BMC Bioinformatics

SOFTWARE Open Access

BLASTGrabber: a bioinformatic tool for visualization, analysis and sequence selection of massive BLAST data

Ralf Stefan Neumann[1], Surendra Kumar[1,2], Thomas Hendricus Augustus Haverkamp[3] and Kamran Shalchian-Tabrizi[1*]

Abstract

Background: Advances in sequencing efficiency have vastly increased the sizes of biological sequence databases, including many thousands of genome-sequenced species. The BLAST algorithm remains the main search engine for retrieving sequence information, and must consequently handle data on an unprecedented scale. This has been possible due to high-performance computers and parallel processing. However, the raw BLAST output from contemporary searches involving thousands of queries becomes ill-suited for direct human processing. Few programs attempt to directly visualize and interpret BLAST output; those that do often provide a mere basic structuring of BLAST data.

Results: Here we present a bioinformatics application named BLASTGrabber suitable for high-throughput sequencing analysis. BLASTGrabber, being implemented as a Java application, is OS-independent and includes a user friendly graphical user interface. Text or XML-formatted BLAST output files can be directly imported, displayed and categorized based on BLAST statistics. Query names and FASTA headers can be analysed by text-mining. In addition to visualizing sequence alignments, BLAST data can be ordered as an interactive taxonomy tree. All modes of analysis support selection, export and storage of data. A Java interface-based plugin structure facilitates the addition of customized third party functionality.

Conclusion: The BLASTGrabber application introduces new ways of visualizing and analysing massive BLAST output data by integrating taxonomy identification, text mining capabilities and generic multi-dimensional rendering of BLAST hits. The program aims at a non-expert audience in terms of computer skills; the combination of new functionalities makes the program flexible and useful for a broad range of operations.

Keywords: Analysis, BLAST, High-throughput, Taxonomy, Text-mining, Visualization

Background

Sequence similarity searches have become an integral part of contemporary life sciences [1]. More than two decades have now passed since the Basic Local Alignment Search Tool (BLAST) was introduced to the bioinformatic community [2,3], constituting a breakthrough for rapid similarity search tools [4]. Despite the staggering changes that have taken place in biology, sequencing and computing technology, BLAST remains the most common used algorithm for sequence similarity searches [5,6]. This is reflected by the exceptionally high numbers of citations for the two original BLAST papers (48632 citations [2], 49238 citations [3]; Google scholar - January 2014). The continued popularity is due to the intuitive appeal of the algorithm [4], its speed and efficiency [7,8], and being supported by a complete, rigorous statistical theory [5].

BLAST has been used for most purposes involving biological sequence searches and alignments, some examples being EST annotation [4,9-11], contig assembly [12,13], genomic fragment reconstruction [14], ORF validation [5], prediction of protein function and origin [4,6], distant homolog [5] and putative ortholog detection [15,16], phylogenetic analysis [8,17,18] and metagenomics [19].

BLAST services can be accessed numerous places on the web and are often free of charge; one of most popular

* Correspondence: kamran@ibv.uio.no
[1]Section for Genetics and Evolutionary Biology (EVOGENE) and Centre for Epigenetics, Development and Evolution (CEDE), University of Oslo, Oslo, Norway
Full list of author information is available at the end of the article

is the BLAST implementation hosted by the National Center for Biotechnology Information (NCBI) [20]. It receives hundreds of thousands requests a day [21,22], and presents BLAST results as textual reports with graphical representations of the calculated alignments.

Such web-based BLAST implementations are convenient to use for the analysis of a small number of sequences. However, both limitations of computational resources and the way results are presented render this approach unfeasible for large BLAST searches that might involve thousands (or even hundreds of thousands) of unique query sequences. In recent years, due to new sequencing technology, high-throughput searches of this magnitude have become a standard situation in many fields of research, such as EST annotation [4], genomics [5], metagenomics [19] and phylogenomics [23]. In addition, projects without high-throughput sequencing data also exhibit a trend towards increasing query numbers. Examples might be all-against-all comparisons useful in EST annotation [4], proteome-against-proteome searches in order to identify orthologs [15], or simply pooling diverse queries together so as to minimize the number of job submissions to an external computing resource.

In order to perform the actual high-throughput BLAST search, it is necessary to use a stand-alone implementation of the BLAST algorithm rather than relying on web-based public installations such as the NCBI resource [5]. Many research institutions have solved this by establishing high-performance computing (HPC) clusters hosting BLAST-related databases and pipelines. Since BLAST scales well when parallelized, such HPC installations can handle large high-throughput sequence input in a reasonable amount of time. The results of high-throughput BLAST runs could still present the user with gigabyte-sized text files, the data volume alone representing a challenge for inexperienced users. For specific research fields, massive BLAST output can be analysed by specialized, user-friendly programs that run on ordinary desktop computers – such as the MEGAN [19] program for metagenomics. For many other research fields however, scientists find suitable tools missing [24]. Surprisingly, this includes generic BLAST output interpreter programs capable of visualizing, analysing and selecting massive BLAST output data.

This reflects a more general problem in the field of biological data visualization and analysis - contemporary biological data generation has outpaced the traditional data processing approaches [20,24]. Some of the important features recently suggested to alleviate the perceived program shortcomings include (amongst others) improved usability, new visual analytical modes, and multi-scale standardized data representations [24]. Especially intriguing is the challenge of visualizing massive amounts of data in a suitable way. New, innovative visual representation techniques are needed that provide both an overview over, and facilitates detailed navigation into, the mass of information in these data sets [20]. Hence, improved visualization technologies are clearly amongst the key aspects of knowledge discovery and data mining [25]; some of which have begun to find their way into mainstream science (for instance, [26-28]). Suitable analysis tools will be in demand as sequencing technology become even more efficient.

We here present a program designed for the effective exploration of BLAST output generated from large scale database searches. It is aimed at an audience of computer non-experts not familiar with programming languages, database retrieval or command-line usage. The program facilitates visualization, analysis and selection of data. Importantly, the application provides new functionalities including taxonomic ordering of data, text search options, multi-dimensional display and a range of possibilities for filtering and downloading of data from public sequence databases. The program, introduction video and additional utilities are freely available for download at http://www.bioportal.no.

Implementation

BLASTGrabber consists of a downloadable desktop Java application capable of visualizing and selecting high-throughput BLAST output. Taxonomic analysis is supported based on mapping of NCBI gi-numbers to taxonomy identifiers, or alternatively parsing the headers of the BLAST hits. Selected BLAST hits and queries can be exported in text format (Figure 1).

The BLASTGrabber application

The BLASTGrabber desktop application is a GUI-based Java program that will run on Windows-, MacOS- or Linux- based computers. Apart from the Java Runtime Environment (version 1.6 or higher), no other installations are required to run the program. BLASTGrabber will work on low-spec systems including netbooks. For high-throughput data however, 4 GB of memory and a processor comparable to 1.4GHz Core i5 are recommended. BLASTGrabber is installed by extracting the installation zip file containing the BLASTGrabber JAR file and additional files (such as example data sets and NCBI taxonomy file). In order to maximize memory usage, the program can be started from the command line, explicitly specifying the java heap space parameters. Alternatively, double-clicking an included shell file can start BLASTGrabber with 2 GB of maximum heap spaces.

BLASTGrabber uses a custom input file format (*.bgr file format) to represent BLAST output. This file is produced by an import wizard included in the BLASTGrabber application (for details, see User guide in Additional file 1). The wizard will recognize BLAST XML output

