BIOINFORMATICS APPLICATIONS NOTE Vol. 29 no. 24 2013, pages 3222–3224
doi:10.1093/bioinformatics/btt553

Genome analysis Advance Access publication September 24, 2013

BlastGraph: a comparative genomics tool based on BLAST and graph algorithms

Yanbo Ye[1,2], Bo Wei[1], Lei Wen[1,2] and Simon Rayner[1,3,*]

[1]Key Laboratory of Agricultural and Environmental Microbiology, Wuhan Institute of Virology, Wuhan, Hubei, China, 430071, [2]Graduate School of Chinese Academy of Sciences, Beijing, China and [3]Exiqon A/S, Vedbaek, Denmark

Associate Editor: John Hancock

ABSTRACT

Summary: BlastGraph is an interactive Java program for comparative genome analysis based on Basic Local Alignment Search Tool (BLAST), graph clustering and data visualization. The software generates clusters of sequences of multiple genomes from all-to-all BLAST results and visualizes the results in graph plots together with related information such as sequence features, gene conservation and similarity relationships. Pruning algorithms are used to reduce results to more meaningful subclusters. Subsequent analyses can then be conducted based on the predicted clusters, including gene content, genome phylogenetics and gene gain and loss.

Availability and implementation: https://github.com/bigwiv/BlastGraph.

Contact: simon.rayner.cn@gmail.com

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

Received on March 28, 2013; revised on July 31, 2013; accepted on September 19, 2013

1 INTRODUCTION

Sequence analysis, especially comparative genomics, often starts with a sequence similarity search using standard tools such as Basic Local Alignment Search Tool (BLAST) (Altschul *et al.*, 1990) or BLAT (Kent, 2002), and homology/orthology detection and clustering using additional tools such as BLASTClust (in the BLASTALL package), Clustering Database at High Identity with Tolerance (CD-HIT) (Li *et al.*, 2006), Inparanoid, OrthoMCL (Markov Cluster Algorithm), EuKaryotic Orthologous Groups (KOG) and TribMCL (Chen *et al.*, 2007). Nevertheless, BLAST remains the most widely used method and is central to the majority of associated secondary packages. Additionally, although there are many clustering tools based on BLAST and graph clustering algorithms, including BLASTClust using single-linkage clustering, OrthoMCL (Li *et al.*, 2003) and TribMCL (Enright *et al.*, 2002) using the Markov Cluster algorithm (MCL) (Van Dongen, 2000), these command line tools provide only a simple clustering result. Currently, there is no single integrated tool available for post clustering analyses, such as phyletic pattern study, genome level phylogenetics or gene gain and loss analysis. We have developed BlastGraph for the generation and visualization of clusters as a graph with accompanying statistics, evaluation metrics and associated background information. The user can check the coherence of a cluster, identify useful relationships (such as indels and fusion events) or gene types and interactively split and edit the cluster. This is especially useful when remote homologies with low similarities exist among sequences. Thus, we present a user-friendly workflow (Supplementary Data S1) for integrated comparative genomics analyses.

2 IMPLEMENTATION

BlastGraph uses several widely used programs and libraries for implementation: BLAST, MCL, BioJava (Holland *et al.*, 2008) and JUNG (O'Madadhain *et al.*, 2003).

2.1 Input data and graph creation

BlastGraph accepts BLAST extensible mark-up language (XML) results generated from a GenBank file with multiple genomes as input. For the GenBank file, all coding DNA sequences in the genomes are extracted into a FASTA amino acid or nucleic acid file with formatted information (Supplementary Data S2) in the FASTA header. The FASTA file is then converted to a BLAST database using and BLASTed to generate an XML result. One or more BLAST XML (Supplementary Data S3) results with different parameters can then be parsed and merged into an undirected graph, in which the vertices represent proteins or nucleotides of coding DNA sequences and edges represent the better sequence alignment of the reciprocal BLAST hits for that pair of vertices. This graph can be saved in graph XML format (Supplementary Data S4) and serves as raw data for clustering and subsequent analysis. Although BlastGraph was originally created to analyze large virus genome data, it is also possible to be applied to bacteria data with bigger genomes (Supplementary Data S9).

