BIOINFORMATICS APPLICATIONS NOTE Vol. 26 no. 16 2010, pages 2055–2056
doi:10.1093/bioinformatics/btq320

Sequence analysis Advance Access publication June 20, 2010

Bridges: a tool for identifying local similarities in long sequences

Alexey S. Kondrashov* and Raquel Assis

Center for Computational Medicine and Bioinformatics and the Life Sciences Institute, University of Michigan, Ann Arbor, MI 48109, USA

Associate Editor: John Quackenbush

ABSTRACT

Summary: Bridges is a heuristic search tool that uses short word matches to rapidly identify local similarities between sequences. It consists of three stages: filtering input sequences, identifying local similarities and post-processing local similarities. As input sequence data are released from memory after the filtering stage, genome-scale datasets can be efficiently compared in a single run. Bridges also includes 20 parameters, which enable the user to dictate the sensitivity and specificity of a search.

Availability: Bridges is implemented in the C programming language and can be run on all platforms. Source code and documentation are available at http://github.com/rassis/bridges.

Contact: kondrash@umich.edu

Received on March 30, 2010; revised on May 21, 2010; accepted on June 9, 2010

1 INTRODUCTION

Identifying homologous genomic segments is fundamental to tackling a number of biological problems, including mapping functional elements, predicting protein structures, quantifying molecular evolutionary dynamics and establishing phylogenetic relationships. Homologous segments can be located with high accuracy by employing the Smith–Waterman approach, a dynamic programming algorithm (Smith and Waterman, 1981). However, because it entails examining every possible alignment, the Smith–Waterman approach is computationally intensive and time-consuming, rendering its use unrealistic for many large-scale projects (Altschul *et al*., 1990).

In the past quarter century, several heuristic tools have been developed to rapidly locate homologous segments (Altschul *et al*., 1990; Kent, 2002; Pearson and Lipman, 1988). Rather than traversing sequences base-by-base, such programs limit their focus to regions with short exact word matches. Though this means that sensitivity is lower when searching for distantly related similarities, heuristic approaches are orders of magnitude faster than the Smith–Waterman approach and have low computational costs associated with them (Altschul *et al*., 1990). For these reasons, such tools have become an invaluable resource for biologists and form the backbone of bioinformatics.

Recently, we were faced with the task of identifying pairs of unique paralogous segments in the *Drosophila melanogaster* genome. We encountered a number of obstacles when attempting to use the entire genome as both a query and database with currently available search tools. Thus, we developed Bridges, which can perform rapid memory-efficient heuristic searches on genome-scale datasets. Another asset of Bridges is that it is highly flexible, with 20 parameters that enable the user to tailor a search to his or her particular goals.

*To whom correspondence should be addressed.

2 IMPLEMENTATION

Bridges requires two files as input: a database sequence file and a query sequence file. The query file can contain either a single query or a list of queries in FASTA format. Additionally, the user can modify 27 parameters, 20 of which influence the results produced by the program. The output file lists parameters used, coordinates and alignment scores of similarities and, optionally, corresponding sequences.

When multiple queries are specified, each query is individually compared to the database. Thus, similar to other heuristic programs, the user can compare several sequences to a database in a single run. The user can also choose to look for similarities on the direct strand, reverse-complemented strand or both. Further, there is the option to ignore similarities residing on the diagonal of the alignment matrix, which is useful when the same sequence is being used as the query and database.

Bridges can be used in the same capacity as other heuristic search tools, such as FASTA (Pearson and Lipman, 1988) and BLAST (Altschul *et al*., 1990). Its algorithm is similar to those employed by such programs and can be split into the following three stages, the third of which is optional:

(1) filtering input sequences;
(2) identifying local similarities; and
(3) post-processing local similarities.

2.1 Filtering input sequences

In this stage, Bridges masks low-complexity regions of the database and query sequences. Strictness of filtering can be adjusted via four parameters, though it is important to note that lax parameters may increase the runtime of the next stage if input sequences are highly repetitive. All *N*'s are automatically masked, but the user can decide whether to filter lowercase letters already present in sequences from previous masking. The user can also choose the word size used for masking, as well as the maximum frequency of a word in both query and database sequences.

Filtering is accomplished in two steps. First, a lookup table of all words and their frequencies in a sequence is constructed. Then, all words that occur more than allowed are masked. The output file specifies the fraction of each sequence that was masked.

Bridges also separately outputs filtered sequences, with masked characters in lowercase.

