Sequenceserver: a modern graphical user interface for custom BLAST databases

Anurag Priyam[1,2,11], Ben J. Woodcroft[3,11,12], Vivek Rai[4], Alekhya Munagala[5], Ismail Moghul[1], Filip Ter[6], Mark Anthony Gibbins[7], HongKee Moon[8], Guy Leonard[9], Wolfgang Rumpf[10], Yannick Wurm[1,11,13,*]

1: School of Biological and Chemical Sciences, Queen Mary University of London, London E1 4NS, United Kingdom.

2: Department of Mechanical Engineering, Indian Institute of Technology Kharagpur, Kharagpur 721302, West Bengal, India.

3: Australian Centre for Ecogenomics, School of Chemistry and Molecular Biosciences, University of Queensland, Brisbane 4072, Queensland, Australia.

4: Department of Biotechnology, Indian Institute of Technology Kharagpur, Kharagpur 721302, West Bengal, India.

5: Department of Mathematics, Indian Institute of Technology Kharagpur, Kharagpur 721302, West Bengal, India.

6: School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, United Kingdom.

7: Department of Computer Science, Royal Holloway, University of London, Egham, Surrey, TW20 0EX, United Kingdom.

8: Scientific Computing Facility, Max Planck Institute of Molecular Cell Biology and Genetics, Pfotenhauerstr 108, 01307 Dresden, Germany.

9: Biosciences, University of Exeter, Geoffrey Pope Building, Exeter EX4 4QD, United Kingdom.

10: The Graduate School, University of Maryland, University College, Largo, MD, 20774, United States; and The Research Institute at Nationwide Children's Hospital, Columbus, Ohio, 43205, United States

11: 5Bases Limited, London E9 7DR, United Kingdom.

12: Previous address: Department of Biochemistry and Molecular Biology, Bio21 Molecular Science and Biotechnology Institute, The University of Melbourne, Parkville, 3010, Victoria, Australia.

13: Previous address: Department of Ecology and Evolution and Swiss Institue of Bioinformatics, Université de Lausanne, 1015 Lausanne, Switzerland.

* To whom correspondence should be addressed: y.wurm@qmul.ac.uk

Abstract

The dramatic drop in DNA sequencing costs has created many opportunities for novel biological research. These opportunities largely rest upon the ability to effectively compare newly obtained and previously known sequences. This is commonly done with BLAST, yet using BLAST directly on new datasets requires substantial technical skills or helpful colleagues. Furthermore, graphical interfaces for BLAST are challenging to install and largely mimic underlying computational processes rather than work patterns of researchers.

We combined a user-centric design philosophy with sustainable software development approaches to create Sequenceserver (`http://sequenceserver.com`), a modern graphical user interface for BLAST. Sequenceserver substantially increases the efficiency of researchers working with sequence data. This is due to innovations at three levels. First, our software can be installed and used on custom datasets extremely rapidly for personal and shared applications. Second, based on analysis of user input and simple algorithms, Sequenceserver reduces the amount of decisions the user must make, provides interactive visual feedback, and prevents common potential errors that would otherwise cause erroneous results. Finally, Sequenceserver provides multiple highly visual and text-based output options that mirror the requirements and work patterns of researchers. Together, these features greatly facilitate BLAST analysis and interpretation and thus substantially enhance researcher productivity.

Introduction

The dramatic drop in cost of DNA sequencing since 2007 (`http://genome.gov/sequencingcosts`) has led to a rapid increase in the number of researchers working with unpublished sequence data from newly obtained transcriptomes, genomes and metagenomes. This has created many opportunities for novel biological research in fields including medicine (Qin *et al.*, 2010), developmental biology (Santos *et al.*, 2015), microbiology (Hugenholtz & Tyson, 2008), ecology and evolution (Ellegren, 2014; Seehausen *et al.*, 2014). Pursuing such opportunities often involves querying newly obtained sequence data with specific aims such as determining the presence or absence of a sequence in a specific sample, obtaining sequence to enable molecular cloning, quantitative PCR or phylogenetic analyses, inferring the names or functions of putative genes, and identifying sequence and structural variation within and between species.

At a basic level such analyses involve comparing one or more query sequences with one or more databases each containing between thousands and hundreds of millions of sequences. This is typically performed with BLAST (Basic Local Alignment Search Tool; Altschul *et al.*, 1990, 1997; Camacho *et al.*, 2009), likely the most commonly used bioinformatics tool (>100,000 citations; see also Van Noorden *et al.*, 2014; Korf *et al.*, 2003). For each query sequence, BLAST heuristically identifies database sequences that are similar and outputs relevant comparisons (Camacho *et al.*, 2009). To perform BLAST searches, biologists typically rely on graphical web interfaces provided by large organizations such as the National Center for Biotechnology Information (NCBI; Johnson *et al.*, 2008) or the European Bioinformatics Institute (Goujon *et al.*, 2010). These are valuable resources for published data, but despite efforts to encourage and facilitate early data deposition (Kaye *et al.*, 2009) it can take months or even years before new sequence data is available on such central repositories.

How can researchers perform BLAST queries on new or unpublished data? First, BLAST software can be freely downloaded. However, using it directly requires substantial command-line and UNIX skills (`http://www.ncbi.nlm.nih.gov/books/NBK52640/`) which many biologists lack (Smith, 2013). Furthermore, BLAST outputs are text-based; they lack potentially helpful graphical visualization. Second, queries and interpretation can be outsourced to bioinformatician collaborators or technicians with appropriate skills. Finally, the most appropriate solution for many laboratories would be to have access to a shared graphical user interface to run BLAST on private datasets. Several free and open source software packages make this possible, including ViroBLAST (Deng *et al.*, 2007), GMOD's BlastGraphic (O'Connor *et al.*, 2008), NCBI's wwwblast (Tao, 2006), NCBI's Amazon Cloud BLAST (NCBI, 2014), and Galaxy BLAST (Cock *et al.*, 2015). Unfortunately, setting up such interfaces is challenging: they are either deprecated and no longer maintained, or have complex installation dependencies such as custom configuration of the Apache web server (Fielding & Kaiser, 1997) or CGI paths (Gundavaram, 1996). Commercial software packages overcome some such problems, but their cost makes them inaccessible to many laboratories.

Once set up, a further challenge is using the graphical user interface. For example, users typically must choose between multiple BLAST algorithms (TBLASTX, TBLASTN, BLASTX, BLASTN, BLASTP) before entering query data or choosing search databases (e.g., Tao, 2006; Deng *et al.*, 2007; O'Connor *et al.*, 2008; NCBI, 2014; Cock *et al.*, 2015; Johnson *et al.*, 2008). This implicitly requires detailed knowledge of BLAST algorithms and available databases – yet even experienced BLAST users make frequent mistakes and must revisit their decisions. This design could be improved because only a single algorithm is appropriate for most combinations of input sequence and database. Many such inadequacies of BLAST user interfaces exist. This is likely because there is only a limited culture of user interface and user experience design for scientific software (Macaulay *et al.*, 2009) including for bioinformatics (Pavelin *et al.*, 2012).

Here we present Sequenceserver, a free software package designed to help overcome the largest of the aforementioned hurdles. By combining user-centric design (Garrett, 2011; Javahery *et al.*, 2004; Pavelin *et al.*, 2012) with modern software development and visualization approaches, Sequenceserver aims to increase the productivity of bioinformaticians setting up web servers for running BLAST on custom datasets, and for biologists performing and interpreting BLAST searches.

Results

Sequenceserver is a graphical user interface wrapper for BLAST designed with extensive consideration of user experience and user interface (Garrett, 2011; Javahery *et al.*, 2004; Pavelin *et al.*, 2012). Below we provide overviews of Sequenceserver's major features that facilitate setting up a BLAST server, performing queries and interpreting results.