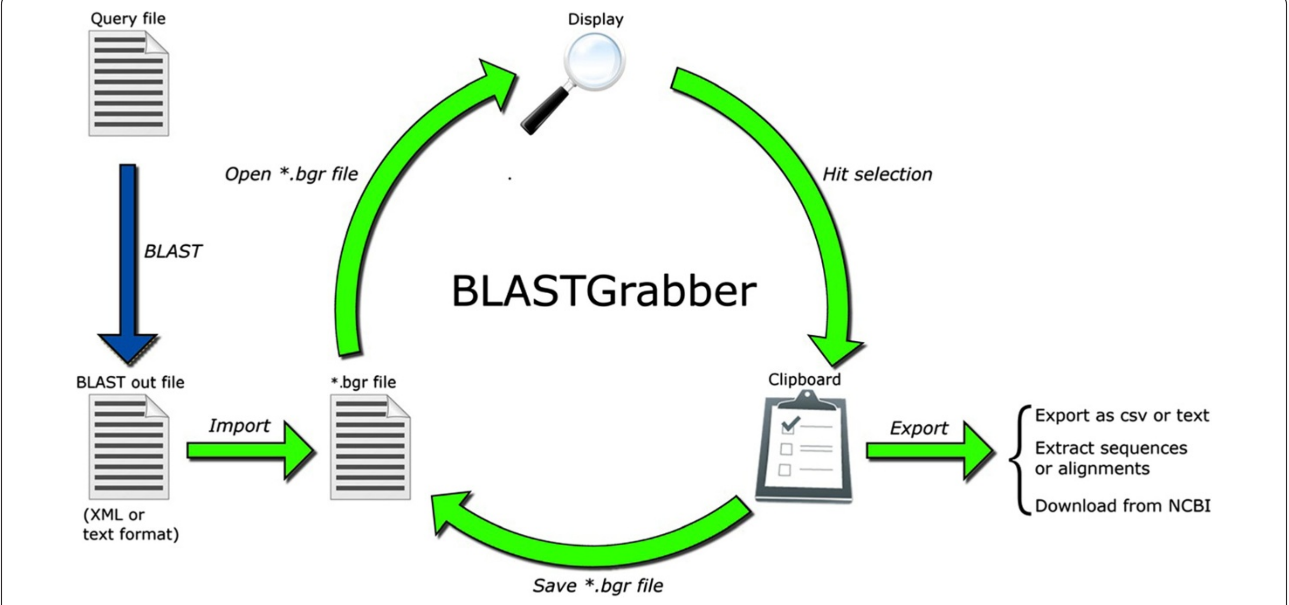


Figure 1 Workflow and interaction of the BLASTGrabber program. The text- or XML-formatted BLAST output can be imported directly into BLASTGrabber. BLASTGrabber supports an iterative mode of analysis, with repeated display, selection, saving and re-opening of BLAST data. Sequences of interest can be exported or downloaded.

format, in addition to standard text-formatted BLAST output from both the current BLAST + version and the older BLAST 2.2.* versions. As taxonomy information is not included in the standard BLAST output, taxonomy identifiers will not be included when importing BLAST files.

The *.bgr file is formatted as a text file, and lists the included queries and sequences (names or headers together with internal identifiers). Multiple hit sequences with identical FASTA headers are represented only once. In addition, the file contains BLAST statistics such as e-values and bit scores. NCBI TAXIDs can be appended to the *.bgr file after importing by using the included taxonomy wizard. These allow a downstream taxonomical representation in the BLASTGrabber program.

Upon loading a *.bgr file, the query, sequence and high scoring pair (HSP) attribute objects are instantiated from the *.bgr file and kept in the computer memory until the program is terminated (or another *.bgr file is loaded). Since all the BLAST data is directly available in computer memory, most BLASTGrabber operations complete in a few seconds at the most. Therefore installed memory limits the maximum BLAST data size usable by BLASTGrabber. For BLAST output exceeding these memory limitations, it is possible to break apart the offending BLAST file into multiple smaller parts. These can be imported and opened individually, and possibly merged together again at a later point in the analysis. The effectual maximum BLAST output size can be doubled using this technique.

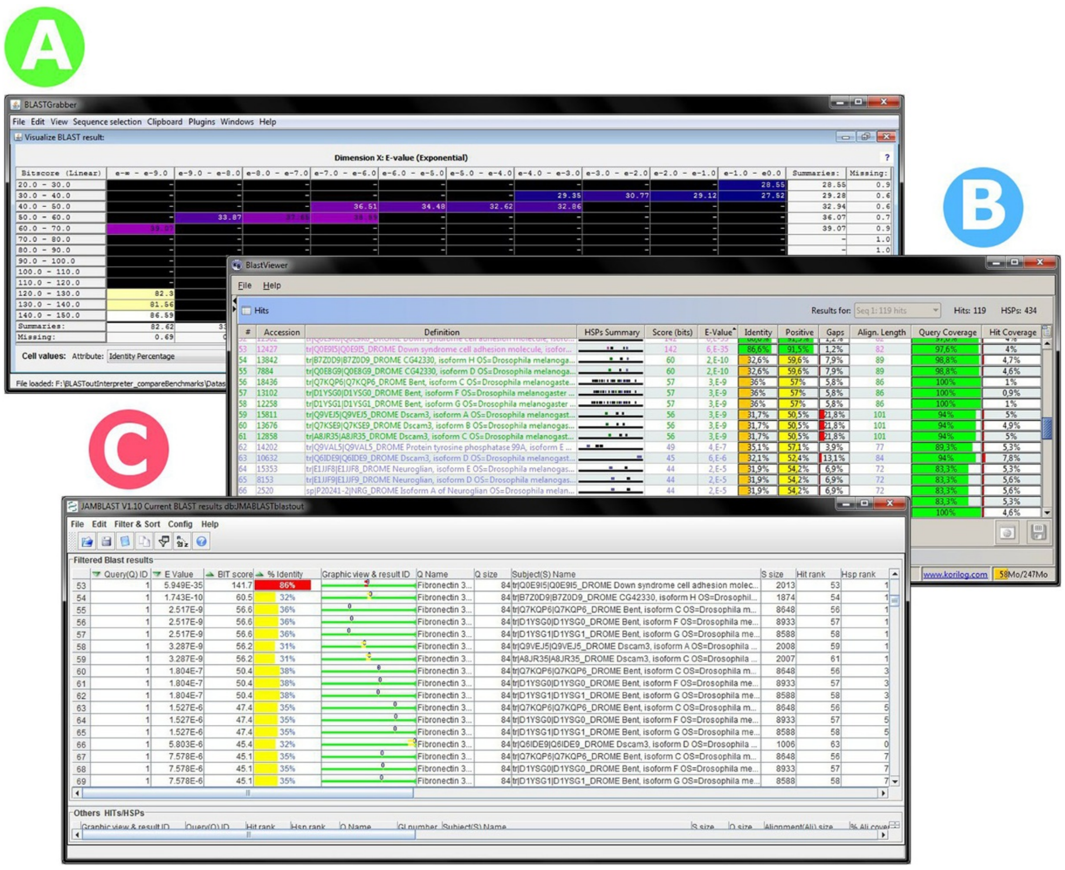
The BLASTGrabber installation folder includes the NCBI taxonomy structure compressed into a custom binary format. In order to keep the start-up time and memory overhead as low as possible, only scientific taxonomy names are included. Loading the taxonomy happens automatically at BLASTGrabber start-up, but can be cancelled by the user in order to preserve memory or improve BLASTGrabber start-up time. The BLASTGrabber taxonomy file can be updated by downloading the NCBI taxonomy files and using the included update functionality.

BLASTGrabber includes text search options; these are implemented using the Java regular expression (regex) API. Graphical representations of hit alignments are based on HSP attributes describing the start and end of the BLAST-generated alignments. No mismatches or gaps are indicated since this information is not included in the *.bgr file. However, sequences and alignments for selected hits can be downloaded from NCBI or extracted from the original BLAST output and query files.

BLASTGrabber architecture supports third party plugin development not depending on BLASTGrabber source code. Communication to and from the plugin is defined by two specific interfaces; JAR folders in the “plugins” subfolder featuring these required interfaces are loaded and activated at BLASTGrabber start-up (API description and plugin example are provided at the BLASTGrabber download site).

Example datasets

Two queries representing the Fibronectin-3 (Pfam id: PF00041) and I-set (Pfam id: PF07679) domains on the *Drosophila melanogaster* “Down syndrome cell adhesion


A
B
C
BLASTGrabber
File Edit View Sequence selection Clipboard Plugins Windows Help
Visualize BLAST result:
Cell values: Attribute: Identity Percentage
File loaded: F:\BLASToutInterpreter_compareBenchmarks\Datas
BlastViewer
File Help
Hits
Results for: Seq 1: 119 hits
Hits: 119 HSPs: 434
www.korilog.com
58Mo/247Mo
JAMBLAST V1.10 Current BLAST results db:JMABLASTblastout
File Edit Filter & Sort Config Help
Filtered Blast results
Others HITs/HSPs

Dimension X: E-value (Exponential)

Bitscore (Linear)	e-∞ - e-9.0	e-9.0 - e-8.0	e-8.0 - e-7.0	e-7.0 - e-6.0	e-6.0 - e-5.0	e-5.0 - e-4.0	e-4.0 - e-3.0	e-3.0 - e-2.0	e-2.0 - e-1.0	e-1.0 - e0.0	Summaries:	Missing:
20.0 - 30.0	-	-	-	-	-	-	-	-	-	28.55	28.55	0.9
30.0 - 40.0	-	-	-	-	-	-	29.35	30.77	29.12	27.52	29.28	0.6
40.0 - 50.0	-	-	-	36.51	34.48	32.62	32.86	-	-	-	32.94	0.6
50.0 - 60.0	-	33.87	37.65	38.89	-	-	-	-	-	-	36.07	0.7
60.0 - 70.0	39.07	-	-	-	-	-	-	-	-	-	39.07	0.9
70.0 - 80.0	-	-	-	-	-	-	-	-	-	-	-	1.0
80.0 - 90.0	-	-	-	-	-	-	-	-	-	-	-	1.0
90.0 - 100.0	-											
100.0 - 110.0	-											
110.0 - 120.0	-											
120.0 - 130.0	82.3											
130.0 - 140.0	81.56											
140.0 - 150.0	86.59											
Summaries:	82.62	33										
Missing:	0.69											

