2 and at the final round; and (iv) re-ranking the merged list of hits according to their FOM.

The SIB-BLAST web server (Figure 1A) requests three inputs from the user: a protein sequence in FASTA format, the number of iterations of the PSI-BLAST search and the maximal number of target sequences reported in the PSI-BLAST search. Users are given the option to either paste a protein sequence or upload a file containing a protein sequence in FASTA format. The number of PSI-BLAST iterations to be performed is limited to be between 3 to 10 rounds, though users are advised to choose no more than five to six iterations as suggested by the PSI-BLAST developers (3). The number of target sequences reported by PSI-BLAST can be chosen as 1000, 2000, 5000, 10 000 or 20 000. This number is purposefully restricted to be higher than the PSI-BLAST default value to ensure that even weak hits are reported in the individual rounds as they might become significant once results from different rounds are combined.

Other than these three user-defined input parameters, the parameters used to conduct the search are preset. The NR database (updated weekly) is used for querying the sequence. The algorithm parameter for the Expect threshold, which reports the number of sequence matches, is set to 1000 instead of PSI-BLAST’s default value of 10. This higher threshold is to ensure that those true but more distant homologs identified in the final iteration but having very large *E*-value in round two are reported in both lists of iterations, which are subjected to downstream SIB processing. All other algorithm parameters, such as the word size, the scoring matrix and PSI-BLAST threshold *E*-value of 0.002 for inclusion of matches in the profile for the next round are set to default values.

Users are asked to provide an email address to which their results will be forwarded. Alternatively, users can bookmark the result link to obtain the output interactively when it is available. A status page, which shows the progress of the SIB-BLAST job is displayed upon submission.

SIB-BLAST outputs (Figure 1B) a combined list of hits from the second iteration and the last iteration rank-ordered by their corresponding FOM. Based on an analysis on the FOM’s coverage versus error curve on the



Welcome to the SIB (SimpleIsBeautiful) BLAST server

SimpleIsBeautiful is a novel methodology aims at improving the statistical assessment of hits returned from a PSI-BLAST search to yield better delineation of true and false positives. This approach is based on the hypothesis that '*benchmarking*' hits from later iterations against hits from iteration two where the model is least corrupted should improve the accuracy of a PSI-BLAST search. Hence, a new figure termed Figure Of Merit (FOM) is calculated by combining *E*-values of iteration 2 with *E*-values of the final iteration. This FOM is then used to re-rank-order hits in the final iteration.

The script for the SimpleIsBeautiful algorithm is available for download here.

To learn how to use SIB-BLAST, please read the SIB-BLAST Manual.

Enter your e-mail

Job Title

Enter or paste a protein sequence in FASTA format: [HELP]

Or upload a FASTA sequence file [HELP] no file selected

To download a test sequence to try [Sample file]

Choose the number of iterations [HELP] 5

Choose the maximal number of hits [HELP] 5000

If you would like to know more about the SimpleIsBeautiful algorithm, please refer to the following article:

M.M. Lee, M. Chan, and R. Bundschuh, "*Simple is beautiful: a straightforward approach to improve the delineation of true and false positives in PSI-BLAST searches*", *Bioinformatics* **24** (2008) 1339-1343.

SimpleIsBeautiful Output (sorted by FOM)

Output options:
Output file sorted by FOM in "Plain Text"
Output file sorted by *E*-value @round2 in "HTML" or in " Plain Text "
Output file sorted by *E*-value @last round in "HTML" or in " Plain Text "