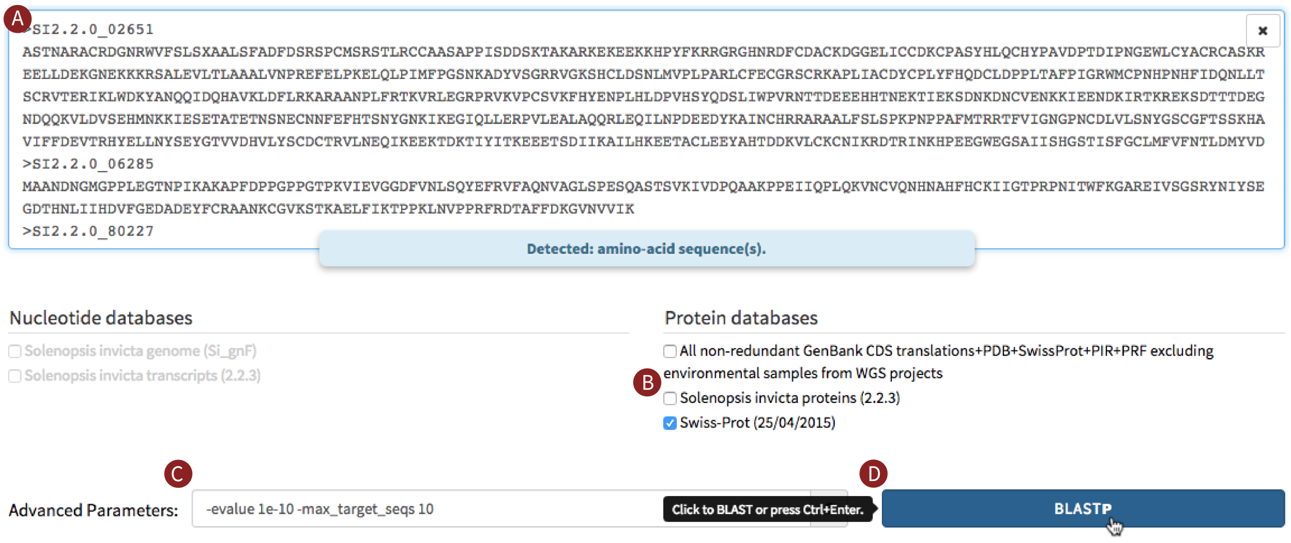
Assisted installation and configuration

Sequenceserver can be rapidly installed and configured by biologists with only little UNIX experience for individual use or for shared use as part of a research group. Entering `sudo gem install sequenceserver` into a terminal will download and install Sequenceserver on standard UNIX operating systems including Mac and Linux. Once installed, entering `sequenceserver` into a terminal downloads NCBI BLAST+ (Camacho *et al.*, 2009) if necessary and prompts the user to indicate the location of a directory containing database sequences in the standard FASTA format or in BLAST's specific BLASTDB database format. Sequenceserver then recursively scans the directory for FASTA files that have not yet been formatted as BLASTDBs. For each such FASTA file, Sequenceserver detects whether the FASTA file contains nucleotide or amino acid sequences and determines a suitable name to display the corresponding BLASTDB in the search form. Sequenceserver then confirms with the user if the FASTA file should be formatted as BLASTDB. Here, the user can modify the proposed name and optionally enter a relevant taxonomic identifier. The location of the directory containing database sequences is stored in a configuration file for subsequent use. Finally, Sequenceserver launches its built in web server (Harris & Haase, 2012) making the graphical user interface (Figure 1 and below) accessible from a web browser locally at `http://localhost:4567` or over the network at `http://user-ip:4567`; it automatically shows all available BLASTDBs, segregated according to nucleotide vs. amino acid sequence type.

Additional configuration possibilities include setting the number of threads to be used by BLAST, integrating Sequenceserver with web servers such as Nginx (Reese, 2008) or Apache (Fielding & Kaiser, 1997), adding password protection, customizing the interface colors and layout, and adding custom links from hits to external resources such as genome browsers (see online documentation).

Assistive BLAST query submission interface

Sequenceserver uses techniques including analysis of user input, interactive feedback and simple algorithms to streamline BLAST query submission and to prevent common errors that can cause BLAST to fail or generate misleading results. The user first types, pastes or drag-and-drops one or multiple FASTA format query sequences into a text-field (Figure 1). In the unlikely event that the user combined nucleotide sequences and amino acid sequences, an alert message is shown and the BLAST button will remain disabled to avoid BLAST generating meaningless results. Subsequently, the user selects one or several BLAST databases using checkboxes. Once a first database is selected, additional database selections are limited to those of the same type (i.e., either nucleotide or protein) to eliminate the risk of users combining incompatible databases that would cause BLAST to fail. Once a valid query is entered and a database is selected, the BLAST submission button activates. For most query-database combinations, the single possible basic BLAST algorithm will be used (Supplementary Figure S1). When multiple algorithms are appropriate (e.g., nucleotide query and nucleotide database: BLASTN and TBLASTX are both appropriate), a pull-down in the BLAST submission button allows the user to toggle between them. Sequenceserver's automatic algorithm selection reduces the risk of attempting to perform impossible BLAST queries. Finally, Sequenceserver includes an "advanced parameters" field providing access to all standard BLAST parameters available in the command-line (Camacho *et al.*, 2009).


A
B
C
D
Detected: amino-acid sequence(s).
Nucleotide databases
Solenopsis invicta genome (Si_gnF)
Solenopsis invicta transcripts (2.2.3)
Protein databases
All non-redundant GenBank CDS translations+PDB+SwissProt+PIR+PRF excluding environmental samples from WGS projects
Solenopsis invicta proteins (2.2.3)
Swiss-Prot (25/04/2015)
Click to BLAST or press Ctrl+Enter.
BLASTP

```
>SI2.2.0_02651
ASTNARACRDGNRWVFSLSXAALSFADFDSRSPCMSRSTLRCCAASAPPISDDSKTAKARKEKEEKKHPYFKRRGRGHNRDFCDACKDGGELICCDKCPASYHLQCHYPAVDPTDIPNGEWLCYACRCASKR
EELLDEKGNEKKKRSALEVLTLAAALVNPREFELPKELQLPIMFPGSNKADYVSGRRVGKSHCLDSNLMVPLPARLCFECGRSCRKAPLIACDYCPLYFHQDCLDPPLTAFPIGRWMCPNHPNHFIDQNLLT
SCRVTERIKLWDKYANQQIDQHAVKLDFLRKARAANPLFRTKVRLEGRPRVKVPCSVKFHYENPLHLDPVHSYQDSLIWPVRNTTDEEEHHTNEKTIEKSDNKDNCVENKKIEENDKIRTKREKSDTTTDEG
NDQQKVLDVSEHMNKKIESETATETNSNECNNFEFHTSNYGNKIKEGIQLLERPVLEALAQQRLEQILNPDEEDYKAINCHRRARAALFSLSPKPNPPAFMTRRTFVIGNGPNCDLVLSNYGSCGFTSSKHA
VIFFDEVTRHYELLNYSEYGTVVDHVLYSCDCTRVLNEQIKEEKTDKTIYITKEEETSDIIKAILHKEETACLEEYAHTDDKVLCKCNIKRDTRINKHPEEGWEGSAIISHGSTISFGCLMFVFNTLDMYVD
>SI2.2.0_06285
MAANDNGMGPPLEGTNPIKAKAPFDPPGPPGTPKVIEVGGDFVNLSQYEFRVFAQNVAGLSPESQASTSVKIVDPQAAKPPEIIQPLQKVNCVQNHNAHFHCKIIGTPRPNITWFKGAREIVSGSRYNIYSE
GDTHNLIIHDVFGEDADEYFCRAANKCGVKSTKAELFIKTPPKLNVPPRFRDTAFFDKGVNVVIK
>SI2.2.0_80227
```

Advanced Parameters: -evalue 1e-10 -max_target_seqs 10

Figure 1. Partial screenshot of the query interface. Dark red letters highlight the steps involved and some specific features. **A:** Three or more sequences were pasted into the query field (typewriter font; only the identifier is visible for the third sequence); a message confirms to the user that these are amino acid sequences. **B:** The Swiss-Prot protein database was the first database to be selected. As a result, additional database selections are limited to protein databases; nucleotide databases are disabled. **C:** The user entered (optional) advanced parameters to constrain the results to the 10 strongest hits with evalues stronger than 10^{-10}. **D:** The BLAST button is automatically activated and labeled "BLASTP" as this is the only possible basic BLAST algorithm for the given query-database combination. As the user's mouse pointer hovers over the BLASTP button, a tooltip indicates that a keyboard shortcut exists for this button.

Display of BLAST results

As a result of performing a BLAST query, an HTML report including graphical overviews is shown in the web browser (Figure 2; an interactive version of this figure is at `http://sequenceserver.com/paper/fig2interactive`). This report will feel familiar to users of NCBI BLAST but includes many additions and revisions for improved navigation, interpretation and follow-up analysis. For example, the report can be downloaded in HTML, XML and tab-delimited table formats; similarly a FASTA file containing all hit sequences can be downloaded. The HTML report clearly brings out the structure of results: If a user submitted multiple query sequences, a clickable index of queries is shown on one side. All queries, hits and BLAST HSPs (high-scoring segment pairs) are numbered to facilitate navigation. For each query, identified hits are summarized in a table and an overview graphic. This graphic indicates the strength and locations of alignment of each hit (hits with stronger e-values are darker).