#	Accession	Definition	Score (bits)	E-Value	Identity	Positive	Gaps	Align. Length	Query Coverage	Hit Coverage
53	12427	tr|Q0E9I5|Q0E9I5_DROME Down syndrome cell adhesion molecule, isofor...	142	6,E-35	86,6%	91,5%	1,2%	82	97,6%	4%
54	13842	tr|B7Z0D9|B7Z0D9_DROME CG42330, isoform H OS=Drosophila melanoga...	60	2,E-10	32,6%	59,6%	7,9%	89	98,8%	4,7%
55	7884	tr|Q0E8G9|Q0E8G9_DROME CG42330, isoform D OS=Drosophila melanoga...	60	2,E-10	32,6%	59,6%	7,9%	89	98,8%	4,6%
56	18436	tr|Q7KQP6|Q7KQP6_DROME Bent, isoform C OS=Drosophila melanogaste...	57	3,E-9	36%	57%	5,8%	86	100%	1%
57	13102	tr|D1YSG0|D1YSG0_DROME Bent, isoform F OS=Drosophila melanogaster ...	57	3,E-9	36%	57%	5,8%	86	100%	0,9%
58	12258	tr|D1YSG1|D1YSG1_DROME Bent, isoform G OS=Drosophila melanogaster ...	57	3,E-9	36%	57%	5,8%	86	100%	1%
59	15811	tr|Q9VEJ5|Q9VEJ5_DROME Dscam3, isoform A OS=Drosophila melanogast...	56	3,E-9	31,7%	50,5%	21,8%	101	94%	5%
60	13676	tr|Q7KSE9|Q7KSE9_DROME Dscam3, isoform B OS=Drosophila melanogas...	56	3,E-9	31,7%	50,5%	21,8%	101	94%	4,9%
61	12858	tr|A8JR35|A8JR35_DROME Dscam3, isoform C OS=Drosophila melanogast...	56	3,E-9	31,7%	50,5%	21,8%	101	94%	5%
62	14202	tr|Q9VAL5|Q9VAL5_DROME Protein tyrosine phosphatase 99A, isoform E ...	49	4,E-7	35,1%	57,1%	3,9%	77	89,3%	5,3%
63	10632	tr|Q6IDE9|Q6IDE9_DROME Dscam3, isoform D OS=Drosophila melanogast...	45	6,E-6	32,1%	52,4%	13,1%	84	94%	7,8%
64	15353	tr|E1JJF8|E1JJF8_DROME Neuroglian, isoform E OS=Drosophila melanogas...	44	2,E-5	31,9%	54,2%	6,9%	72	83,3%	5,3%
65	8153	tr|E1JJF9|E1JJF9_DROME Neuroglian, isoform D OS=Drosophila melanogas...	44	2,E-5	31,9%	54,2%	6,9%	72	83,3%	5,6%
66	2520	sp|P20241-2|NRG_DROME Isoform A of Neuroglian OS=Drosophila melan...	44	2,E-5	31,9%	54,2%	6,9%	72	83,3%	5,6%

	Query(Q) ID	E Value	BIT score	% Identity	Q Name	Q size	Subject(S) Name	S size	Hit rank	Hsp rank
53	1	5.949E-35	141.7	86%	Fibronectin 3...	84	tr|Q0E9I5|Q0E9I5_DROME Down syndrome cell adhesion molec...	2013	53	1
54	1	1.743E-10	60.5	32%	Fibronectin 3...	84	tr|B7Z0D9|B7Z0D9_DROME CG42330, isoform H OS=Drosophil...	1874	54	1
55	1	2.517E-9	56.6	36%	Fibronectin 3...	84	tr|Q7KQP6|Q7KQP6_DROME Bent, isoform C OS=Drosophila m...	8648	56	1
56	1	2.517E-9	56.6	36%	Fibronectin 3...	84	tr|D1YSG0|D1YSG0_DROME Bent, isoform F OS=Drosophila me...	8933	57	1
57	1	2.517E-9	56.6	36%	Fibronectin 3...	84	tr|D1YSG1|D1YSG1_DROME Bent, isoform G OS=Drosophila me...	8588	58	1
58	1	3.287E-9	56.2	31%	Fibronectin 3...	84	tr|Q9VEJ5|Q9VEJ5_DROME Dscam3, isoform A OS=Drosophila ...	2008	59	1
59	1	3.287E-9	56.2	31%	Fibronectin 3...	84	tr|A8JR35|A8JR35_DROME Dscam3, isoform C OS=Drosophila ...	2007	61	1
60	1	1.804E-7	50.4	38%	Fibronectin 3...	84	tr|Q7KQP6|Q7KQP6_DROME Bent, isoform C OS=Drosophila m...	8648	56	3
61	1	1.804E-7	50.4	38%	Fibronectin 3...	84	tr|D1YSG0|D1YSG0_DROME Bent, isoform F OS=Drosophila me...	8933	57	3
62	1	1.804E-7	50.4	38%	Fibronectin 3...	84	tr|D1YSG1|D1YSG1_DROME Bent, isoform G OS=Drosophila me...	8588	58	3
63	1	1.527E-6	47.4	35%	Fibronectin 3...	84	tr|Q7KQP6|Q7KQP6_DROME Bent, isoform C OS=Drosophila m...	8648	56	5
64	1	1.527E-6	47.4	35%	Fibronectin 3...	84	tr|D1YSG0|D1YSG0_DROME Bent, isoform F OS=Drosophila me...	8933	57	5
65	1	1.527E-6	47.4	35%	Fibronectin 3...	84	tr|D1YSG1|D1YSG1_DROME Bent, isoform G OS=Drosophila me...	8588	58	5
66	1	5.803E-6	45.4	32%	Fibronectin 3...	84	tr|Q6IDE9|Q6IDE9_DROME Dscam3, isoform D OS=Drosophila ...	1006	63	0
67	1	7.578E-6	45.1	35%	Fibronectin 3...	84	tr|Q7KQP6|Q7KQP6_DROME Bent, isoform C OS=Drosophila m...	8648	56	7
68	1	7.578E-6	45.1	35%	Fibronectin 3...	84	tr|D1YSG0|D1YSG0_DROME Bent, isoform F OS=Drosophila me...	8933	57	7
69	1	7.578E-6	45.1	35%	Fibronectin 3...	84	tr|D1YSG1|D1YSG1_DROME Bent, isoform G OS=Drosophila me...	8588	58	7

Figure 2 The BLASTGrabber, BlastViewer and JAMBLAST program ordering BLAST hits according to high-scoring hit attributes. A) Hits are sorted according to e-values and bit scores along the horizontal and vertical axes in the BLASTGrabber matrix viewer. The average identity percentages are displayed inside the table cells. Heat map cell rendering has been applied, reflecting the identity percentage magnitudes. **B)** In the BlastViewer program, the hits for the selected query ("Seq 1") are sorted by clicking on the corresponding column headers. **C)** The JAMBLAST program supports explicit multiple sorting, as indicated by the arrows shown in the corresponding column headers. Note the additional sorting on the query id column header necessary due to JAMBLAST displaying both query and hit information on the same table row.

molecule" protein (uniprot id: A1Z6X1_DROME) were used in a BLAST search against the *Drosophila* uniprot reference proteome. These two motifs are found multiple times next to each other in various *Drosophila* multi-domain proteins and are well suited to illustrated sequence alignment visualization in BLASTGrabber.

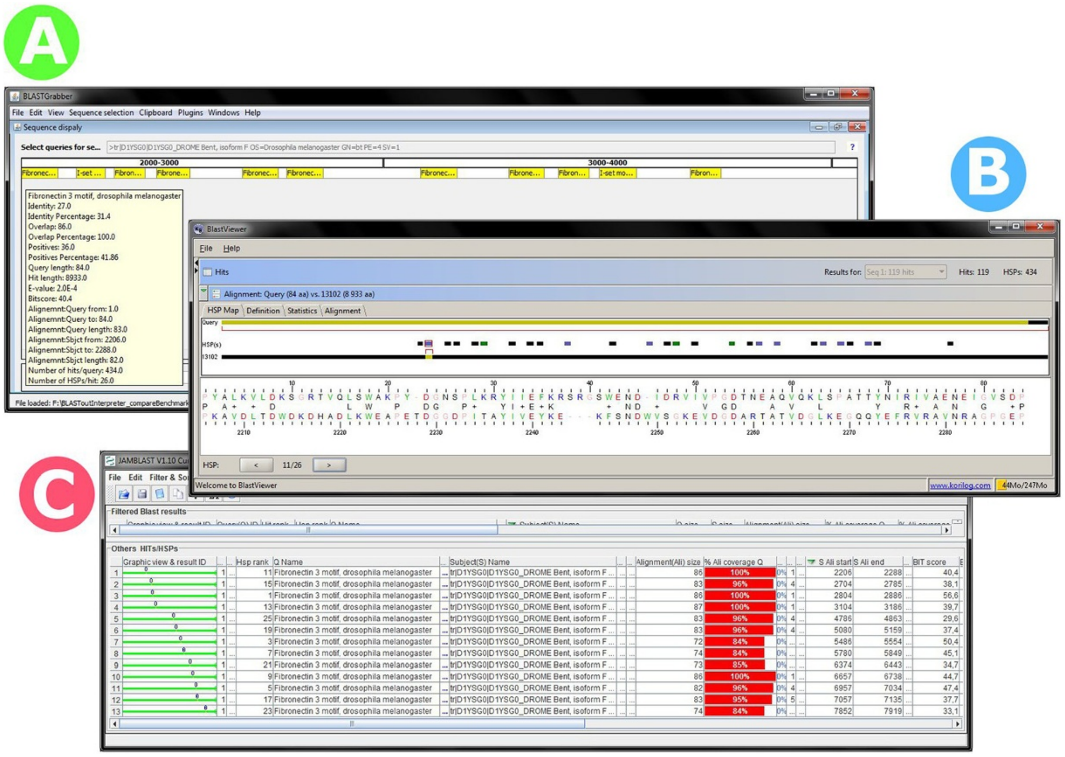
The BLASTGrabber visualization and analysis of the example datasets was compared to the JAMBLAST [29] and BlastViewer (Korilog SARL, Questembert, France) programs. As JAMBLAST utilizes a customary csv format (*.bls) produced by the accompanying NOBLAST [29] BLAST implementation, identical searches were performed producing the required BLAST output formats. In addition to the NOBLAST *.bls format required by JAMBLAST, BLAST XML output is required for BlastViewer.

