Appl Bioinformatics 2006; 5 (1): 49-53
1175-5636/06/0001-0049/$39.95/0

BioParser

A Tool for Processing of Sequence Similarity Analysis Reports

Marcos Catanho,[1,2] *Daniel Mascarenhas,*[1] *Wim Degrave*[1] and *Antonio Basílio de Miranda*[1]

1 Department of Biochemistry and Molecular Biology, Oswaldo Cruz Institute, Fiocruz, Rio de Janeiro, Brazil
2 Department of Genetics, Fernandes Figueira Institute, Fiocruz, Rio de Janeiro, Brazil

Abstract

The widely used programs BLAST (in this article, 'BLAST' includes both the National Center for Biotechnology Information [NCBI] BLAST® and the Washington University version WU BLAST) and FASTA for similarity searches in nucleotide and protein databases usually result in copious output. However, when large query sets are used, human inspection rapidly becomes impractical. BioParser is a Perl program for parsing BLAST and FASTA reports. Making extensive use of the BioPerl toolkit, the program filters, stores and returns components of these reports in either ASCII or HTML format. BioParser is also capable of automatically feeding a local MySQL® database with the parsed information, allowing subsequent filtering of hits and/or alignments with specific attributes. For this reason, BioParser is a valuable tool for large-scale similarity analyses by improving the access to the information present in BLAST or FASTA reports, facilitating extraction of useful information of large sets of sequence alignments, and allowing for easy handling and processing of the data.

Availability: BioParser is licensed under the Creative Commons Attribution-NonCommercial-NoDerivs 2.0 license terms (http://creativecommons.org/licenses/by-nc-nd/2.0/) and is available upon request. Additional information can be found at the BioParser website (http://www.dbbm.fiocruz.br/BioParser.html).

Contact: Bioinformatics Team at Fiocruz (bioinfoteam@fiocruz.br)

Background

Searching for similarities between biological sequences (nucleic acids and proteins) is one of the most frequent computational activities for the modern biological research community. This procedure allows the detection of evolutionary, structural and functional relationships among compared sequences, providing decisive clues for the characterisation of biological properties of new sequences from stored sequences' annotations.[1] Two well known and widely used similarity search programs are BLAST[2,3] (in this article, 'BLAST' includes both the National Center for Biotechnology Information [NCBI] BLAST® and the Washington University version WU BLAST) and FASTA,[4,5] developed to speed up local alignment processes given the elevated computational cost of doing exhaustive comparisons using a rigorous dynamic algorithm, such as the Smith-Waterman[6] algorithm.

An outcome of the recent technological advances in DNA sequencing is the huge increase in the size of public sequence databases, such as GenBank® (http://www.ncbi.nih.gov/Genbank) and Swiss-Prot (http://us.expasy.org/sprot/), and the number of new sequences to be analysed, originating from large-scale sequencing projects. Also, the manual inspection of large datasets derived from high-throughput sequence similarity analyses employing the aforementioned programs has become impractical, requiring automation.

To help analyse BLAST and FASTA comparison results, we developed the program BioParser, which offers a graphical user interface (GUI) for parsing many varieties of BLAST and FASTA reports.

Resource Description

Capabilities

Through the Bio::SearchIO module of the BioPerl toolkit[7] (http://bioperl.org [tested with version 1.5.0]; currently the best

collection of open-source Perl codes for life science research), BioParser stores and returns the components of the BLAST and FASTA reports in two different formats, according to the user's choice: ASCII (tabular) or HTML (list). BioParser is also capable of automatically feeding a local MySQL® (http://www.mysql.com/ [tested with version 4.0.22]) database with the parsed information, allowing subsequent filtering of hits and/or alignments with specific attributes through its database interface, providing a valuable tool especially for large-scale similarity analy-

BioParser has many advantages over others BLAST report parsers, such as MSPcrunch (http://www.cgr.ki.se/cgb/groups/sonnhammer/MSPcrunch.html), Boulder (http://stein.cshl.org/software/boulder/), MuSeqBox[8] and Zerg.[9] Firstly, it is applicable to almost all flavours of BLAST and FASTA reports, not just the NCBI versions, as well as to the SSEARCH (ftp://ftp.virginia.edu/pub/fasta) sequence alignment program report. Secondly, it is multi-platform, since most operating systems support Perl and MySQL®. Thirdly, it offers multiple parsing and filtering options. Fourthly, the BioPerl modules, which are part of "an international effort to develop open source Perl tools for Bioinformatics, Genomics and Life Sciences research",[10] are constantly updated and improved, providing a reliable library source.