2.2 Graph clustering, filtering and visualization

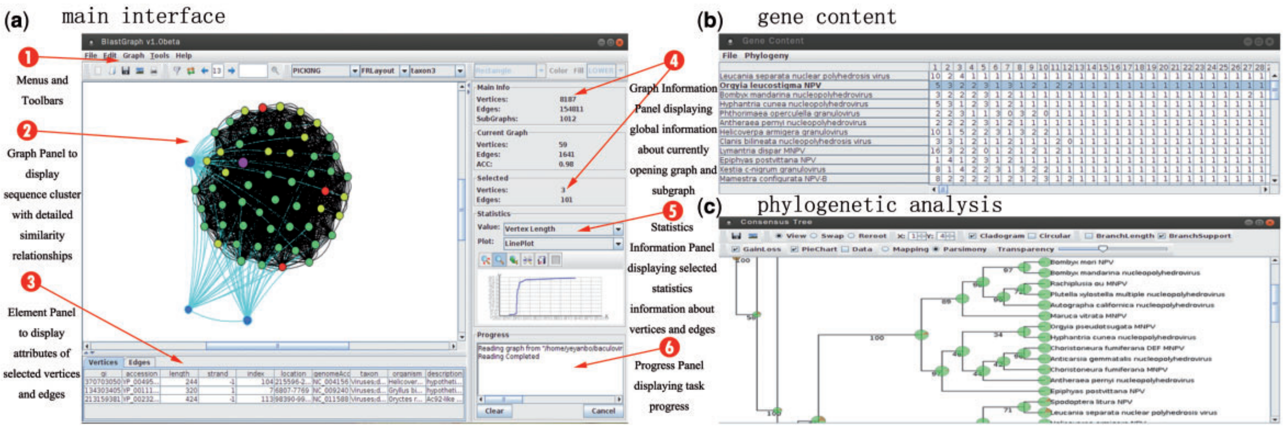
Normally, the raw graph from a BLAST result is too complex to show any meaningful information. We use two strategies to reduce the raw graph into smaller independent subgraphs: MCL clustering and edge filtering.

The MCL algorithm is designed for graph clustering by flow simulation (Van Dongen, 2000) and is widely used for biological network clustering (e.g. TribMCL, OrthoMCL) (Enright *et al.*, 2002; Li *et al.*, 2003) as it can minimize effects introduced by multiple domains in a single protein or shorter 'promiscuous' domains that have similar sequence but different function. In BlastGraph, there are three steps for MCL clustering: (i) generation of a

[*]To whom correspondence should be addressed.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

© The Author 2013. Published by Oxford University Press. All rights reserved. For Permissions, please e-mail: journals.permissions@oup.com


(a) main interface
1
Menus and Toolbars
2
Graph Panel to display sequence cluster with detailed similarity relationships
3
Element Panel to display attributes of selected vertices and edges
BlastGraph v1.0beta
File Edit Graph Tools Help
PICKING
FRLayout
taxon3
Rectangle
Color Fill LOWER
Vertices Edges
Clear
Cancel
4
Graph Information Panel displaying global information about currently opening graph and subgraph
5
Statistics Information Panel displaying selected statistics information about vertices and edges
6
Progress Panel displaying task progress
(b) gene content
Gene Content
File Phylogeny
(c) phylogenetic analysis
Consensus Tree
● View ○ Swap ○ Reroot
X: 1 Y: 4
☑ Cladogram ☐ Circular
☐ BranchLength ☑ BranchSupport
☑ GainLoss ☑ PieChart ☐ Data ○ Mapping ● Parsimony Transparency
100
58
100
100
88
86
97
72
90
34
46
66
42
44
97
71
Bombyx mori NPV
Bombyx mandarina nucleopolyhedrovirus
Rachiplusia ou MNPV
Plutella xylostella multiple nucleopolyhedrovirus
Autographa californica nucleopolyhedrovirus
Maruca vitrata MNPV
Orgyia pseudotsugata MNPV
Hyphantria cunea nucleopolyhedrovirus
Choristoneura fumiferana DEF MNPV
Anticarsia gemmatalis nucleopolyhedrovirus
Choristoneura fumiferana MNPV
Antheraea pernyi nucleopolyhedrovirus
Epiphyas postvittana NPV
Spodoptera litura NPV
Leucania separata nuclear polyhedrosis virus

Main Info
Vertices: 8187
Edges: 154811
SubGraphs: 1012

Current Graph
Vertices: 59
Edges: 1641
ACC: 0.98

Selected
Vertices: 3
Edges: 101

Statistics
Value: Vertex Length
Plot: LinePlot

Progress
Reading graph from "/home/yeyanbo/baculovir
Reading Completed

gi	accession	length	strand	index	location	genomeAcc	taxon	organism	description
370703050	YP_00495...	244	-1	104	215596-2...	NC_004156	Viruses;d...	Helicover...	hypotheti...
134303405	YP_00111...	320	1	7	6807-7769	NC_009240	Viruses;d...	Gryllus bi...	hypotheti...
213159381	YP_00232...	424	-1	113	98390-99...	NC_011588	Viruses;d...	Oryctes r...	Ac92-like ...