In order demonstrate taxonomy display, a BLAST search (using the BLOSUM45 matrix; expect threshold at e-value = 1) was performed against the NCBI "nr" database using the two "Fibronectin-3" and "I-set" queries mentioned above.

Results and discussion

User-friendly visualization and analysis

Despite the continuing high popularity of the BLAST algorithm and ever-increasing data loads, there is a perceived lack of high-throughput-capable, user-friendly generic BLAST output visualisation tools. BLASTGrabber is designed to be usable by computer non-experts and implements a truly multi-query mode-of-analysis which is backed by visual displays. The included taxonomy wizard adds taxonomy identifiers to the BLAST data, allowing the downstream integration of sequence descriptions, origins and hit quality assessment. Also, the BLASTGrabber architecture facilitates the development of third-party plugins, opening new possibilities for BLAST-related analysis algorithms. Some aspects of BLASTGrabber overlap with the JAMBLAST and BlastViewer programs



Fibronectin 3 motif, drosophila melanogaster
Identity: 27.0
Identity Percentage: 31.4
Overlap: 86.0
Overlap Percentage: 100.0
Positives: 36.0
Positives Percentage: 41.86
Query length: 84.0
Hit length: 8933.0
E-value: 2.0E-4
Bitscore: 40.4
Alignemnt:Query from: 1.0
Alignemnt:Query to: 84.0
Alignemnt:Query length: 83.0
Alignemnt:Sbjct from: 2206.0
Alignemnt:Sbjct to: 2288.0
Alignemnt:Sbjct length: 82.0
Number of hits/query: 434.0
Number of HSPs/hit: 26.0

File loaded: F:\BLASToutInterpreter_compareBenchmark

	Graphic view & result ID	...	...	Hsp rank	Q Name	...	Subject(S) Name	...	...	Alignment(Ali) size	% Ali coverage Q	...	...	...	S Ali start	S Ali end	...	BIT score
1	0	1	...	11	Fibronectin 3 motif, drosophila melanogaster	...	tr|D1YSG0|D1YSG0_DROME Bent, isoform F ...	...	...	86	100%	0%	1	...	2206	2288	...	40,4
2	0	1	...	15	Fibronectin 3 motif, drosophila melanogaster	...	tr|D1YSG0|D1YSG0_DROME Bent, isoform F ...	...	...	83	96%	0%	4	...	2704	2785	...	38,1
3	0	1	...	1	Fibronectin 3 motif, drosophila melanogaster	...	tr|D1YSG0|D1YSG0_DROME Bent, isoform F ...	...	...	86	100%	0%	1	...	2804	2886	...	56,6
4	0	1	...	13	Fibronectin 3 motif, drosophila melanogaster	...	tr|D1YSG0|D1YSG0_DROME Bent, isoform F ...	...	...	87	100%	0%	1	...	3104	3186	...	39,7
5	0	1	...	25	Fibronectin 3 motif, drosophila melanogaster	...	tr|D1YSG0|D1YSG0_DROME Bent, isoform F ...	...	...	83	96%	0%	4	...	4786	4863	...	29,6
6	0	1	...	19	Fibronectin 3 motif, drosophila melanogaster	...	tr|D1YSG0|D1YSG0_DROME Bent, isoform F ...	...	...	83	96%	0%	4	...	5080	5159	...	37,4
7	0	1	...	3	Fibronectin 3 motif, drosophila melanogaster	...	tr|D1YSG0|D1YSG0_DROME Bent, isoform F ...	...	...	72	84%	0%	...	...	5486	5554	...	50,4
8	0	1	...	7	Fibronectin 3 motif, drosophila melanogaster	...	tr|D1YSG0|D1YSG0_DROME Bent, isoform F ...	...	...	74	84%	0%	...	...	5780	5849	...	45,1
9	0	1	...	21	Fibronectin 3 motif, drosophila melanogaster	...	tr|D1YSG0|D1YSG0_DROME Bent, isoform F ...	...	...	73	85%	0%	...	...	6374	6443	...	34,7
10	0	1	...	9	Fibronectin 3 motif, drosophila melanogaster	...	tr|D1YSG0|D1YSG0_DROME Bent, isoform F ...	...	...	86	100%	0%	1	...	6657	6738	...	44,7
11	0	1	...	5	Fibronectin 3 motif, drosophila melanogaster	...	tr|D1YSG0|D1YSG0_DROME Bent, isoform F ...	...	...	82	96%	0%	4	...	6957	7034	...	47,4
12	0	1	...	17	Fibronectin 3 motif, drosophila melanogaster	...	tr|D1YSG0|D1YSG0_DROME Bent, isoform F ...	...	...	83	95%	0%	5	...	7057	7135	...	37,7
13	0	1	...	23	Fibronectin 3 motif, drosophila melanogaster	...	tr|D1YSG0|D1YSG0_DROME Bent, isoform F ...	...	...	74	84%	0%	...	...	7852	7919	...	33,1

Figure 3 Comparing the graphical representations of BLAST alignment by the BLASTGrabber, BlastViewer and JAMBLAST programs. A) HSP alignments from both "Fibronectin-3 motif" and "I-Set motif" queries aligned against the "D1YSG0_DROME Bent" protein sequence in the BLASTGrabber program. Triggered by positioning the mouse over it, details for the first HSP are displayed. Based on a selected HSP attribute such as e-values or bit scores, the HSP representations can be colour-coded as a heat map (not shown because this would hide the HSP query names) **B)** HSP for the "Fibronectin-3 motif" query aligned against same protein in the BlastViewer program. In this program, HSPs for multiple queries cannot be displayed in the same graphical representation. The HSP overlaps are colour-coded (purple, green and black) so as to reflect significance categories. **C)** The same HSPs ("Fibronectin-3 motif" vs. "D1YSG0_DROME Bent") as displayed by the JAMBLAST program. Again, alignments from multiple queries cannot be displayed along the same sequence.