GI number & Description	Seq length	*E*-value@round2	*E*-value@last round	FigureOfMerit
gb|AAD38789.1|AF153452_2 trimethylamine corrinoid protein methyltransferase MttB	483	1e-180	1e-165	0
gb|ACA21548.1| trimethylamine methyltransferase [Candidatus Pelagibacter ubique]	519	1e-155	1e-173	0
ref|NP_102853.1| hypothetical protein mlr1212 [Mesorhizobium loti MAFF303099]	1299	1e-150	1e-168	0
ref|NP_102866.1| hypothetical protein mll1230 [Mesorhizobium loti MAFF303099]	524	1e-162	0	0
ref|NP_384926.1| hypothetical protein SMc00886 [Sinorhizobium meliloti 1021]	514	1e-147	1e-171	0
ref|NP_386082.1| putative trimethylamine methyltransferase protein [Sinorhizobium	524	1e-161	0	0
ref|NP_615492.1| trimethylamine methyltransferase [Methanosarcina acetivorans C2A]	495	0	1e-165	0
ref|NP_615885.1| trimethylamine methyltransferase [Methanosarcina acetivorans C2A]	495	1e-180	1e-164	0
sp|O93658.4|MTTB_METBA RecName: Full=Trimethylamine methyltransferase mttB; Short=TMA	495	0	1e-168	0
sp|P0C0W7.3|MTTB_METBF RecName: Full=Trimethylamine methyltransferase mttB; Short=TMA	495	0	1e-169	0
sp|P58973.2|MTTB1_METMA RecName: Full=Trimethylamine methyltransferase mttB1; Short=TMA	495	0	1e-166	0
sp|P58974.2|MTTB2_METMA RecName: Full=Trimethylamine methyltransferase mttB2; Short=TMA	495	0	1e-167	0
sp|Q12TR2.3|MTTB_METBU RecName: Full=Trimethylamine methyltransferase mttB; Short=TMA	497	1e-165	1e-156	0
sp|Q8TS73.3|MTTB2_METAC RecName: Full=Trimethylamine methyltransferase mttB2; Short=TMA	495	1e-180	1e-164	0
sp|Q8TTA9.4|MTTB1_METAC RecName: Full=Trimethylamine methyltransferase mttB1; Short=TMA	495	0	1e-165	0
sp|Q9P995.4|MTTB_METTE RecName: Full=Trimethylamine methyltransferase mttB; Short=TMA	483	1e-180	1e-165	0
ref|YP_001167914.1| trimethylamine methyltransferase [Rhodobacter sphaeroides ATCC	513	1e-164	0	0
ref|YP_001326124.1| trimethylamine methyltransferase [Sinorhizobium medicae WSM419]	514	1e-148	1e-172	0

Figure 1. (**A**) Snapshot of the SIB-BLAST front page. The main section allows the user to submit the query protein sequence either by pasting in the window or by uploading a file. Dropdown menus allow the user to choose the number of rounds and the maximal number of target sequences. A brief explanation of each of these input parameters can be obtained by clicking the HELP link. Links to the Help manual and a downloadable SIB package are included on the front page. (**B**) Snapshot of the SIB-BLAST result page. The list of hits is displayed and rank-ordered by its corresponding FOM, along with its *E*-value at round two and at final round. Users can access the corresponding annotation of each hit by clicking the hit's GI number.

Aravind test set (14) in our original study, it appeared that the incurrence of errors increases dramatically at a FOM of $\sim 10^{-8}$. We have mentioned this empirical FOM threshold of 10^{-8} in the Manual page as a 'point of reference' for users to 'gauge' the statistical significance of each hit. Users are cautioned to the use of this threshold as this value is expected to depend on the size of the database.

Each hit on the results page is shown with its GI number, which has a link to the protein annotation page at the NCBI's website that allows the user to obtain details of the hit to ascertain whether the hit is indeed a true homolog or not. Along with the GI number and FOM, the *E*-values from the second and the final iterations are provided for each hit. Users can view the PSI-BLAST pairwise alignment between the query and the hit sequence by clicking on the *E*-value link. Different output options sorted by *E*-value at iteration 2 or *E*-value at the last iteration and in HTML or text format are also available to the users. The results page for these different options are organized identically to the default results page.

Documentation and runtime

The SIB-BLAST manual page connected by a clickable link on the SIB-BLAST front page provides users with a brief overview of the SIB algorithm in addition to a detailed description for each input parameter and an explanation of the result page. A sample sequence is made available for users to test a trial run of the SIB-BLAST server.