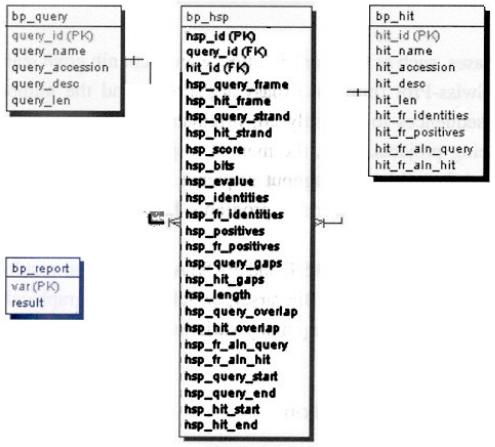

Fig. 1. Entity-relationship diagram showing the relational structure of the BioParser database. The entities and their relationships are described in the article and on the BioParser website. The *bp_report* table contains general information about the sequence similarity search performed, such as algorithm and version, database name and length, and searching parameters. **FK** = foreign key; **PK** = primary key.

Implementation and Developer Resources

The proposed BioParser database structure is simple and intuitive: for each aligned pair present in the sequence similarity report, the attributes related to the query and hit sequences are stored (without redundancy) in the *bp_query* and *bp_hit* tables, respectively. The attributes that characterise each alignment (high-scoring segment pair; HSP) are stored in the *bp_hsp* table, which is linked to the query and hit tables by two foreign keys: *query_id* and *hit_id*, respectively (figure 1).

BioParser was written in Perl (version 5.8.4; http://www.perl.org/), and its GUI was programmed using the Perl/Tk module (http://www.perltk.org/) [figure 2]. BioParser makes extensive use of the BioPerl toolkit and runs on all platforms that support Perl and MySQL®.

A schematic representation of the full system architecture is presented in figure 3. Basically, BioParser takes a BLAST, FASTA or SSEARCH flat file as an input and uses the Bio::SearchIO module of the BioPerl library to parse most of the information in that file. At least three different parsing options are offered, saving the parsed information in (i) ASCII or (ii) HTML format, in which the parsed elements are displayed as a table or a list, respectively, or (iii) transferring the parsed information to a local MySQL® database, which is done automatically by BioParser. The Bio::SearchIO module parses almost all varieties of BLAST (BLASTN, BLASTP, BLASTX, TBLASTN, TBLASTX) and FASTA (FASTA, FASTX/FASTY, TFASTX/TFASTY) reports, supporting ungapped and gapped BLAST versions 1.x and 2.x, respectively. Additionally, it also parses the output of another sequence alignment program, SSEARCH, which compares protein or DNA sequences with a protein or DNA sequence database using the Smith-Waterman algorithm and is distributed with the standalone version of FASTA (ftp://ftp.virginia.edu/pub/fasta/).

The BioParser software also includes a web-based interface (BioParser browser), which offers a user-friendly environment to interact with the MySQL® database, allowing the user to apply a number of selection criteria to the parsed data so as to filter out hits and/or alignments with specific features (figure 3 and figure 4). The available filtering options are the following: *QueryName* (name of the query sequence), *HitName* (name of the hit sequence), *Score* (raw score), *Bitscore* (bit score), *Identity(%)* (fraction of identical positions for a given HSP), *AlnQuery(%)* (fraction of the query sequence that has been aligned within a given HSP), *AlnHit(%)* (fraction of the hit sequence that has been aligned within a given HSP), *Evalue* (expectation value for the HSP) and *SizeDiff* (difference in length, expressed as a fraction, between the