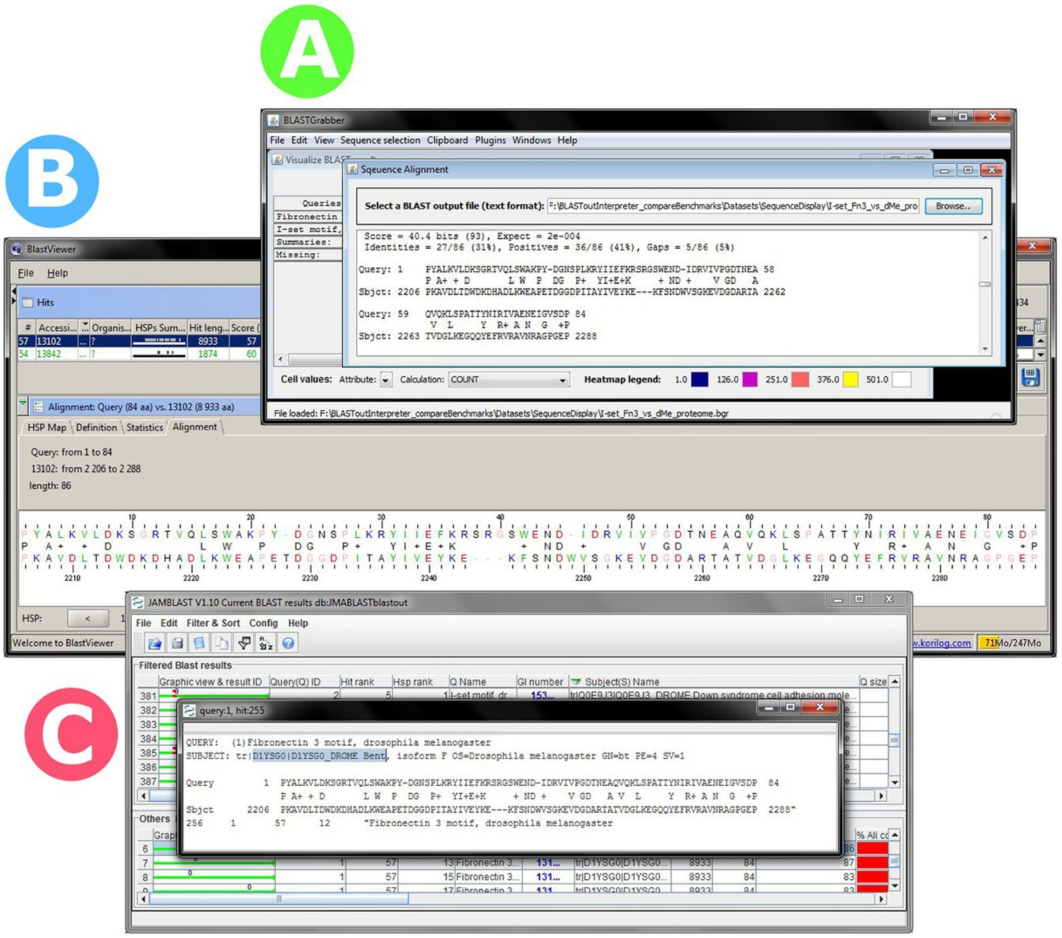
functionalities. Just as BLASTGrabber, both are free and user-friendly Java programs that visualize BLAST files. While supporting many additional analysis features when compared to existing similar programs, we find BLASTGrabber to perform as well or better in terms of data input capacity and speed, all the time retaining minimal installation requirements and an intuitive and user-friendly interface.

BLASTGrabber uses a matrix viewer of hits and queries

BLAST-related programs often display hits and queries as a hierarchical structure, where each query is associated with an ordered list of hits for that query. Each hit in turn can possibly contain multiple HSPs for that hit and query; often graphical representations of the hit alignments are displayed. Ordering of hit and HSP lists is done by some measurement of similarity (often configurable), such as e-values or bit scores. This analytical structure, which demands the selection of individual queries to inspect associated hits, is so pervasive it has been claimed to be almost without alternatives [20].

The visualisation structure adopted in BLASTGrabber differs fundamentally from this hierarchical approach. Using the matrix viewer (Figure 2A), hits are distributed in table rows and columns depending on two selected HSP attributes (in this example, two measures of similarity such as e-values, bit scores etc.). The cell content thus represents all hits (i.e. queries with their associated sequences) that conform to the cell specifics (for instance, e-values between e-5 and e-10 and bit scores between 200 and 250). A third attribute statistic is displayed in the cells themselves, the default being the number of hits therein, but other statistics such as average identities are available. Heat map rendering based on these values can be applied. Row and column sort in response to header clicks; lexicographical sorting is implicitly possible by selecting the desired attributes for row and column intervals, in addition to cell statistics.

This hit-focused (in contrast to query-focused) display mode allows the inspection of hundreds of thousands of hits at a glance, distributed across a few table cells only. Cell contents (i.e. the queries and hits) are easily


A
B
C
BLASTGrabber
File Edit View Sequence selection Clipboard Plugins Windows Help
Sqeuence Alignment
Select a BLAST output file (text format):
Browse..
Cell values: Attribute: Calculation: COUNT
Heatmap legend: 1.0 126.0 251.0 376.0 501.0
BlastViewer
File Help
Hits
HSP Map Definition Statistics Alignment
Welcome to BlastViewer
JAMBLAST V1.10 Current BLAST results db:JMABLASTblastout
File Edit Filter & Sort Config Help
Filtered Blast results
query:1, hit:255

```
Score = 40.4 bits (93), Expect = 2e-004
Identities = 27/86 (31%), Positives = 36/86 (41%), Gaps = 5/86 (5%)

Query: 1    PYALKVLDKSGRTVQLSWAKPY-DGNSPLKRYIIEFKRSRGSWEND-IDRVIVPGDTNEA 58
            P A+ + D       L W  P  DG  P+  YI+E+K     + ND +    V GD   A
Sbjct: 2206 PKAVDLTDWDKDHADLKWEAPETDGGDPITAYIVEYKE---KFSNDWVSGKEVDGDARTA 2262

Query: 59   QVQKLSPATTYNIRIVAENEIGVSDP 84
             V  L     Y  R+ A N  G  +P
Sbjct: 2263 TVDGLKEGQQYEFRVRAVNRAGPGEP 2288
```

File loaded: F:\BLASToutInterpreter_compareBenchmarks\Datasets\SequenceDisplay\I-set_Fn3_vs_dMe_proteome.bgr

Alignment: Query (84 aa) vs. 13102 (8 933 aa)

Query: from 1 to 84
13102: from 2 206 to 2 288
length: 86

```
QUERY:   (1)Fibronectin 3 motif, drosophila melanogaster
SUBJECT: tr|D1YSG0|D1YSG0_DROME Bent, isoform F OS=Drosophila melanogaster GN=bt PE=4 SV=1

Query        1   PYALKVLDKSGRTVQLSWAKPY-DGNSPLKRYIIEFKRSRGSWEND-IDRVIVPGDTNEAQVQKLSPATTYNIRIVAENEIGVSDP  84
                 P A+ + D       L W  P  DG  P+  YI+E+K     + ND +    V GD   A V  L     Y  R+ A N  G  +P
Sbjct      2206  PKAVDLTDWDKDHADLKWEAPETDGGDPITAYIVEYKE---KFSNDWVSGKEVDGDARTATVDGLKEGQQYEFRVRAVNRAGPGEP  2288"
256      1      57      12      "Fibronectin 3 motif, drosophila melanogaster
```

Figure 4 BLAST alignment sequences as presented by the three BLAST interpreter programs. A) Alignment details for a selected hit can be extracted from the original BLAST output file in BLASTGrabber. **B)** After selecting the hit for a given query, alignment details for the HSPs are given in the lower half of the BlastViewer screen. The alignment details cannot, however, be copied or exported. **C)** Likewise, alignment details for a select hit can be visualized in the JAMBLAST program if the actual alignments have been included in the custom NOBLAST output.

selected, either by manually clicking on cells of interest, or else defining cut-off values for cell statistics (for instance selecting all hit with e-values below e-5). Selected cells are "grabbed", i.e. copied to the BLASTGrabber clipboards. Clipboard contents are persistable as *.bgr files, allowing iterative purifying selection by *.bgr file opening, subset selection and persistence.

In contrast to BLASTGrabber, the BlastViewer program displays hits and queries as a hierarchical structure. When using multiple-query BLAST output each query must be selected before the associated hits are displayed (Figure 2B). The hit list is sortable by any HSP attribute by simply clicking a column header; top-scoring hits can be exported to a text editor in csv format. Lexicographical sorting (for instance based on e-values, bit scores and identity percentages in this priority) can be accomplished by clicking column headers in the reverse order of the desired priority. BlastViewer visualizes all the BLAST attributes for each of the hits, but because queries have to be selected one at the time without any filtering options, it becomes virtually impossible to use for analysis of thousands, or even hundreds of thousand, queries.

JAMBLAST also employs a hierarchical structure but has a slightly different visualization mode than BlastViewer, displaying all queries and top scoring hits in the same table (Figure 2C). The program supports lexicographical ordering by the explicit selection of the HSP attributes sort order. The best hits for multiple queries can be selected in one go with subsequent export to a text editor program. Without clustering multiple hits into fewer categories (such as dividing hits into significant and insignificant e-value categories), it is virtually impossible to extract relevant information from high-throughput data.

Alignment visualization

Most BLAST output interpreter programs provide a graphical representation of the BLAST-generated alignment, i.e. the sequence overlap between query and hit. In addition to displaying hits as they are aligned against a selected query, BLASTGrabber can also do the opposite; i.e. display multiple queries overlapping one selected hit. Thus, for instance the position of multiple genes (queries) along one chromosome (one hit) can be displayed.

"Grabbing" is integrated across BLASTGrabber viewers; hits represented by the displayed alignments can be copied to the clipboards just as for the matrix viewer. Both BLASTGrabber and BlastViewer support a colour-coding schema reflecting the statistical significance of the hit (in BLASTGrabber this is part of a generic heat map rendering based on any selected HSP attribute).

BlastViewer (Figure 3B) restricts alignment visualization to one query/sequences pair, displaying all HSPs as coloured bars along the sequence representation. JAMBLAST (Figure 3C) includes just one such alignment at the time; multiple hits for one query cannot be included in one graphical representation. Neither of these two programs can display multiple queries for one selected hit (an included option in BLASTGrabber).