Each hit includes multiple links for subsequent analysis. For example, each hit to a sequence with an NCBI or UniProt identifier includes a link to the relevant page; links to private genome browsers or other sites can be added (see online documentation). Additionally, each hit includes one link to download the full sequence in FASTA format and another link to display the sequence in the browser. This sequence viewing interface (Supplementary Figure S2) includes GenBank-style visualization for readability and displays appropriate coordinate information when the mouse pointer hovers over one or selects multiple residues (Gómez *et al.*, 2013). Furthermore, each hit includes a checkbox making it possible to simultaneously download a selection of multiple hit sequences as a single FASTA file.

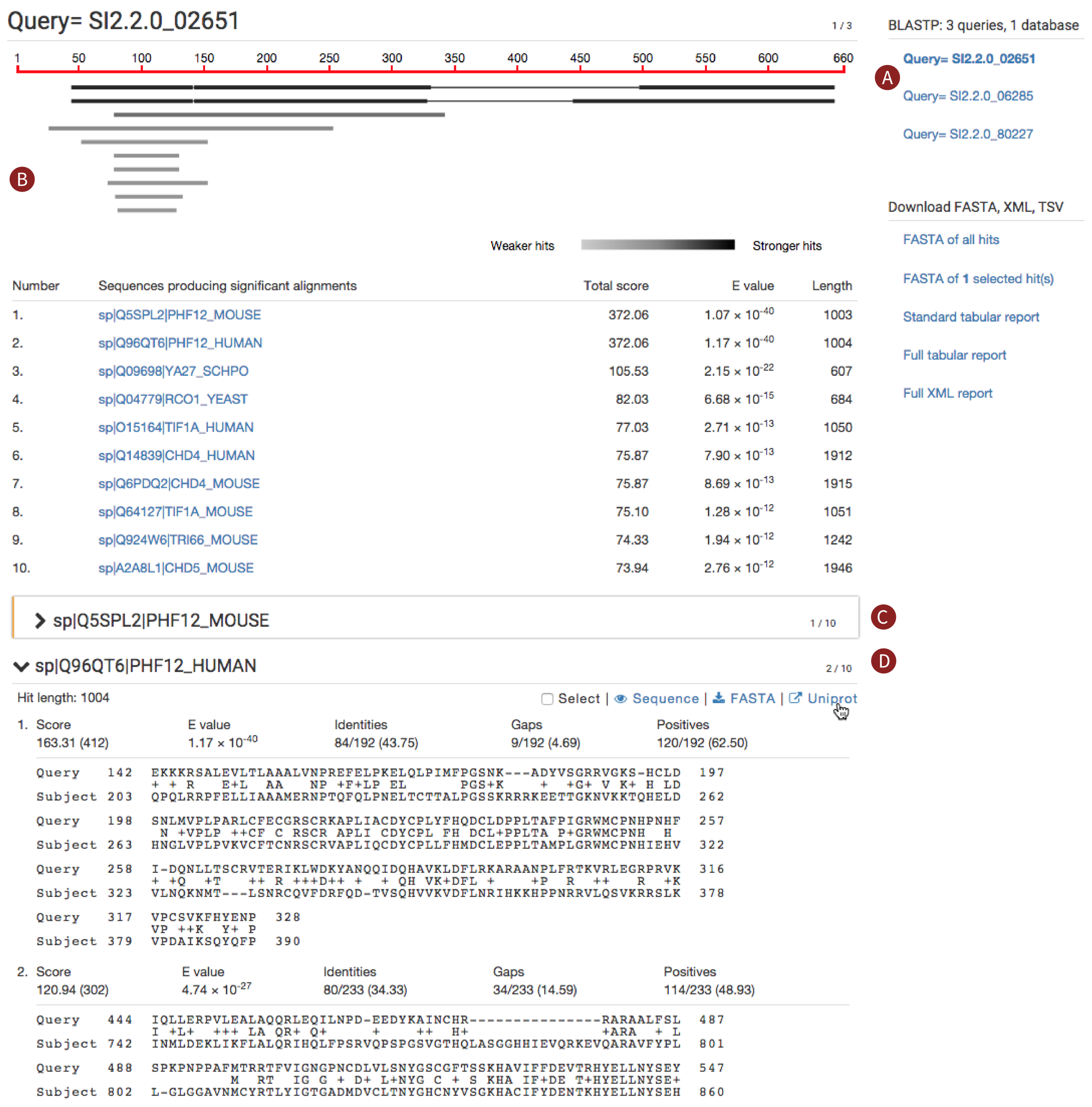
Discussion

We created Sequenceserver to overcome many of the challenges of performing BLAST on custom datasets. Below we review known applications of Sequenceserver, compare it to alternatives, consider its compatibility with tools for follow-up analyses and discuss future directions.

Applications and relevance

Sequenceserver has accelerated our own research (Gotzek *et al.*, 2011; Ingram *et al.*, 2012; Wurm *et al.*, 2011; Wang *et al.*, 2013; Kulmuni *et al.*, 2013; Privman *et al.*, 2013; Mondav *et al.*, 2014; Schrader *et al.*, 2014; Nygaard & Wurm, 2015) and that of others. Indeed, despite this being the first formal publication reporting Sequenceserver, the software has already been cited for research on emerging model organisms (Wurm, 2015) such as sea cucumber (Rowe *et al.*, 2014), starfish (Elphick *et al.*, 2013; Semmens *et al.*, 2013, 2015), falcons (Seim *et al.*, 2015), the sugar-apple tree (Gupta *et al.*, 2015), the Hessian fly *Mayetiola destructor* (Shreve *et al.*, 2013) and the house plant *Streptocarpus rexii* (Chiara *et al.*, 2013), as well as for research in bioadhesion (Rodrigues *et al.*, 2014), environmental microbiology (Castro *et al.*, 2015), and general research on BLAST (Sharma & Mantri, 2014).

Google Analytics referral links to `http://sequenceserver.com`, social web statistics, and community engagement through the mailing list indicates growing adoption of Sequenceserver (Table 1). For example, Sequenceserver is among the 10% most frequently downloaded packages on the biogems portal for bioinformatics software (Bonnal *et al.*, 2012, $> 28,000$ downloads), and is a main querying mechanism for community genomic databases including the SymGRASS database of sugarcane orthologous genes (Belarmino *et al.*, 2013), the *Drosophila suzukii* genome database (Chiu *et al.*, 2013), the ant genomes database (Wurm *et al.*, 2009), the planarian database (Brandl *et al.*, 2015), the butterfly genomics database (`http://blast.lepbase.org`), a portal for exploring low-complexity amino acid sequences (Kirmitzoglou & Promponas, 2015), the *Amborella* database (`http://amborella.huck.psu.edu/`), a *Fusarium* database (`http://www.fusariumnrpspks.dk/`), two ash genome projects (`https://geefu.oadb.tsl.ac.uk/` and `http://www.ashgenome.org/`) and an echinoderm transcriptome database (`http://echinodb.uncc.edu/blast/`). We are aware of at least 50 publicly accessible Sequenceserver instances, but precise usage statistics are difficult to confirm, as there is no centralized list of deployments. Our experience with deployment suggests that public servers are in the minority with most Sequenceserver instances being created for use by individual researchers or small groups as needed on the personal computers of



Number	Sequences producing significant alignments	Total score	E value	Length
1.	sp|Q5SPL2|PHF12_MOUSE	372.06	1.07×10^{-40}	1003
2.	sp|Q96QT6|PHF12_HUMAN	372.06	1.17×10^{-40}	1004
3.	sp|Q09698|YA27_SCHPO	105.53	2.15×10^{-22}	607
4.	sp|Q04779|RCO1_YEAST	82.03	6.68×10^{-15}	684
5.	sp|O15164|TIF1A_HUMAN	77.03	2.71×10^{-13}	1050
6.	sp|Q14839|CHD4_HUMAN	75.87	7.90×10^{-13}	1912
7.	sp|Q6PDQ2|CHD4_MOUSE	75.87	8.69×10^{-13}	1915
8.	sp|Q64127|TIF1A_MOUSE	75.10	1.28×10^{-12}	1051
9.	sp|Q924W6|TRI66_MOUSE	74.33	1.94×10^{-12}	1242
10.	sp|A2A8L1|CHD5_MOUSE	73.94	2.76×10^{-12}	1946