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Leucania separata nuclear polyhedrosis virus	10	2	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Orgyia leucostigma NPV	5	3	2	2	3	1	3	1	2	1	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Bombyx mandarina nucleopolyhedrovirus	3	2	2	2	3	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1
Hyphantria cunea nucleopolyhedrovirus	5	3	1	2	3	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Phthorimaea operculella granulovirus	2	2	3	1	1	3	0	3	2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Antheraea pernyi nucleopolyhedrovirus	2	2	2	2	3	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Helicoverpa armigera granulovirus	10	1	5	2	2	3	1	3	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Clanis bilineata nucleopolyhedrosis virus	3	3	1	2	1	1	2	1	1	1	2	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Lymantria dispar MNPV	16	3	2	2	0	1	2	1	2	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Epiphyas postvittana NPV	1	4	1	2	3	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Xestia c-nigrum granulovirus	8	1	4	2	2	3	1	3	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Mamestra configurata NPV-B	8	2	2	2	2	1	2	1	2	3	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Fig. 1. **a**) BlastGraph main interface showing the one cluster of the baculovirus data; (**b**) gene content window showing the presence time of different gene families by summarizing all gene clusters; (**c**) genome phylogenetic analysis window showing the gene content tree and the gene gain and loss result

weighted graph in 'abc' format (O'Madadhain *et al*., 2003) from the raw graph based on BLAST *E*-value or score; (ii) MCL clustering; (iii) creation of subgraphs from the raw graph according to clustering results. At the first step, the user can specify the parameters for weighting the graph (Supplementary Data S5), thus controlling the granularity of the MCL results along with the inflation parameter in the MCL algorithm.

Besides MCL clustering, edge filtering can serve as a supplementary method to create subgraphs and achieve meticulous trimming. We use several filtering criteria based on BLAST results (*E*-value, score density, percentage identity, percentage positive and three criterion of alignment coverage) to provide greater flexibility for pruning the graph (Supplementary Data S6).

BlastGraph allows visualization and navigation of subgraphs using the Java Universal Network/Graph (JUNG) graph library (O'Madadhain *et al*., 2003) (Fig. 1). Each subgraph can be customized by layout, user annotation and vertex colors to represent different attributes. Vertices and edges of a subgraph can be selected to show the associated information and imported data together with corresponding statistical analysis (Fig. 1c) to provide a comprehensive overview of the conservation of gene families and gene relationships among different species, and filtering and graph editing may be performed in the presence of multiple gene families. This also makes it possible to identify sequence similarities in more distantly related species.

2.3 Phyletic pattern and phylogenetics

After clustering and downstream analysis and processing, the final result can be used to create a phyletic pattern (or profile) in the form of a binary gene content matrix N, where '1' or '0' indicates the presence or absence of a gene, respectively. Rows correspond to genomes and columns correspond to gene families (i.e. element N_{ij} of row i and column j corresponds to gene j in genome i).

Based on the phyletic pattern, BlastGraph provides several configurable methods to build the gene content trees. To calculate the genome distance matrix, the Simple Matching, Jeccard Distance and the Snel (Snel *et al*., 1999) methods are implemented (Supplementary Data S6). Two distance-based tree construction algorithms (UPGMA, NJ) are implemented, and the consensus tree can be viewed directly within the program. The Bootstrap and Delete-Half Jackknife methods are used to estimate the tree branch support (Supplementary Data S7).

2.4 Core gene and ancestor estimation

The aforementioned features also facilitate the identification of core genes or branch-specific genes and the estimation of an ancestor genome (gene gain and loss) among different species, two important tasks in comparative genomics that can provide insight into function and phenotype changes or speciation history during evolution. In the tree window (Fig. 1), they can be conducted by the simple mapping and the Dollo parsimony method, respectively, and the result can be mapped on to the gene content tree (Supplementary Data S8).

3 CONCLUSION

BlastGraph is a user-friendly tool for comparative genomics analyses such as homologous sequence identification and clustering, phyletic patterns, genome phylogenetics, core gene identification and gene gain and loss. The workflow implemented in BlastGraph provides an enhanced and convenient environment that allows researchers to address specific questions relevant to genome comparison and evolution.

ACKNOWLEDGEMENTS

The authors would like to thank all members in the bioinformatics group and Dr James Adams for their valuable comments and suggestions.

Funding: 973 program [2012CB721102] and CAS project [KSCX2-EW-Z-3].

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al*. (1990) Basic local alignment search tool. *J. Mol. Biol*., **215**, 403–410.

Chen,F. *et al*. (2007) Assessing performance of orthology detection strategies applied to eukaryotic genomes. *PloS One*, **2**, e383.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016

Enright,A.J. *et al.* (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.*, **30**, 1575–1584.

Holland,R.C.G. *et al.* (2008) BioJava: an open-source framework for bioinformatics. *Bioinformatics*, **24**, 2096–2097.

Kent,W.J. (2002) BLAT—the BLAST-like alignment tool. *Genome Res.*, **12**, 656–664.

Li,L. *et al.* (2003) OrthoMCL: identification of ortholog groups for eukaryotic genomes. *Genome Res.*, **13**, 2178–2189.

Li,W. *et al.* (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**, 1658–1659.

O'Madadhain,J. *et al.* (2003) *The Jung (Java Universal Network/Graph) Framework*. University of California, Irvine, California.

Snel,B. *et al.* (1999) Genome phylogeny based on gene content. *Nat. Genet.*, **21**, 108–110.

Van Dongen,S.M. (2000) Graph clustering by flow simulation. PhD thesis, University of Utrecht, http://micansorg/mcl/.

Downloaded from http://bioinformatics.oxfordjournals.org/ at :: on August 30, 2016