All three programs can include the display of alignments in text format (see Figure 4). In BLASTGrabber the alignment is extracted from the original BLAST output file; the JAMBLAST database includes alignments only if the relevant option was included when running NOBLAST. The BlastViewer alignment is read-only and cannot be copied.

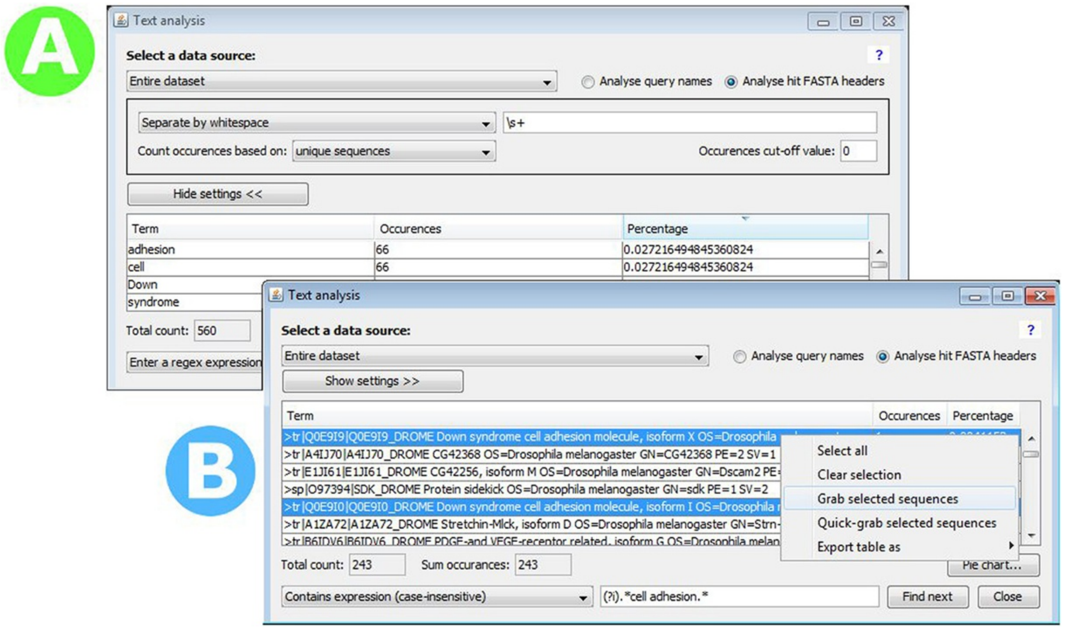
Text searches with the description viewer

JAMBLAST and BlastViewer, like most BLAST output interpreter programs, do not support text searches in the sequence FASTA headers. The BLASTGrabber description viewer (Figure 5) however, supports flexible text search with subsequent hit selection in both headers and BLAST query names. Using regex search terms, matching entries can be selected and copied to BLASTGrabber clipboards. Suitable regex templates are provided, demanding no prior knowledge of Java regex syntax.

In addition to performing searches in the complete headers or query names, these can be split into individual words. This allows determination of the relative frequencies of keywords of interest, such as enzyme or species names. Word splitting is regex-based and fully configurable.

Taxonomical ordering of data in the matrix viewer

All biological sequences reflect evolutionary events such as adaption, speciation or separation. Thus they are best classified according to a taxonomical order believed to fit these events. The classification of organisms has been revolutionized by DNA sequence analysis, which has become one of the primary means of taxonomic identification of species [30]. Besides the obvious goal of constructing phylogenies and an evolutionary tree of life, taxonomical sequence information can be utilized in many diverse tasks. These include ecological studies and metagenomics of un-



A)

Select a data source: Entire dataset
Analyse query names [✓ Analyse hit FASTA headers]
Separate by whitespace \s+
Count occurences based on: unique sequences
Occurences cut-off value: 0

Term	Occurences	Percentage
adhesion	66	0.027216494845360824
cell	66	0.027216494845360824
Down		
syndrome		

Total count: 560

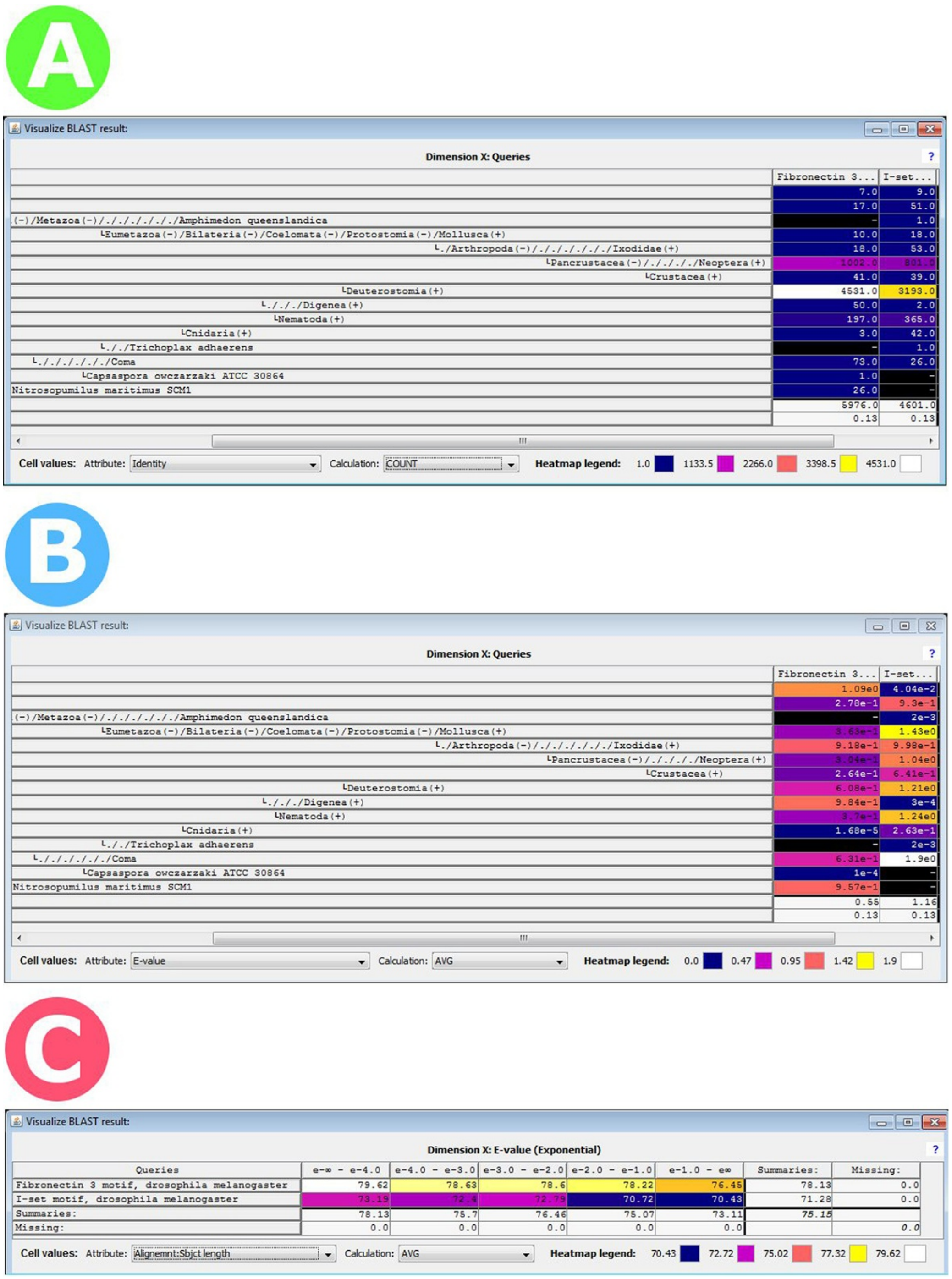
B)

Select a data source: Entire dataset
Analyse query names [✓ Analyse hit FASTA headers]

Term	Occurences	Percentage
>tr|Q0E9I9|Q0E9I9_DROME Down syndrome cell adhesion molecule, isoform X OS=Drosophila		
>tr|A4IJ70|A4IJ70_DROME CG42368 OS=Drosophila melanogaster GN=CG42368 PE=2 SV=1		
>tr|E1JI61|E1JI61_DROME CG42256, isoform M OS=Drosophila melanogaster GN=Dscam2 PE		
>sp|O97394|SDK_DROME Protein sidekick OS=Drosophila melanogaster GN=sdk PE=1 SV=2		
>tr|Q0E9I0|Q0E9I0_DROME Down syndrome cell adhesion molecule, isoform I OS=Drosophila		
>tr|A1ZA72|A1ZA72_DROME Stretchin-Mlck, isoform D OS=Drosophila melanogaster GN=Strn-		
>tr|B6IDV6|B6IDV6_DROME PDGF-and VEGF-receptor related, isoform G OS=Drosophila melan		

Total count: 243 Sum occurances: 243
Contains expression (case-insensitive) (?i).*cell adhesion.*

Figure 5 The BLASTGrabber description viewer. A) The descriptions in the sequence headers are broken apart by whitespaces, allowing a per-word analysis of the BLAST results. Many hits clearly belong to the "Cell adhesion" and "Down syndrome" protein categories. **B)** Unique headers are displayed in the description viewer. Using a regular expression search term (user-friendly regex templates are provided), all headers containing the "cell adhesion" term are selected, and can now be copied to the BLASTGrabber clipboard.