> sp|Q5SPL2|PHF12_MOUSE 1 / 10

sp|Q96QT6|PHF12_HUMAN 2 / 10

Hit length: 1004

	Score	E value	Identities	Gaps	Positives
1.	163.31 (412)	1.17×10^{-40}	84/192 (43.75)	9/192 (4.69)	120/192 (62.50)

```
Query   142  EKKKRSALEVLTLAAALVNPREFELPKELQLPIMFPGSNK---ADYVSGRRVGKS-HCLD  197
             + + R   E+L  AA   NP +F+LP EL      PGS+K    +  +G+ V K+ H LD
Subject 203  QPQLRRPFELLIAAAMERNPTQFQLPNELTCTTALPGSSKRRRKEETTGKNVKKTQHELD  262

Query   198  SNLMVPLPARLCFECGRSCRKAPLIACDYCPLYFHQDCLDPPLTAFPIGRWMCPNHPNHF  257
              N +VPLP ++CF C RSCR APLI CDYCPL FH DCL+PPLTA P+GRWMCPNH  H
Subject 263  HNGLVPLPVKVCFTCNRSCRVAPLIQCDYCPLLFHMDCLEPPLTAMPLGRWMCPNHIEHV  322

Query   258  I-DQNLLTSCRVTERIKLWDKYANQQIDQHAVKLDFLRKARAANPLFRTKVRLEGRPRVK  316
             + +Q  +T   ++ R +++D++ +  + QH VK+DFL +    +P  R  ++    R  +K
Subject 323  VLNQKNMT---LSNRCQVFDRFQD-TVSQHVVKVDFLNRIHKKHPPNRRVLQSVKRRSLK  378

Query   317  VPCSVKFHYENP  328
             VP ++K  Y+ P
Subject 379  VPDAIKSQYQFP  390
```

	Score	E value	Identities	Gaps	Positives
2.	120.94 (302)	4.74×10^{-27}	80/233 (34.33)	34/233 (14.59)	114/233 (48.93)

```
Query   444  IQLLERPVLEALAQQRLEQILNPD-EEDYKAINCHR---------------RARAALFSL  487
             I +L+  +++ LA QR+ Q+     +    ++  H+               +ARA  + L
Subject 742  INMLDEKLIKFLALQRIHQLFPSRVQPSPGSVGTHQLASGGHHIEVQRKEVQARAVFYPL  801

Query   488  SPKPNPPAFMTRRTFVIGNGPNCDLVLSNYGSCGFTSSKHAVIFFDEVTRHYELLNYSEY  547
                      M  RT  IG G + D+ L+NYG C + S KHA IF+DE T+HYELLNYSE+
Subject 802  L-GLGGAVNMCYRTLYIGTGADMDVCLTNYGHCNYVSGKHACIFYDENTKHYELLNYSEH  860
```

Figure 2. **Screenshot of a Sequenceserver BLAST report.** *An interactive version of this figure is at* `http://sequenceserver.com/paper/fig2interactive`. Three amino acid sequences were compared against the Swiss-Prot database using BLASTP with an evalue cutoff of 10^{-10} and keeping only the 10 strongest hits per query. This screenshot shows a portion of the results for the first query. Dark red letters highlight some of the specific features of this report. **A:** An index overview summarizes the query and database information and provides clickable links to query-specific results. **B:** Results for the first query are shown. These include a graphical overview indicating which parts of the query sequence aligns to each hit, a tabular summary of all hits, and alignment details for each hit. **C:** The first hit is selected for download; its alignment details have been folded away. **D:** The user is studying the second hit; the mouse pointer hovers over the link to the hit's UniProt page.

researchers. For instance, the authors collectively run 12 private servers and only 4 public servers at the time of this writing.

Sequenceserver is also used as an educational resource (e.g., `http://culture-bioinformatics.org/blast.shtml`) to provide entry-level bioinformatics students a controlled environment to run BLAST queries and explore the effects of changing BLAST parameters on the results. Indeed, SequenceServer allows the students to access their classroom's BLAST server from anywhere and the instructor to control the software and databases versions so that the results for any given query are consistent and predictable, as is critical for a classroom setting and not feasible using publicly-available servers such as NCBI.

Metric	**Statistic**
Hits to sequenceserver.com in the last 12 months	$11,050$
Mailing list users	102
Mailing list threads	137
RubyGems downloads	$29,500$
GitHub stars	80
Twitter mentions	102
Sites with publicly available instances	52

Table 1. Usage and impact statistics of Sequenceserver at the time of writing.

Alternatives and compatibility with tools for follow-up analysis

Multiple alternatives to Sequenceserver similarly provide graphical user interfaces (GUIs) for BLAST (Tao, 2006; Deng *et al.*, 2007; O'Connor *et al.*, 2008; Cock *et al.*, 2015; NCBI, 2014). However, they have three broad levels of shortcomings (summarized in Supplementary Table S1). First, most of these tools first require other software to be installed and include little or no automation of common tasks such as downloading and configuring BLAST+ software or adding custom databases. Installation of such tools is out of reach of most biologists. In contrast, biologists with only limited UNIX experience are able to set up Sequenceserver. Importantly, this ease of setup does not limit possible usage options. Indeed, experienced power users retain the flexibility of deploying Sequenceserver on personal computers, compute clusters and the cloud, customizing all aspects of Sequenceserver, and migrating from its built-in webserver to industry-strength standards.

Second, the query submission interface of alternatives to Sequenceserver are designed around how BLAST is implemented rather than accompanying the user through their thought and work process. For example, in some cases (Cock *et al.*, 2015; Johnson *et al.*, 2008) users must select a BLAST algorithm even before they have entered the query sequence and selected the databases to search. This is counterintuitive as a user's thought process generally begins with their query sequence. In other cases (Tao, 2006; Deng *et al.*, 2007; O'Connor *et al.*, 2008; NCBI, 2014), users must select a BLAST algorithm after they have entered the query or after they have selected the databases to search. This is more in-line with users workflow, but both approaches fail to consider that only a single basic BLAST algorithm is applicable for most query-database combinations, thus the users wastes time choosing the appropriate algorithm. Furthermore, none of the alternative GUIs consider that the user may erroneously select an algorithm that is inappropriate for their query-database combination, or consider that the user may erroneously combine nucleotide and amino acid sequence in a single query. In both scenarios, BLAST will either fail or incorrectly interpret the nucleotides adenine (A), thymine (T), cytosine (C) and guanine (G) as the amino acids arginine, threonine, cysteine and glutamine.

Finally, most GUIs (Tao, 2006; Deng *et al.*, 2007; O'Connor *et al.*, 2008; Cock *et al.*, 2015; NCBI, 2014) either present the default HTML output of command-line BLAST or slightly reformat it. They thus lack many of the features of Sequenceserver that facilitate navigation, interpretation and follow-up analyses including exporting analysis results in different formats. Sequenceserver uses an approach that allows exporting a single analysis in many different formats. As part of this, Sequenceserver's HTML report is generated from scratch, making it much easier to extend it with additional features.

Several commercial providers offer alternatives to Sequenceserver (e.g., BlastStation, Geneious, CLC). While such tools can overcome some of shortcomings listed here, they have other drawbacks. These include costs, risks in terms of reproducibility and long-term sustainability, limited flexibility for sharing with distant colleagues, and limited customization possibilities.

BLAST reports from Sequenceserver (e.g., in XML and tab-delimited formats) can be used as input for analysis tools that work directly with BLAST output. Thus users can access an instance of Sequenceserver running on a high-performance computer to create a BLAST report which they subsequently use with tools that facilitate specific analyses [e.g., MEGAN (Huson *et al.*, 2007), Blast2GO (Conesa *et al.*, 2005), BLAST2GENE (Suyama *et al.*, 2004), EPoS (Griebel *et al.*, 2008)], provide different visualization facilities [e.g., Kablammo (Wintersinger & Wasmuth, 2015), BLASTGrabber (Neumann *et al.*, 2014), Circoletto (Darzentas, 2010), JAMBLAST (Lagnel *et al.*, 2009), BOV (Gollapudi *et al.*, 2008)] or facilitate importing BLAST results into a spreadsheet application or an SQL database [e.g., MuSeqBox (Xing & Brendel, 2001), Zerg (Paquola *et al.*, 2003), Batch BLAST extractor (Pirooznia *et al.*, 2008), BioParser (Catanho *et al.*, 2006)]. A comparison of tools to interpret and work with BLAST output is provided by Neumann *et al.* (2013).

Future directions

The community of Sequenceserver developers will continue adding novel functionalities to our software in a public and open manner. We expect improvements along three main lines. First, additional output visualization facilities will be immensely beneficial for making sense of large datasets (Nielsen *et al.*, 2010). In particular, this will involve integrating visualization tools developed by others (Gollapudi *et al.*, 2008; Neumann *et al.*, 2013; Gómez *et al.*, 2013; Wintersinger & Wasmuth, 2015; `http://wurmlab.github.io/tools/genevalidator/`), while HMMER provides high-sensitivity for comparisons against sequence profiles (Eddy, 2009). Similarly, DIAMOND is up to 20,000 times faster than BLASTX or BLASTP; this can be useful when datasets are particularly large (Buchfink *et al.*, 2015, e.g., metagenomics). New versions of Sequenceserver and their features will be announced on Sequenceserver's mailing list and Github repository.