As reflected in the algorithm's name, the beauty of our approach is in its simplicity—it requires minimal changes to the existing PSI-BLAST algorithm since it uses information already output by PSI-BLAST and the processing time to calculate the FOM for individual hits and re-sorting them are negligible. Thus, SIB-BLAST improves the search accuracy of PSI-BLAST without compromising its computational efficiency.

Future development

The current FOM provides a relative measure of the statistical significance of the resultant hits against one another. It is not an *E*-value, however, due to SIB's implicit assumption in the calculation of the combined *P*-value—that the hits identified in the second and last rounds of PSI-BLAST are independent. It would be more meaningful if it were possible to calculate the combined *P*-value under the condition of the hits being dependent. Then the FOM calculated would be the true *E*-value, and the value obtained would provide an accurate measure of the error probability. Bailey and Grundy's POP (product of *P*-values) algorithm, which calculates the product of the *P*-values of correlated variables may provide some suggestions as to how to achieve this as a direction for future study (16). To improve on the functionality of SIB-BLAST, it would also be useful to provide a multiple sequence alignment of the hits (as is being done now in PSI-BLAST) in future versions of SIB-BLAST.

ACKNOWLEDGEMENTS

NSF (DBI-0317335 to R.B., partial); OSU internal funds (to M.K.C.); OSU Presidential Fellowship (to M.M.L.).

FUNDING

Funding for open access charge: Ohio State University.

Conflict of interest statement. None declared.

REFERENCES

1. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
2. Altschul,S.F., Madden,T.L., Schaffer,A.A., Zhang,J., Zhang,Z., Miller,W. and Lipman,D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.
3. Schaffer,A.A., Aravind,L., Madden,T.L., Shavirin,S., Spouge,J.L., Wolf,Y.I., Koonin,E.V. and Altschul,S.F. (2001) Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. *Nucleic Acids Res.*, **29**, 2994–3005.
4. Gotoh,O. (1996) Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. *J. Mol. Biol.*, **264**, 823–838.
5. Thompson,J.D., Plewniak,F. and Poch,O. (1999) A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Res.*, **27**, 2682–2690.
6. Gribskov,M., McLachlan,A.D. and Eisenberg,D. (1987) Profile analysis: detection of distantly related proteins. *Proc. Natl Acad. Sci. USA*, **84**, 4355–4358.
7. Eddy,S.R. (1996) Hidden Markov models. *Curr. Opin. Struct. Biol.*, **6**, 361–365.
8. Hughey,R. and Krogh,A. (1996) Hidden Markov models for sequence analysis: extension and analysis of the basic method. *Comput. Appl. Biosci.*, **12**, 95–107.
9. Grundy,W.N. (1998) Homology detection via family pairwise search. *J. Comput. Biol.*, **5**, 479–491.
10. Karplus,K., Barrett,C. and Hughey,R. (1998) Hidden Markov models for detecting remote protein homologies. *Bioinformatics*, **14**, 846–856.
11. Eddy,S.R. (1995) Multiple alignment using hidden Markov models. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, **3**, 114–120.
12. Park,J., Karplus,K., Barrett,C., Hughey,R., Haussler,D., Hubbard,T. and Chothia,C. (1998) Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. *J. Mol. Biol.*, **284**, 1201–1210.
13. Lee,M.M., Chan,M.K. and Bundschuh,R. (2008) Simple is beautiful: a straightforward approach to improve the delineation of true and false positives in PSI-BLAST searches. *Bioinformatics*, **24**, 1339–1343.
14. Aravind,L. and Koonin,E.V. (1999) Gleaning non-trivial structural, functional and evolutionary information about proteins by iterative database searches. *J. Mol. Biol.*, **287**, 1023–1040.
15. Bailey,T.L. and Gribskov,M. (1998) Combining evidence using p-values: application to sequence homology searches. *Bioinformatics*, **14**, 48–54.
16. Bailey,T.L. and Grundy,W.M. (1999) Classifying proteins by family using the product of correlated p-values. *Proceedings of the Third Annual International Conference on Computational Molecular Biology*, ACM New York, NY, USA. pp. 10–14.