Dimension X: Queries

	Fibronectin 3...	I-set...
	7.0	9.0
	17.0	51.0
(-)/Metazoa(-)/./././././././Amphimedon queenslandica	-	1.0
└Eumetazoa(-)/Bilateria(-)/Coelomata(-)/Protostomia(-)/Mollusca(+)	10.0	18.0
└./Arthropoda(-)/./././././././Ixodidae(+)	18.0	53.0
└Pancrustacea(-)/./././././Neoptera(+)	1002.0	801.0
└Crustacea(+)	41.0	39.0
└Deuterostomia(+)	4531.0	3193.0
└./././Digenea(+)	50.0	2.0
└Nematoda(+)	197.0	365.0
└Cnidaria(+)	3.0	42.0
└././Trichoplax adhaerens	-	1.0
└./././././././Coma	73.0	26.0
└Capsaspora owczarzaki ATCC 30864	1.0	-
Nitrosopumilus maritimus SCM1	26.0	-
	5976.0	4601.0
	0.13	0.13

Cell values: Attribute: Identity Calculation: COUNT Heatmap legend: 1.0 🟦 1133.5 🟪 2266.0 🟥 3398.5 🟨 4531.0 ⬜

Dimension X: Queries

	Fibronectin 3...	I-set...
	1.09e0	4.04e-2
	2.78e-1	9.3e-1
(-)/Metazoa(-)/./././././././Amphimedon queenslandica	-	2e-3
└Eumetazoa(-)/Bilateria(-)/Coelomata(-)/Protostomia(-)/Mollusca(+)	3.63e-1	1.43e0
└./Arthropoda(-)/./././././././Ixodidae(+)	9.18e-1	9.98e-1
└Pancrustacea(-)/./././././Neoptera(+)	3.04e-1	1.04e0
└Crustacea(+)	2.64e-1	6.41e-1
└Deuterostomia(+)	6.08e-1	1.21e0
└./././Digenea(+)	9.84e-1	3e-4
└Nematoda(+)	3.7e-1	1.24e0
└Cnidaria(+)	1.68e-5	2.63e-1
└././Trichoplax adhaerens	-	2e-3
└./././././././Coma	6.31e-1	1.9e0
└Capsaspora owczarzaki ATCC 30864	1e-4	-
Nitrosopumilus maritimus SCM1	9.57e-1	-
	0.55	1.16
	0.13	0.13

Cell values: Attribute: E-value Calculation: AVG Heatmap legend: 0.0 🟦 0.47 🟪 0.95 🟥 1.42 🟨 1.9 ⬜

Dimension X: E-value (Exponential)

Queries	e-∞ - e-4.0	e-4.0 - e-3.0	e-3.0 - e-2.0	e-2.0 - e-1.0	e-1.0 - e∞	Summaries:	Missing:
Fibronectin 3 motif, drosophila melanogaster	79.62	78.63	78.6	78.22	76.45	78.13	0.0
I-set motif, drosophila melanogaster	73.19	72.4	72.79	70.72	70.43	71.28	0.0
Summaries:	78.13	75.7	76.46	75.07	73.11	*75.15*	
Missing:	0.0	0.0	0.0	0.0	0.0		*0.0*

Cell values: Attribute: Alignemnt:Sbjct length Calculation: AVG Heatmap legend: 70.43 🟦 72.72 🟪 75.02 🟥 77.32 🟨 79.62 ⬜

Figure 6 The NCBI taxonomy as displayed by the BLASTGrabber matrix viewer. A) The taxonomic distribution for the two Fibronectin-3 and I-Set motifs as queried against the NCBI nr database, represented by the absolute number of hits per query/taxonomic unit. Heat map display mode has been chosen. Most taxonomic groups (such as "Deuterostomia" or "Nematoda") are collapsed, as indicated by the "+" (expandable) sign. The "Amphimedon Queenslandica", "Trichoplax adhaerens" and "Capsaspora owczarzaki" species are fully expanded. By far the most hits are found inside the "Deuterostomia" taxonomic group. **B)** Average e-values are displayed inside the same taxonomic setup. Many groups are characterized by rather dubious average e-values, such as the "Deuterostomia"-specific hits (6.08e-1 and 1.21e0 for the two queries). **C)** Analysis of the distribution of e-values for the two queries. Numbers of hits are shown for five e-value intervals. It is now possible to select the leftmost column (containing significant hits with e-values better than e-4), saving and re-loading the selection using the clipboard functionality, so as to recreate the above taxonomic setup with significant hits only.

Table 1 Maximum data capacities of the BLAST interpreter programs*

Program	Maximal capacity	Input file
BlastViewer	~1.200.000 sequences	XML file (BLAST XML format, size ~2.1 GB)
JAMBLAST	~1.000.000 sequences	bls file (NOBLAST csv format, size ~290 MB)
BLASTGrabber	~2.200.000 sequences	bgr file (custom text format, size ~ 280 MB)

*as tested on a laptop computer; 1.4 GHz Core i5 2537 M processor, 4 GB of memory, SSD hard drive. Java heap space was set to 4 GB. On average, BLAST data consisted of 100 sequences pr. query.

culturable micro-organisms [31], the removal of contamination (for instance sequences with bacterial origin) from shotgun or contig datasets, or the study of horizontal gene transfer. Taxonomy even facilitates tasks such as protein domain discovery [32] or protein fold recognition [33].

In BLASTGrabber, the NCBI taxonomy is fully integrated into the matrix viewer (Figure 6), displayed as an interactive tree structure. Along the matrix viewer vertical axis, taxonomy nodes can be expanded or collapsed at will, facilitating selection or taxonomic annotation at a level higher than the species level. The horizontal axis can harbour intervals for any BLAST HSP attribute, or even the queries or hit sequences themselves. Taxonomic distributions can easily be assessed by ordering the columns according to frequency counts or accumulative distributions, possibly using the heat map display mode. Hit selection and “grabbing” is fully supported. For instance hits identified as cnidarian sequences containing the “Fibronectin 3” motif can be “grabbed” by selecting the corresponding cell. Alternatively, undesired groups such as for instance “nematode” can be excluded from further analysis (Figure 6A and B). Neither BlastViewer nor JAMBLAST support this mode of analysis.

Splitting and merging files

Contemporary high-throughput projects may generate millions of sequences used as queries in BLAST similarity searches (for one example, see [13]). Depending on the installed computer memory, BLASTGrabber can handle BLAST output containing a few hundred thousand queries. If the relevant data set is too substantial to be imported into BLASTGrabber, the BLAST file can be split into smaller parts; each single part can then be imported individually into BLASTGrabber. Subsequently, several *.bgr files can be merged to join previously disjoint data sets.

Table 2 Performance examples of typical BLASTGrabber operations*

BLAST file	~90.000 queries (file size 640 MB)
Import (creation of *.bgr file)	~5 minutes
Resulting *.bgr file	~45.000 queries** (file size 140 MB).
Opening the *.bgr file	~30 seconds
Opening matrix viewer, 10 rows × 10 cols (100 cells)	~1 second
Opening matrix viewer, all queries × 10 cols (450.000 cells)	~2 seconds
Selecting a significant subset (e-value < e-5)	<1 second
Saving the selected significant subset	~10 seconds

*as tested on a laptop computer; 1.4 GHz Core i5 2537 M processor, 4 GB of memory, SSD hard drive.
**on average 23 hits pr. query; queries without hits were excluded from the BLASTGrabber input file.

BLASTGrabber performance

Given ever-increasing BLAST output sizes, the ability to handle such data on a normal desktop computer will be considered crucial by many users. In implementing BLASTGrabber, care was taken to minimize the data representation memory footprint. The validity of the approach was confirmed when comparing BLASTGrabber to the BlastViewer and JAMBLAST applications (Table 1). Compared to BlastViewer and JAMBLAST, maximum BLASTGrabber capacity was about twice as high.

Once BLAST data has been loaded, all three programs were quite responsive and performed most operations within seconds, even when being used close to their maximum capacities (for performance examples of typical BLASTGrabber operations, see Table 2). Data loading by itself also executed quite rapidly as long as the input file sizes did not exceed ~75 percent of the maximal capacity.

Conclusions

Along with the rest of bioinformatics, performing BLAST searches has changed substantially over the past two decades. The default textual BLAST output is well adapted to single-query BLAST runs, but next-generation sequencing technologies and growing database sizes demand advanced visualization and analysis capabilities. At the same time, an increasing number of biologists without expert knowledge of programming languages or database retrieval techniques need to work with BLAST and BLAST-related tools.