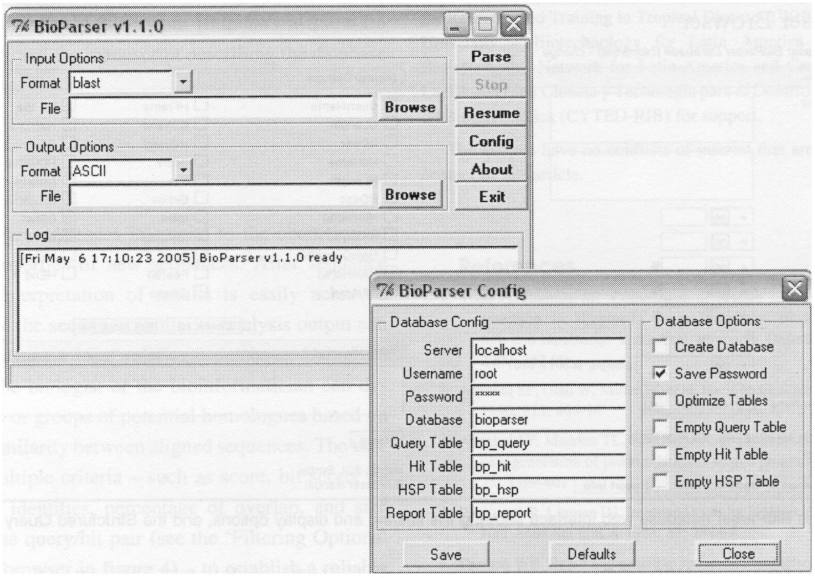

Fig. 2. An overview of the BioParser graphical interface showing the parsing option fields and the options for database configuration.

query and hit sequences). The user can also choose which of the following BLAST, FASTA or SSEARCH attributes must be displayed in the filtering result: *QueryName*, *QDesc* (query description), *QLength* (query length), *HitName*, *HDesc* (hit description), *HLength* (hit length), *HIdent(%)* (overall fraction of identical positions across all HSPs), *HPos(%)* (overall fraction of conserved positions across all HSPs), *HAlnQuery(%)* (fraction of the query sequence which has been aligned across all HSPs), *HAlnHit(%)* (fraction of the hit sequence which has been aligned across all HSPs), *QFrame* (frame of the query sequence), *HFrame* (frame of the hit sequence), *QStrand* (strand of the query), *HStrand* (strand of the hit), *Score*, *Bits* (bit score), *Evalue*, *Ident* (number of identical residues), *Ident(%)* (fraction of identical positions for a given HSP), *Pos* (number of conserved residues), *Pos(%)* (fraction

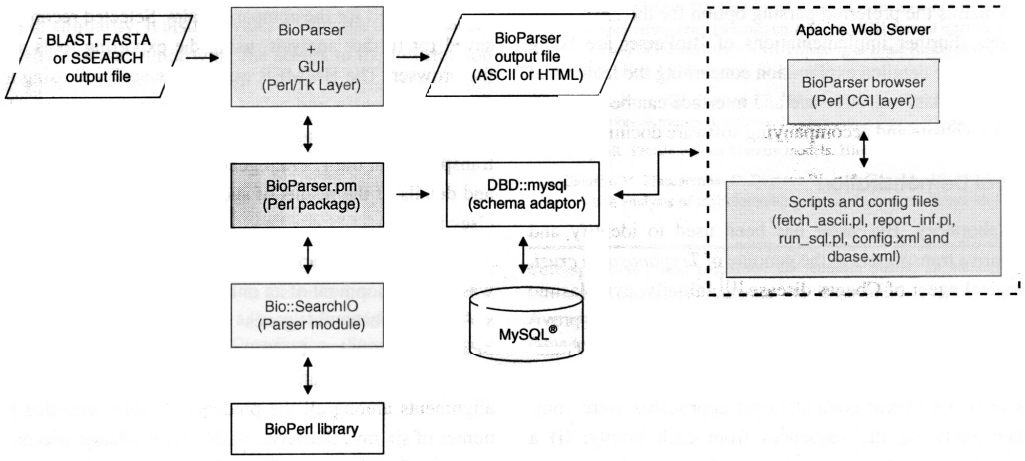

Fig. 3. Schematic representation of the BioParser system architecture. The grey boxes represent software developed by the authors. **CGI** = common gateway interface; **GUI** = graphical user interface.