Conclusion

Recent improvements in DNA sequencing technologies are enabling ever smaller groups of biologists to create previously unimaginable datasets on diverse ranges of species (Koboldt *et al.*, 2015; Ellegren, 2014). Realizing that a major challenge for many biologist researchers is to effectively query such datasets, we built the Sequenceserver graphical frontend for BLAST using modern user interface and user experience paradigms. Our software is particularly enabling for researchers and communities focusing on species for which genomic data is novel as it empowers them to independently and relatively easily perform BLAST queries and share datasets. At the base of this software are modern software development technologies and approaches that have over the last few years become heavily used by the small groups of software developers at the core of many internet startup companies (Ries, 2011; Thiel & Masters, 2014). We are excited that the computational tools underlying basic scientific research can benefit from such technical possibilities (Altschul *et al.*, 2013) and appropriate consideration of user experience (Pavelin *et al.*, 2012). The increased usability, maintainability and sustainability of well-built tools ultimately translate into increased agility and productivity of researchers and in turn the impacts of their breakthroughs.

Methods

Programming environment

We developed Sequenceserver from scratch rather than basing our work on the NCBI's initial Perl/CGI www-blast wrapper (Tao, 2006) to reduce technical debt (Lehman, 1980). The core of Sequenceserver is written in the Ruby programming language (Flanagan & Matsumoto, 2008) popular for creating dynamic websites (Ruby & Thomas, 2011) and bioinformatics tools (Goto *et al.*, 2010), while JavaScript and HTML/CSS are used for layout and interactions in the web browser. We use multiple preexisting tools and libraries to facilitate development: The lightweight framework Sinatra (Harris & Haase, 2012) is used to create URL endpoints to load search form and run a BLAST search from the browser. BLAST searches are delegated to the compiled command line version of BLAST (Camacho *et al.*, 2009); we use Ox (`https://github.com/ohler55/ox`) to parse BLAST XML and create the HTML report. Underscore (`http://underscorejs.org/`), HTML5 Shiv (`https://github.com/afarkas/html5shiv`), jQuery (`http://jquery.com`), jQuery UI (`http://jqueryui.com`), Webshim (`https://afarkas.github.io/webshim/demos/`), and Twitter Bootstrap (`http://getbootstrap.com`) libraries are used to create a uniform scripting environment (for dynamic aspects of the user interface) and a consistent look-and-feel (for visual layout) across most browsers. The d3 (`http://d3js.org/`) and BioJS (Gómez *et al.*, 2013) libraries are used respectively for generating the graphical overview and the sequence viewing interface.

Sustainable software development approach

Throughout the development of Sequenceserver we followed six modern software engineering practices designed to facilitate and accelerate development while improving the long-term sustainability of the software (Prlić & Procter, 2012; Wilson *et al.*, 2014; `http://www.software.ac.uk/resources/guides-everything`). First, we used an open source and agile development approach (Shore & Warden, 2007) involving frequent incremental improvements, peer review and frequent deployment on our own servers and within the community. Second, we structured the software according to object-oriented paradigm (Weisfeld, 2008), thus leading to a clean separation of different parts of the code. Third, we followed two important software development principles: "don't repeat yourself" (DRY) leads to fewer lines of code and thus fewer bugs, and makes it easier to read and understand code than if similar commands are repeated in several places (Hunt & Thomas, 2000); "keep it simple, stupid" (KISS) reduces unnecessary complexity and thus lowers risks and leads to higher maintainability (Raymond, 2003). Fourth, we reuse widely established software packages and libraries (see programming environment) to avoid reimplementation. This accelerates our work and reduces the amount of Sequenceserver-specific code, which in turn further reduces the likelihood of adding bugs (Sametinger, 1997). Fifth, we developed unit tests (Ammann & Offutt, 2008) and integration tests (Ammann & Offutt, 2008) for many parts of Sequenceserver's code, and used continuous integration (`https://travis-ci.org/`) to ensure these tests are automatically run whenever any change is made to the code, thus increasing the chances of rapidly detecting errors. Sixth, we use the rubocop code analyser (`https://github.com/bbatsov/rubocop`) to ensure that our code respects the Ruby community style guide in terms of names of variables and methods, code structure and code formatting. Such respect of style standards makes code more accessible to other programmers and scientists than if code is inconsistently styled or if we had chosen our own conventions (Martin, 2008; Wurm, 2015). Finally, we use the Codeclimate service (`http://codeclimate.com`) for automatic detection of potential problems with software design.

Graphical user interface design principles

To ensure a fluid user experience that increases researcher productivity, we designed Sequenceserver around eight modern user interface design principles. First, the interface contains only essential information so as to minimize distractions for the user. Second, the information is laid out in a clear and hierarchically structured manner. As part of this, we paid special attention to typography, using Roboto (`https://www.google.com/fonts/specimen/Roboto`) for headings and Open Sans (`https://www.google.com/fonts/specimen/Open+Sans`) for normal text. These free, contemporary typefaces were designed to maximize legibility and overall aesthetics across electronic devices and print media. Third, we used automation where possible to minimize the number of decisions required from the user. For example, based on query type and databases selection we limit the choices for algorithm selection (except in the case of nucleotide-nucleotide search only a single BLAST algorithm is possible; see Supplementary Figure S1). Fourth, we use interactive visual feedback and cues for step-by-step discovery of the workflow. For example, the BLAST button remains disabled until the user has provided query sequence(s) and selected target databases. If the user tries to click the BLAST button while it is disabled a tooltip indicates that a required input is missing. Similarly, selection of protein databases is automatically disabled if the user has already selected a nucleotide database (and vice versa). Fifth, we remain consistent and contextual with regards to user interaction. For example, notification of sequence type does not depend on how query sequence was provided. This notification is shown below the query sequence input field – where the user's eyes are likely to go after query input – instead of using a global designated notification area or displaying pop-up windows that can be disruptive or are ignored. Similarly, a "clear query" button is shown only after the user has provided query sequence(s) and is positioned where a user is likely to look for it. Sixth, we try not to let the advantages of a graphical interface and efforts to create an easily accessible user experience limit the scope of what the user can do. For example, all possible advanced BLAST search options can be entered via a generic input field. Similarly, tooltips over report download links are only shown after the mouse pointer has hovered for at least 500ms. This delay means most users won't be bothered by tooltips after they have used the interface a few times. Seventh, we exploit intuitive human notions of colors. For example, if the user erroneously tries to combine nucleotide and amino acid sequences in the query, the query input-area is gently highlighted using a red border to indicate an error. At a different level, in the graphical overview shown for each query, the color of each hit indicates its strength, with stronger e-values being darker. Finally, the wording of error messages is similar to informal human conversation to create empathy and familiarity, which may also clarify that Sequenceserver is built by a community of scientists. For example, if a user encounters a bug in Sequenceserver, the error popup reads "Oops! Something went wonky. Below is the error detail. Please could you report it to our Google Group."

Community building

We took four measures to help create a community and allow easy customization of our software. First, we performed all development in an open manner on the GitHub source code sharing/development site (`http://github.com`). This ensures that bioinformatician users can easily access the source code and customize the software. Indeed, such customization can be easily merged back into the main codebase of the software; several GitHub users are included as coauthors of this manuscript because of such contributions. Second, we provide extensive documentation as part of the source code and on the website `http://sequenceserver.com`. Third, we created a dialog with users (via GitHub issue tracking as well as a specific Google Groups mailing list); here users ask and answer each other's questions and help each other regarding issues arising with customization, installation and specific datasets. Finally, we isolated the most commonly used customization code into separate files (a configuration file and a specific file for adding links from BLAST search results to custom databases such as custom genome browsers or other databases). Isolating the parts of code that a bioinformatician is most likely to edit facilitates customization and reduces risks when upgrading the software.

Acknowledgments

The authors would like to thank the large community of Sequenceserver users and contributors for their feedback and input throughout this project.

Funding YW was funded by an ERC grant to Laurent Keller during the creation of Sequenceserver. YW and AP were supported during the writing of this manuscript by BBSRC grant BB/K004204/1, NERC grant NE/L00626X/1 and NERC EOS Cloud. YW is a fellow of the Software Sustainability Institute (http://software.ac.uk). BJW was supported by a University of Melbourne MRS, as well as Genomic Science Program of the United States Department of Energy Office of Biological and Environmental Research, grant DE-SC0004632.