We believe that the BLASTGrabber program can serve as such a tool. It enables the handling of large amounts of data on desktop computers, while at the same time delivering new and advanced visualization styles. In addition, the flexible text-mining options provide easy selection of BLAST hits or queries. Finally, the integrated taxonomy display can add an extra dimension to traditional sequence similarity searches.

Availability and requirements

Project name: BLASTGrabber.

Project home page: http://www.bioportal.no.

This web address also hosts the BLASTGrabber user guide and a tutorial video demonstrating the use of BLASTGrabber. The BLASTGrabber zip file contains both sample files and the user guide.

Operating system: Platform independent.

Programming language: Java.

Other requirements: Java Runtime Environment (version 1.6 or higher).

License: Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License.

Any restrictions to use by non-academics: None.

Additional file

Additional file 1: BLASTGrabber User Guide.

Abbreviations

BLAST: Basic Local Alignment Search Tool; Csv: Comma-separated values; GB: Giga-bytes; JAR: Java Archive; HPC: High-performance computing; HSP: High-scoring pair; MB: Mega-bytes; NCBI: National Center for Biotechnology Information; Regex: Regular expressions; UoO: University of Oslo.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

Concept and design: RSN, SK, KST. Programming: RSN, SK. Testing: RSN, SK, THAH, KST. Drafting manuscript: RSN, SK. Critical revisions: RSN, SK, THAH, KST. Final manuscript approval: RSN, SK, THAH, KST.

Acknowledgements

We thank student and researchers at the Microbial Evolution Research Group (MERG) and Section for Genetics and Evolutionary Biology for testing of the BLASTGrabber program.

Funding

This work has been supported by a VISTA/Statoil fellowship to Thomas H. A. Haverkamp, PhD fellowships to Surendra Kumar and Ralf Neumann and starting grant to Kamran Shalchian-Tabrizi from the University of Oslo.

Author details

[1]Section for Genetics and Evolutionary Biology (EVOGENE) and Centre for Epigenetics, Development and Evolution (CEDE), University of Oslo, Oslo, Norway. [2]Current address: Department of Clinical Molecular Biology and Laboratory Science (EpiGen), Division of Medicine, Akershus University Hospital, 1478 Akershus, Norway. [3]Centre of Ecological and Evolutionary synthesis, Department of Biosciences, University of Oslo, Oslo, Norway.

Received: 7 January 2014 Accepted: 31 March 2014
Published: 5 May 2014

References

1. Nilsson R, Larsson K, Ursing BM: Galaxie—CGI scripts for sequence identification through automated phylogenetic analysis. *Bioinformatics* 2004, **20**:1447–1452.
2. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: Basic local alignment search tool. *J Mol Biol* 1990, **215**:403–410.
3. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, Lipman DJ: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 1997, **25**:3389–3402.
4. Kanehisa M, Bork P: Bioinformatics in the post-sequence era. *Nat Genet* 2003, **33**:305–310.
5. Koonin E: *Sequence-Evolution-Function: Computational Approaches in Comparative Genomics*. Dordrecht: Kluwer Academic Publishers; 2003.
6. Dong Q, Brendel V: Computational Identification of Related Proteins. In *The Proteomics Protocols Handbook*. Edited by Walker JM. Totowa: Humana Press; 2005:555–570.
7. She R, Shih-Chieh Chu J, Uyar B, Wang J, Wang K, Chen N: genBlastG: using BLAST searches to build homologous gene models. *Bioinformatics* 2011, **27**:2141–2143.
8. Dereeper A, Audic S, Claverie J-M, Blanc G: BLAST-EXPLORER helps you building datasets for phylogenetic analysis. *BMC Evol Biol* 2010, **10**:1–6.
9. Xing L, Brendel V: Multi-query sequence BLAST output examination with MuSeqBox. *Bioinformatics* 2001, **17**:744–745.
10. Pirooznia M, Perkins EJ, Deng Y: Batch Blast Extractor: an automated blastx parser application. *BMC Genomics* 2008, **9**:1–5.
11. Zhao S, Shao C, Goropashnaya AV, Stewart NC, Xu Y, Tøien Ø, Barnes BM, Fedorov VB, Yan J: Genomic analysis of expressed sequence tags in American black bear Ursus americanus. *BMC Genomics* 2010, **11**:201.
12. Koltes J, Hu Z, Fritz E: BEAP: The BLAST Extension and Alignment Program-a tool for contig construction and analysis of preliminary genome sequence. *BMC Res Notes* 2009, **2**:11.
13. Paquola ACM, Machado AA, Reis EM, da Silva AM, Verjovski-Almeida S: Zerg: a very fast BLAST parser library. *Bioinformatics* 2003, **19**:1035–1036.
14. Suyama M, Torrents D, Bork P: BLAST2GENE: a comprehensive conversion of BLAST output into independent genes and gene fragments. *Bioinformatics* 2004, **20**:1968–1970.
15. Wall DP, Fraser HB, Hirsh AE: Detecting putative orthologs. *Bioinformatics* 2003, **19**:1710–1711.
16. Zhao S, Burki F, Keeling P: Collodictyon – an ancient lineage in the tree of eukaryotes. *Mol Biol Evol* 2012, **29**:1557–1568.
17. Koski LB, Golding GB: The closest BLAST hit is often not the nearest neighbor. *J Mol Evol* 2001, **52**:540–542.
18. Shalchian-Tabrizi K, Minge MA, Espelund M, Orr R, Ruden T, Jakobsen KS, Cavalier-Smith T: Multigene phylogeny of choanozoa and the origin of animals. *PLoS ONE* 2008, **3**:1–7.
19. Huson DH, Auch AF, Qi J, Schuster SC: MEGAN analysis of metagenomic data. *Genome Res* 2007, **17**:377–386.
20. Procter JB, Thompson J, Letunic I, Creevey C, Jossinet F, Barton GJ: Visualization of multiple alignments, phylogenies and gene family evolution. *Nat Publ Group* 2010, **7**:S16–S25.
21. Mount DW: *Bioinformatics: Sequence and Genome Analysis*. New York: Cold Spring Harbor Laboratory Press; 2004.
22. Weintraub B: Building BLAST for Coprocessor Accelerators Using Macah. *PhD thesis*. University of Washington, Computer Science And Engineering; 2008.
23. Decker JE, Pires JC, Conant GC, McKay SD, Heaton MP, Chen K, Cooper A, Vilkki J, Seabury CM, Caetano AR: Resolving the evolution of extant and extinct ruminants with high-throughput phylogenomics. *Proc Natl Acad Sci* 2009, **106**:18644–18649.
24. O'Donoghue SI, Gavin A-C, Gehlenborg N, Goodsell DS, Hériché J-K, Nielsen CB, North C, Olson AJ, Procter JB, Shattuck DW, Walter T, Wong B: Visualizing biological data — now and in the future. *Nat Publ Group* 2010, **7**:S2–S4.
25. Fayyad U, Piatetsky-Shapiro G, Smyth P: From data mining to knowledge discovery in databases. *AI Mag* 1996, **17**:37–54.
26. Darzentas N: Circoletto: visualizing sequence similarity with circos. *Bioinformatics* 2010, **26**:2620–2621.
27. Otto A, Bernhardt J, Meyer H, Schaffer M, Herbst F-A, Siebourg J, Mäder U, Lalk M, Hecker M, Becher D: Systems-wide temporal proteomic profiling in glucose-starved Bacillus subtilis. *Nat Commun* 2010, **1**:1–9.
28. Krzywinski M, Birol I, Jones SJ, Marra M: Hive plots–rational approach to visualizing networks. *Brief Bioinform* 2011, **13**:1–18.
29. Lagnel J, Tsigenopoulos CS, Iliopoulos I: NOBLAST and JAMBLAST: New Options for BLAST and a Java Application Manager for BLAST results. *Bioinformatics* 2009, **25**:824–826.
30. Nilsson RH, Kristiansson E, Ryberg M, Larsson KH: Approaching the taxonomic affiliation of unidentified sequences in public databases–an example from the mycorrhizal fungi. *BMC Bioinforma* 2005, **6**:178.
31. Pozhitkov A: Molecular taxonomy. Bioinformatics and practical evaluation. *PhD thesis*. Universitet Köln, Mathematisch-Naturwissenschaftliche Fakultät; 2003.

32. Coin L, Bateman A: **Enhanced protein domain discovery using taxonomy.** *BMC Bioinforma* 2004, **5**:56.
33. Yang JY, Chen X: **Improving taxonomy-based protein fold recognition by using global and local features.** *Proteins Struct Funct Bioinformatics* 2011, **79**:2053–2064.

doi:10.1186/1471-2105-15-128
Cite this article as: Neumann *et al.*: BLASTGrabber: a bioinformatic tool for visualization, analysis and sequence selection of massive BLAST data. *BMC Bioinformatics* 2014 **15**:128.