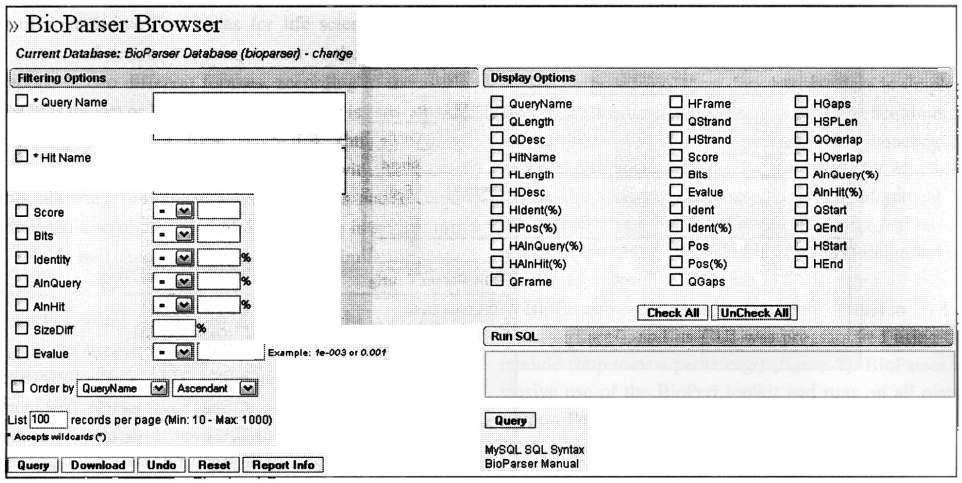

Fig. 4. A snapshot of the BioParser database web interface showing the filtering and display options, and the Structured Query Language (SQL) field.

of conserved positions for a given HSP), *QGaps* (number of gaps in the query alignment), *HGaps* (number of gaps in the hit alignment), *HSPLen* (full length of the alignment), *QOverlap* (length of query participating in alignment minus gaps), *HOverlap* (length of hit participating in alignment minus gaps), *AlnQuery(%)*, *AlnHit(%)*, *QStart* (query start position from the alignment), *QEnd* (query end position from the alignment), *HStart* (hit start position from the alignment) and *HEnd* (hit end position from the alignment).

Currently, the filtering is applied to the data stored in a database, which is the preferred parsing option for the analysis of large datasets. Further implementations of BioParser are being developed. A more detailed explanation concerning the features of BioParser and its database structure and interface can be found on the BioParser website and accompanying software documentation.

Empirical Demonstration

In our laboratory, BioParser has been used to identify and classify putative transporters in the genome of *Trypanosoma cruzi*, the aetiological agent of Chagas disease.[11] Initially, cytoplasmic membrane transport protein sequences from 177 organisms, previously classified in 130 families, were obtained from the TransportDB database[12] (http://www.membranetransport.org/). The efficiencies of two different computational approaches were compared when analysing the sequences from each family: (i) a BLAST search against the *T. cruzi* dataset of annotated sequences (preliminary sequence data were obtained from The Institute for Genomic Research website at http://www.tigr.org) and (ii) a HMMER[13] (http://hmmer.wustl.edu/) search against the same dataset using hidden Markov models built for each group achieved after clustering. The BLAST output file was analysed with BioParser, which had been set to parse and store the results in a local MySQL® database with the proposed structure (figure 1). Significant alignments (selected from the database through the BioParser browser) were those above a similarity threshold based on (i) the fraction of identical positions in the HSP, (ii) the fraction of the query and hit sequences aligned within the HSP and (iii) the E-value assigned for the compared pairs. Selected records were then saved for further analysis, using the exporting tools available in the browser. The HMMER output was analysed using a different approach to parse and select significant alignments. Using both methods, we were able to identify a total of 37 families of transporters in the *T. cruzi* genome (data not shown). Discussions and details of the results of such an application will be published elsewhere.

Another relevant application of BioParser in our research group was the development of an online database for comparative analysis of mycobacterial genes and genomes (GenoMycDB). BioParser was used to build the central structure of this database, composed of the results obtained after FASTA pairwise sequence alignments among all the predicted proteins encoded by the genomes of six mycobacteria. Basically, BioParser was employed to parse the FASTA output file and load the parsed information into a local MySQL® database with the aforementioned structure (fig-

ure 1). GenoMycDB is freely accessible (http://www.dbbm.fiocruz.br/GenoMycDB), and the manuscript describing the database will be published soon.[14]