Supplementary Information

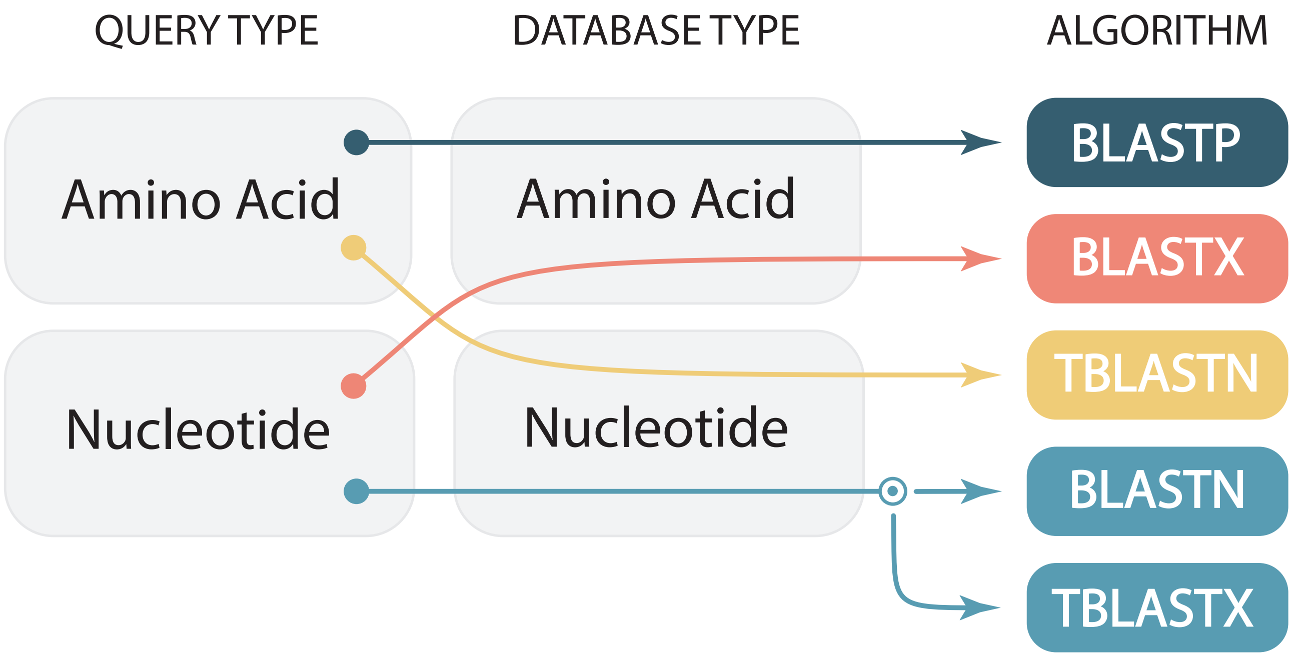
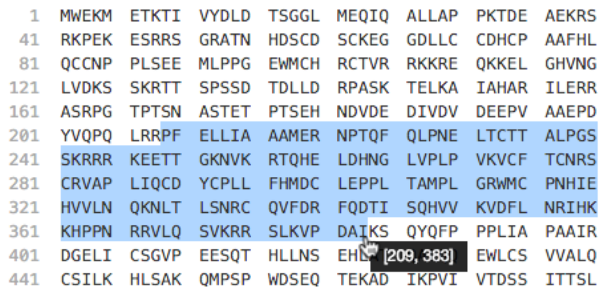


Figure S1. Automatic BLAST algorithm selection. BLAST includes five basic algorithms (right column). Arrows indicate how Sequenceserver automatically selects an appropriate BLAST algorithm based on the sequence types of query (left column) and selected databases (middle column). For the first three combinations of query and database types, only one algorithm is possible. The circle indicates that for nucleotide query and nucleotide database, the user can choose between BLASTN and TBLASTX.



```
  1  MWEKM  ETKTI  VYDLD  TSGGL  MEQIQ  ALLAP  PKTDE  AEKRS
 41  RKPEK  ESRRS  GRATN  HDSCD  SCKEG  GDLLC  CDHCP  AAFHL
 81  QCCNP  PLSEE  MLPPG  EWMCH  RCTVR  RKKRE  QKKEL  GHVNG
121  LVDKS  SKRTT  SPSSD  TDLLD  RPASK  TELKA  IAHAR  ILERR
161  ASRPG  TPTSN  ASTET  PTSEH  NDVDE  DIVDV  DEEPV  AAEPD
201  YVQPQ  LRRPF  ELLIA  AAMER  NPTQF  QLPNE  LTCTT  ALPGS
241  SKRRR  KEETT  GKNVK  RTQHE  LDHNG  LVPLP  VKVCF  TCNRS
281  CRVAP  LIQCD  YCPLL  FHMDC  LEPPL  TAMPL  GRWMC  PNHIE
321  HVVLN  QKNLT  LSNRC  QVFDR  FQDTI  SQHVV  KVDFL  NRIHK
361  KHPPN  RRVLQ  SVKRR  SLKVP  DAIKS  QYQFP  PPLIA  PAAIR
401  DGELI  CSGVP  EESQT  HLLNS  EHL [209, 383] EWLCS  VVALQ
441  CSILK  HLSAK  QMPSP  WDSEQ  TEKAD  IKPVI  VTDSS  ITTSL
```

Figure S2. Screenshot of viewing interface for a protein hit sequence. Residues are grouped in multiples of 5, and the coordinates of the first residue of each line are shown in gray. Here, the user selected a range of amino acids; their specific coordinates are shown in a tooltip. The viewing interface for nucleotides is similar.

		ViroBlast	NCBI's wwwblast	GMOD's BlastGraphic	Galaxy Blast	NCBI Amazon Cloud BLAST	Sequenceserver
Reference and/or URL		Deng et al 2007	Tao T (2006) wwwblast: Setup and usage. http://www.ncbi.nlm.nih.gov/staff/tao/URLAPI/wwwblast/.	Oconnor et al 2008	Cock et al 2015	NCBI (2014) BLAST searches at the cloud. http://blast.ncbi.nlm.nih.gov/ blast/docs/BLAST_at_AWS.pdf.	This publication
Installation and configuration	**Specific software requirements**	Apache web server, PHP	Apache web server, CGI	Apache web server, Perl and several Perl modules from CPAN	Galaxy server	Only runs on Amazon Web Services	Ruby programming language[1]
	BLAST software download	Manual	Manual	Manual	Automatic[2]	Included	Automatic
	BLAST configuration	Manual	Manual	Manual	Automatic	Included	Automatic
	Detection and formatting of new datasets	Manual	Manual	Manual	Manual	Manual	Automatic/ assisted
	Adding new datasets to interface	Manual configuration	Manual configuration	Manual configuration	Manual configuration	Manual configuration	Automatic
	Support for NCBI's current BLAST+ software (Camacho et al 2009).	Yes	No	Yes	Yes	Yes	Yes
	Updating software while preserving configuration	Involved	Involved	Involved	Involved	Involved	Trivial
	Can be integrated with standard web servers for community website	Yes	Yes	Yes	Yes	Yes	Yes
BLAST query submission	**Automatic BLAST algorithm selection**	No	No	No	No	No	Yes
	Prevent user from mixing protein and nucleotide query sequences	No	No	No	No	No	Yes
	Prevent user from running BLAST with impossible query-database-algorithm combinations	No	No	No	No	No	Yes
	Ability to select multiple databases for search	Yes	No	Yes	Yes	No	Yes
	Ability to enter advanced options	Yes	Yes	Limited	Yes	Yes[3]	Yes
Output	**Graphical overview of hits**	No	Yes	Yes	No	No	Yes
	Results clearly structured as in the NCBI's public BLAST interface.	No	No	No	No	No	Yes
	Ability to filter results	Yes: by e-value and identity.	No	No	No	No	No
	Link hits to custom external resources	No	No	Yes	No	No	Yes
	Ability to download hit sequences	Yes	No	No	No	No	Yes
	Ability to download BLAST reports	No	Yes[4]: XML	No	No	Yes[4,5]: XML, Tab-delimited, JSON & text	Yes: XML & Tab-delimited
Development status		Unclear	Inactive	Inactive	Active, open community	Active, developed only by NCBI	Active, open community
Notes		[1] Pre-installed on Mac. One-line installation through package managers on Linux. [2] Through '`ncbi_blast_plus`' package from galaxy tool shed [3] Default: only e-value; configuration enables more options. [4] The format in which results are desired can be pre-selected from the search form. [5] Using the command line `blast_formatter` tool.					

Table S1. Alternatives to Sequenceserver. Presence or absence of features in BLAST GUIs is highlighted across four broad categories: Installation and configuration, BLAST query submission, Output, and Development status. Preferred features are indicated in green.