2.2 Identifying local similarities

As with other heuristic search tools, this stage is performed by examining regions containing exact word matches between the database and query sequences. Word length is given as a parameter, allowing the user to control the sensitivity of a search. However, choice of word length is also critical to runtime and memory usage. While decreasing word length increases search sensitivity, it also significantly increases runtime and memory requirements. Thus, one should only use short words (<10 nt) when looking for weak similarities. Additional parameters are maximum distance between words, mismatch and gap penalties, and the minimum score for a local similarity.

Identification of local similarities begins with the construction of a lookup table of all words in the database sequence. Next, the query sequence is scanned, and positions for all words it has in common with the database are recorded. Consecutive word matches are then linked, forming long chains of exact matches. Bridges compares all pairs of chains, temporarily linking them if the distance between them is less than or equal to the maximum distance specified. The alignment score is calculated by subtracting the multiple of the gap length and the gap penalty from the number of exact matches. All possible linked and unlinked similarities are scored, and the highest scoring configurations are kept. Resulting local similarities with scores greater than or equal to the minimum score undergo post-processing if the user elects this option. Otherwise, these similarities are sent to the output file.

2.3 Post-processing local similarities

Though optional, post-processing includes two unique features that can be exploited for specialized project goals. One is the removal of local similarities that occur at a lower or higher copy number than desired by the user. For example, one may want to look for similarities that occur at least three times and a maximum of five times. In our case, since we sought only pairs of paralogs, we set both the minimum and maximum number of similarities to two. Other heuristic search tools report similarities of all copy numbers, which would have required us to filter the results accordingly.

The second feature is merging neighboring local similarities. Here, the user specifies the maximum distance between similarities for merging. This is useful when one is looking for long regions of homology or similarities that may include large insertions or deletions. For example, this feature was valuable to us since our goal was to obtain full-length paralogs, and would be similarly advantageous to someone attempting to locate orthologs.

The first step of post-processing is calculating the coverages of each sequence. Similarities residing within or within a fraction of (another parameter) low- or high-coverage regions designated by the user are removed. Next, each pair of remaining similarities is evaluated. If members of a pair reside along the same diagonal (i.e. there are no gaps within their alignment), they are linked if their distance is less than or equal to the maximum distance. Otherwise, gap length, a penalty chosen by the user, and maximum distance are used together to determine whether they should be linked. Once all comparisons are completed, finished similarities are sent to the output file.

3 DISCUSSION

Bridges is a fast and efficient search tool for identifying homologous segments between long sequences. In a single run of Bridges on a Linux machine, we were able to compare the entire *D.melanogaster* genome to itself and specifically locate paralogous pairs. This took <2 h and used a maximum of 1.4 GB of memory during the entire run. Some important parameters used were a masking word size of 13, a searching word size of 12 and a minimum score of 100. While BLAST took approximately the same amount of time and memory to run, it produced shorter similarities, including those along the diagonal of the alignment. BLAT, with default parameters, ran for over 2 weeks before reaching the upper limit of 32 GB of memory and crashing. Thus, using either of these programs would have required us to filter and stitch together local similarities or to split up the genome substantially.

An added strength of Bridges lies in its ability to be guided by the user via an array of parameters. This flexibility allows the user to control the sensitivity and specificity of a search. For our purposes, having a parameter-rich program to work with was invaluable in that we were able to specifically locate the types of similarities we were interested in. Such flexibility can also be problematic if the user does not know what parameter values will produce the desired output. However, because Bridges produces output rapidly, the user is free to experiment with several sets of parameters. In fact, we found it helpful to examine different types of output to better understand how modification of certain parameters affected our results. For example, increasing the word size or the minimum score resulted in fewer, but stronger, similarities. Thus, the ability of the user to experiment with parameters is, in itself, yet another strong asset to Bridges.

ACKNOWLEDGEMENTS

Bridges is named after Calvin B. Bridges, who described the first pair of paralogous genomic segments (Bridges, 1936).

Funding: University of Michigan Rackham Merit Fellowship.

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.
Bridges,C.B. (1936) The bar 'gene' a duplication. *Science*, **83**, 210–211.
Kent,W.J. (2002) BLAT—The BLAST-like alignment tool. *Genome Res.*, **12**, 656–664.
Pearson,W.R. and Lipman,D.J. (1988) Improved tools for biological sequence comparison. *Proc. Natl Acad. Sci. USA*, **85**, 2444–2448.
Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.