Discussion

Finding homologous sequences is crucial to the characterisation of biological properties of new sequences. After sequence similarity analysis, interpretation of results is easily achieved using BioParser, since the sequence similarity analysis output can be parsed and loaded into a local relational database. Using the BioParser browser, the biologist or the bioinformatician can dynamically select pairs or groups of potential homologues based on different aspects of similarity between aligned sequences. The user can define one or multiple criteria – such as score, bit score, E-value, percentage of identities, percentage of overlap, and size difference between the query/hit pair (see the 'Filtering Options' section of BioParser browser in figure 4) – to establish a reliable cut-off of similarity so as to infer homology. For instance, successive searches in the database, varying the number and/or the value of the parameters used as threshold of similarity, can easily be done, and the result of each query can be saved automatically in a flat file for further inspection.

Hence, BioParser simplifies the analysis of the results produced by the most common sequence similarity analysis programs, making it easier, for instance, to identify evolutionary, structural or functional relationships among the compared sequences, based on their degree of similarity. It also provides a valuable tool for large-scale similarity analyses, improving the access to the information present in BLAST, FASTA or SSEARCH reports, facilitating extraction of useful information on large sets of sequence alignments, and allowing for easy handling and processing of the data.

Acknowledgements

We wish to thank Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Programa de Apoio à Pesquisa Estratégica em Saúde – Fiocruz (PAPES-Fiocruz), World Health Organization – Special Programme for Research and Training in Tropical Diseases (WHO/TDR), United Nations University – Biotechnology for Latin America and the Caribbean – Bioinformatics Network for Latin-America and Caribbean (UNU-BIOLAC LacBioNet) and Ciencia y Tecnología para el Desarrollo– Red Iberoamericana de Bioinformática (CYTED-RIB) for support.

The authors have no conflicts of interest that are directly relevant to the content of this article.

References

1. Yona G, Brenner SE. Comparison of protein sequences and practical database searching. In: Higgins D, Taylor W, editors. Bioinformatics: sequence, structure and databanks: a practical approach. Oxford: Oxford University Press, 2000: 167-90
2. Altschul SF, Gish W, Miller W, et al. Basic local alignment search tool. J Mol Biol 1990; 215: 403-10
3. Altschul SF, Madden TL, Schaffer AA, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res 1997; 25: 3389-402
4. Pearson WR, Lipman DJ. Improved tools for biological sequence comparison. Proc Natl Acad Sci U S A 1988; 85: 2444-8
5. Pearson WR. Rapid and sensitive sequence comparison with FASTP and FASTA. Methods Enzymol 1990; 183: 63-98
6. Smith TF, Waterman MS. Comparison of biosequences. Adv Appl Math 1981; 2: 482-9
7. Stajich JE, Block D, Boulez K, et al. The Bioperl toolkit: Perl modules for the life sciences. Genome Res 2002; 12: 1611-8
8. Xing L, Brendel V. Multi-query sequence BLAST output examination with MuSeqBox. Bioinformatics 2001; 17: 744-5
9. Paquola AC, Machado AA, Reis EM, et al. Zerg: a very fast BLAST parser library. Bioinformatics 2003; 19: 1035-6
10. BioPerl [online]. Available from URL: http://www.bioperl.org/ [Accessed 2005 June]
11. Henriques C, Otto TD, Catanho M, et al. Classification of transporter families in *Trypanosoma cruzi* [abstract no. BM128]. XXI Annual meeting of the Brazilian Society of Protozoology/XXXII Meeting of Basic Research in Chagas Disease; 2005 Nov 7-9; Caxambú, Brazil; 119
12. Ren Q, Kang KH, Paulsen IT. TransportDB: a relational database of cellular membrane transport systems. Nucleic Acids Res 2004; 32: D284-8
13. Eddy SR. Profile hidden Markov models. Bioinformatics 1998; 14 (9): 755-63
14. Catanho M, Mascarenhas D, Degrave W, et al. GenoMycDB: database for comparative analysis of mycobacterial genes and genomes. Genet Mol Res. In press

Correspondence and offprints: Dr *Antonio Basílio de Miranda*, Fundação Oswaldo Cruz, Departamento de Bioquímica e Biologia Molecular, Avenida Brasil 4365, Manguinhos, Rio de Janeiro, RJ, CEP 21045-900, Brazil. E-mail: antonio@fiocruz.br