References

Qin J, Li R, Raes J, Arumugam M, Burgdorf KS, Manichanh C, Nielsen T, Pons N, Levenez F, Yamada T, *et al.*. 2010. A human gut microbial gene catalogue established by metagenomic sequencing. *Nature*, **464**: 59–65.

Santos ME, Berger CS, Refki PN, Khila A. 2015. Integrating evo-devo with ecology for a better understanding of phenotypic evolution. *Briefings in Functional Genomics*, p. elv003.

Hugenholtz P, Tyson GW. 2008. Microbiology: Metagenomics. *Nature*, **455**: 481–483.

Ellegren H. 2014. Genome sequencing and population genomics in non-model organisms. *Trends in Ecology & Evolution*, **29**: 51–63.

Seehausen O, Butlin RK, Keller I, Wagner CE, Boughman JW, Hohenlohe PA, Peichel CL, Saetre GP, Bank C, Brännström A, *et al.*. 2014. Genomics and the origin of species. *Nature Reviews Genetics*, **15**: 176–192.

Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. 1990. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, **215**: 403–410.

Altschul S, Madden T, Schaffer A, Zhang J, Zhang Z, Miller W, Lipman D. 1997. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, **25**: 3389–3402.

Camacho C, Coulouris G, Avagyan V, Ma N, Papadopoulos J, Bealer K, Madden TL. 2009. BLAST+: Architecture and applications. *BMC Bioinformatics*, **10**: 421.

Van Noorden R, Maher B, Nuzzo R. 2014. The top 100 papers. *Nature*, **514**: 550–553.

Korf I, Yandell M, Bedell J. 2003. *BLAST*. O'Reilly, Sebastopol, CA.

Johnson M, Zaretskaya I, Raytselis Y, Merezhuk Y, McGinnis S, Madden TL. 2008. NCBI BLAST: A better web interface. *Nucleic Acids Research*, **36**: W5–W9.

Goujon M, McWilliam H, Li W, Valentin F, Squizzato S, Paern J, Lopez R. 2010. A new bioinformatics analysis tools framework at EMBL-EBI. *Nucleic Acids Research*, **38**: W695–W699.

Kaye J, Heeney C, Hawkins N, de Vries J, Boddington P. 2009. Data sharing in genomics – re-shaping scientific practice. *Nature Reviews Genetics*, **10**: 331–335.

Smith DR. 2013. The battle for user-friendly bioinformatics. *Frontiers in Genetics*, **4**: 187.

Deng W, Nickle DC, Learn GH, Maust B, Mullins JI. 2007. ViroBLAST: A stand-alone BLAST web server for flexible queries of multiple databases and user's datasets. *Bioinformatics*, **23**: 2334–2336.

O'Connor BD, Day A, Cain S, Arnaiz O, Sperling L, Stein LD. 2008. GMODWeb: A web framework for the Generic Model Organism Database. *Genome Biology*, **9**: R102.

Tao T. 2006. wwwblast: Setup and usage. `http://www.ncbi.nlm.nih.gov/staff/tao/URLAPI/wwwblast/`.

NCBI. 2014. BLAST searches at the cloud. `http://blast.ncbi.nlm.nih.gov/blast/docs/BLAST_at_AWS.pdf`.

Cock PJA, Chilton JM, Grüning B, Johnson JE, Soranzo N. 2015. NCBI BLAST+ integrated into Galaxy. bioRxiv doi: 10.1101/014043.

Fielding RT, Kaiser G. 1997. The Apache HTTP server project. *Internet Computing, IEEE*, **1**: 88–90.

Gundavaram S. 1996. *CGI programming on the World Wide Web*. O'Reilly, Sebastopol, CA.

Macaulay C, Sloan D, Jiang X, Forbes P, Loynton S, Swedlow J, Gregor P. 2009. Usability and user-centered design in scientific software development. *Software, IEEE*, **26**: 96–102.

Pavelin K, Cham JA, de Matos P, Brooksbank C, Cameron G, Steinbeck C. 2012. Bioinformatics meets user-centred design: A perspective. *PLoS Computational Biology*, **8**: e1002554.

Garrett JJ. 2011. *The elements of user experience: User-centered design for the Web and beyond, 2nd ed.* New Riders, Berkeley, CA.

Javahery H, Seffah A, Radhakrishnan T. 2004. Beyond power: Making bioinformatics tools user-centered. *Communcations of the ACM*, **47**: 58–63.

Harris A, Haase K. 2012. *Sinatra: Up and running*. Oreilly and Associates Inc, Sebastopol, CA.

Reese W. 2008. Nginx: The high-performance web server and reverse proxy. *Linux Journal*, **2008**: 2.

Gómez J, García LJ, Salazar GA, Villaveces J, Gore S, García A, Martín MJ, Launay G, Alcántara R, del Toro N, *et al.*. 2013. BioJS: An open source JavaScript framework for biological data visualization. *Bioinformatics*, **29**: 1103–1104.

Gotzek D, Robertson HM, Wurm Y, Shoemaker D. 2011. Odorant binding proteins of the red imported fire ant, *Solenopsis invicta*: An example of the problems facing the analysis of widely divergent proteins. *PLoS ONE*, **6**: e16289.

Ingram KK, Kutowoi A, Wurm Y, Shoemaker D, Meier R, Bloch G. 2012. The molecular clockwork of the fire ant *Solenopsis invicta*. *PLoS ONE*, **7**: e45715.

Wurm Y, Wang J, Riba-Grognuz O, Corona M, Nygaard S, Hunt BG, Ingram KK, Falquet L, Nipitwattanaphon M, Gotzek D, *et al.*. 2011. The genome of the fire ant *Solenopsis invicta*. *Proceedings of the National Academy of Sciences of the United States of America*, **108**: 5679–5684.

Wang J, Wurm Y, Nipitwattanaphon M, Riba-Grognuz O, Huang YC, Shoemaker D, Keller L. 2013. A Y-like social chromosome causes alternative colony organization in fire ants. *Nature*, **493**: 664–668.

Kulmuni J, Wurm Y, Pamilo P. 2013. Comparative genomics of chemosensory protein genes reveals rapid evolution and positive selection in ant-specific duplicates. *Heredity*, **110**: 538–547.

Privman E, Wurm Y, Keller L. 2013. Duplication and concerted evolution in a master sex determiner under balancing selection. *Proceedings of The Royal Society of London B: Biological Sciences*, **280**: 20122968.

Mondav R, Woodcroft BJ, Kim EH, McCalley CK, Hodgkins SB, Crill PM, Chanton J, Hurst GB, Verberkmoes NC, Saleska SR, *et al.*. 2014. Discovery of a novel methanogen prevalent in thawing permafrost. *Nature Communications*, **5**: 3212.

Schrader L, Kim JW, Ence D, Zimin A, Klein A, Wyschetzki K, Weichselgartner T, Kemena C, Stökl J, Schultner E, *et al.*. 2014. Transposable element islands facilitate adaptation to novel environments in an invasive species. *Nature Communications*, **5**: doi: 10.1038/ncomms6495.

Nygaard S, Wurm Y. 2015. Ant genomics (Hymenoptera: Formicidae): Challenges to overcome and opportunities to seize. *Myrmecological News*, **21**: 59–72.

Wurm Y. 2015. Arthropod genomics beyond fruit flies: bridging the gap between proximate and ultimate causation. *Briefings in Functional Genomics*, **14**: 381–383.

Rowe ML, Achhala S, Elphick MR. 2014. Neuropeptides and polypeptide hormones in echinoderms: New insights from analysis of the transcriptome of the sea cucumber *Apostichopus japonicus*. *General and Comparative Endocrinology*, **197**: 43–55.

Elphick MR, Achhala S, Martynyuk N. 2013. The evolution and diversity of SALMFamide neuropeptides. *PLoS ONE*, **8**: e59076.

Semmens DC, Dane RE, Pancholi MR, Slade SE, Scrivens JH, Elphick MR. 2013. Discovery of a novel neurophysin-associated neuropeptide that triggers cardiac stomach contraction and retraction in starfish. *The Journal of Experimental Biology*, **216**: 4047–4053.

Semmens DC, Beets I, Rowe ML, Blowes LM, Oliveri P, Elphick MR. 2015. Discovery of sea urchin ngfffamide receptor unites a bilaterian neuropeptide family. *Open Biology*, **5**: 150030.

Seim I, Jeffery PL, Herington AC, Chopin LK. 2015. Comparative analysis reveals loss of the appetite-regulating peptide hormone ghrelin in falcons. *General and Comparative Endocrinology*, **216**: 98–102.

Gupta Y, Pathak AK, Singh K, Mantri SS, Singh SP, Tuli R. 2015. De novo assembly and characterization of transcriptomes of early-stage fruit from two genotypes of *Annona squamosa* L. with contrast in seed number. *BMC genomics*, **16**: 86.

Shreve JT, Shukle RH, Subramanyam S, Johnson AJ, Schemerhorn BJ, Williams CE, Stuart JJ. 2013. A genome-wide survey of small interfering RNA and microRNA pathway genes in a galling insect. *Journal of Insect Physiology*, **59**: 367–376.

Chiara M, Horner DS, Spada A. 2013. *De novo* assembly of the transcriptome of the non-model plant *Streptocarpus rexii* employing a novel heuristic to recover locus-specific transcript clusters. *PLoS ONE*, **8**: e80961.

Rodrigues M, Lengerer B, Ostermann T, Ladurner P. 2014. Molecular biology approaches in bioadhesion research. *Beilstein Journal of Nanotechnology*, **5**: 983–993.

Castro JF, Razmilic V, Gomez-Escribano JP, Andrews B, Asenjo JA, Bibb MJ. 2015. Identification and heterologous expression of the chaxamycin biosynthetic gene cluster from *Streptomyces leeuwenhoekii*. *Applied and Environmental Microbiology*, **81**: 5820–5831.

Sharma P, Mantri SS. 2014. WImpiBLAST: Web interface for mpiBLAST to help biologists perform large-scale annotation using high performance computing. *PLoS ONE*, **9**: e101144.

Bonnal RJ, Aerts J, Githinji G, Goto N, MacLean D, Miller CA, Mishima H, Pagani M, Ramirez-Gonzalez R, Smant G, *et al.*. 2012. Biogem: an effective tool-based approach for scaling up open source software development in bioinformatics. *Bioinformatics*, **28**: 1035–1037.

Belarmino LC, Silva RLdO, Cavalcanti NdMS, Krezdorn N, Kido EA, Horres R, Winter P, Kahl G, Benko-Iseppon AM. 2013. SymGRASS: A database of sugarcane orthologous genes involved in arbuscular mycorrhiza and root nodule symbiosis. *BMC Bioinformatics*, **14 Suppl 1**: S2.

Chiu JC, Jiang X, Zhao L, Hamm CA, Cridland JM, Saelao P, Hamby KA, Lee EK, Kwok RS, Zhang G, *et al.*. 2013. Genome of *Drosophila suzukii*, the spotted wing *Drosophila*. *G3*, **3**: 2257–2271.

Wurm Y, Uva P, Ricci F, Wang J, Jemielity S, Iseli C, Falquet L, Keller L. 2009. Fourmidable: A database for ant genomics. *BMC Genomics*, **10**: 5.

Brandl H, Moon H, Vila-Farré M, Liu SY, Henry I, Rink JC. 2015. Planmine – a mineable resource of planarian biology and biodiversity. *Nucleic Acids Research*, **doi**: 10.1093/nar/gkv1148.

Kirmitzoglou I, Promponas VJ. 2015. LCR-eXXXplorer: A web platform to search, visualize and share data for low complexity regions in protein sequences. *Bioinformatics*, **31**: 2208–2210.

Huson DH, Auch AF, Qi J, Schuster SC. 2007. MEGAN analysis of metagenomic data. *Genome Research*, **17**: 377–386.

Conesa A, Götz S, García-Gómez JM, Terol J, Talón M, Robles M. 2005. Blast2GO: A universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics*, **21**: 3674–3676.

Suyama M, Torrents D, Bork P. 2004. BLAST2GENE: A comprehensive conversion of blast output into independent genes and gene fragments. *Bioinformatics*, **20**: 1968–1970.

Griebel T, Brinkmeyer M, Böcker S. 2008. EPoS: a modular software framework for phylogenetic analysis. *Bioinformatics*, **24**: 2399–2400.

Wintersinger JA, Wasmuth JD. 2015. Kablammo: An interactive, web-based blast results visualizer. *Bioinformatics*, **31**: 1305–1306.

Neumann RS, Kumar S, Haverkamp THA, Shalchian-Tabrizi K. 2014. BLASTGrabber: A bioinformatic tool for visualization, analysis and sequence selection of massive BLAST data. *BMC Bioinformatics*, **15**: 128.

Darzentas N. 2010. Circoletto: Visualizing sequence similarity with Circos. *Bioinformatics*, **26**: 2620–2621.

Lagnel J, Tsigenopoulos CS, Iliopoulos I. 2009. NOBLAST and JAMBLAST: New options for BLAST and a Java application manager for BLAST results. *Bioinformatics*, **25**: 824–826.

Gollapudi R, Revanna KV, Hemmerich C, Schaack S, Dong Q. 2008. BOV – a web-based BLAST output visualization tool. *BMC Genomics*, **9**: 414.

Xing L, Brendel V. 2001. Multi-query sequence BLAST output examination with MuSeqBox. *Bioinformatics*, **17**: 744–745.

Paquola AC, Machado AA, Reis EM, da Silva AM, Verjovski-Almeida S. 2003. Zerg: A very fast BLAST parser library. *Bioinformatics*, **19**: 1035–1036.

Pirooznia M, Perkins EJ, Deng Y. 2008. Batch Blast Extractor: An automated BLASTX parser application. *BMC Genomics*, **9**: S10.

Catanho M, Mascarenhas D, Degrave W, de Miranda AB. 2006. BioParser: A tool for processing of sequence similarity analysis reports. *Applied Bioinformatics*, **5**: 49–53.

Neumann RS, Kumar S, Shalchian-Tabrizi K. 2013. BLAST output visualization in the new sequencing era. *Briefings in Bioinformatics*, **15**: 484–503.

Nielsen CB, Cantor M, Dubchak I, Gordon D, Wang T. 2010. Visualizing genomes: Techniques and challenges. *Nature Methods*, **7**: S5–S15.

Eddy SR. 2009. A new generation of homology search tools based on probabilistic inference. *Genome Informatics*, **23**: 205–211.

Buchfink B, Xie C, Huson DH. 2015. Fast and sensitive protein alignment using DIAMOND. *Nature Methods*, **12**: 59–60.

Koboldt DC, Steinberg KM, Larson DE, Wilson RK, Mardis ER. 2015. The next-generation sequencing revolution and its impact on genomics. *Cell*, **155**: 27–38.

Ries E. 2011. *The lean startup: How today's entrepreneurs use continuous innovation to create radically successful businesses*. Crown Business, New York.

Thiel P, Masters B. 2014. *Zero to one: Notes on startups, or how to build the future*. Crown Business, New York, NY.

Altschul S, Demchak B, Durbin R, Gentleman R, Krzywinski M, Li H, Nekrutenko A, Robinson J, Rasband W, Taylor J, *et al.*. 2013. The anatomy of successful computational biology software. *Nature Biotechnology*, **31**: 894–897.

Lehman M. 1980. Programs, life cycles, and laws of software evolution. *Proceedings of the IEEE*, **68**: 1060–1076.

Flanagan D, Matsumoto Y. 2008. *The Ruby programming language*. O'Reilly, Beijing.

Ruby S, Thomas D. 2011. *Agile web development with Rails, 4th ed*. Pragmatic Bookshelf, Raleigh, NC.

Goto N, Prins P, Nakao M, Bonnal R, Aerts J, Katayama T. 2010. BioRuby: Bioinformatics software for the Ruby programming language. *Bioinformatics*, **26**: 2617–2619.

Prlić A, Procter JB. 2012. Ten simple rules for the open development of scientific software. *PLoS Computational Biology*, **8**: e1002802.

Wilson G, Aruliah DA, Brown CT, Chue Hong NP, Davis M, Guy RT, Haddock SHD, Huff KD, Mitchell IM, Plumbley MD, *et al.*. 2014. Best practices for scientific computing. *PLoS Biology*, **12**: e1001745.

Shore J, Warden S. 2007. *The art of agile development*. O'Reilly, Sebastopol, CA.

Weisfeld M. 2008. *The Object-Oriented Thought Process*. Pearson Education, Boston, MA.

Hunt A, Thomas D. 2000. *The pragmatic programmer: From journeyman to master*. Addison-Wesley Professional, Boston, MA.

Raymond ES. 2003. *The art of Unix programming*. Pearson Education, Boston, MA.

Sametinger J. 1997. *Software engineering with reusable components*. Springer-Verlag, New York, NY.

Ammann P, Offutt J. 2008. *Introduction to software testing*. Cambridge University Press, New York, NY.

Martin RC. 2008. *Clean code: A handbook of agile software craftsmanship*. Pearson Education, Upper Saddle River, NJ.

Wurm Y. 2015. Avoid having to retract your genomics analysis. *The Winnower*, **2**: e143696.68941.