SOFTWARE TOOL ARTICLE

Biobtree: A tool to search, map and visualize bioinformatics identifiers and special keywords [version 1; peer review: 1 approved with reservations, 1 not approved]

Tamer Gur

European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, CB10 1SD Cambridge, UK

v1 **First published:** 04 Feb 2019, **8**:145 (https://doi.org/10.12688/f1000research.17927.1)
Latest published: 16 Sep 2019, **8**:145 (https://doi.org/10.12688/f1000research.17927.2)

Abstract

Due to their nature, bioinformatics datasets are often closely related to each other. For this reason, search, mapping and visualization of these relations are often performed manually or programmatically via identifiers or special keywords such as gene symbols. Although various tools exist for these situations, the growing volume of bioinformatics datasets, emerging new software tools and approaches motivates new solutions. To provide a new tool for these current cases, I present the Biobtree bioinformatics tool. Biobtree effectively fetches and indexes identifiers and special keywords with their related identifiers from supported datasets, optionally with user pre-defined datasets and provides a web interface, web services and direct B+ tree data structure based single uniform database output. Biobtree can handle billions of identifiers and runs via a single executable file with no installation and dependency required. It also aims to provide a relatively small codebase for easy maintenance, addition of new features and extension to larger datasets. Biobtree is available to download from GitHub.

Keywords

bioinformatics, identifiers, search, mapping, visualization

This article is included in the International Society for Computational Biology Community Journal gateway.

This article is included in the EMBL-EBI collection.

Open Peer Review

Reviewer Status ✗ ?

	Invited Reviewers	
	1	**2**
REVISED **version 2** published 16 Sep 2019		
version 1 published 04 Feb 2019	✗ report	? report

1 **Maxim N. Shokhirev**, Salk Institute for Biological Studies, La Jolla, USA

2 **Samuel Lampa**, Uppsala University, Uppsala, Sweden

Any reports and responses or comments on the article can be found at the end of the article.

Corresponding author: Tamer Gur (tgur@ebi.ac.uk)

Author roles: Gur T: Conceptualization, Software, Writing – Original Draft Preparation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: The author(s) declared that no grants were involved in supporting this work.

How to cite this article: Gur T. **Biobtree: A tool to search, map and visualize bioinformatics identifiers and special keywords [version 1; peer review: 1 approved with reservations, 1 not approved]** F1000Research 2019, **8**:145 (https://doi.org/10.12688/f1000research.17927.1)

First published: 04 Feb 2019, **8**:145 (https://doi.org/10.12688/f1000research.17927.1)

Introduction

Bioinformatics datasets often consist of entries, where each entry is represented by unique identifier. Depending on the dataset, each entry contains various types of information such as sequence data, biological function, chemical structure or literature reference etc. In addition, entries often contain cross-referencing information to other dataset entries via identifiers. Let's take as an example entry the proto-oncogene vav protein in humans, which is encoded by the *VAV1* gene. If we display this protein on the UniProt website we see cross references to many other datasets. These cross references represent relations of datasets with each other. Various tools exist to deal with such data; however, the growing volume of bioinformatics datasets, emerging new software tools and analysis approaches motivates new solutions. Biobtree[1] presented herein is capable of improved and rapid processing of large numbers of unique identifiers of entries and related identifiers that are specified via cross-reference data.

In some datasets, in addition to unique identifiers there is information that is strongly related to entries but not necessarily unique for each entry. Species names or UniProt secondary accessions are example of this type. Information that is strongly related to the entries but not necessarily unique is a second data source for Biobtree. In Biobtree, these types are called special keywords and each of these can be related to multiple entries among the datasets.

Biobtree retrieves all these identifiers, related identifiers and special keywords from various bioinformatics resources and stores it in a single database. The data resources currently used are ChEBI[2], HGNC[3], HMDB[4], InterPro[5], Europe PMC[6] and UniProt[7]. Table 1 shows details of these datasets.

Based on stored data, Biobtree provides search, map and visualization functionalities via provided web services or a web interface. For instance, all the UniProt proteins entries belonging to a gene name, or, all Ensembl[8] genome transcripts identifiers and ENA[9] sequence identifiers that map to a protein identifier can be accessed. These relations, determined via identifiers, are stored bidirectionally so all actions can also be done in the opposite way.

Biobtree is managed from a single executable file for each major operating system without requiring any installation or compilation. As a database, Biobtree uses a B+ tree data structure based LMDB key value store. LMDB provides fast batch inserts and reads and allows effective operation on a large number of records. LMDB is embedded into Biobtree's executable binary code so it does not require a separate installation.

Methods

Implementation

Biobtree[1] has been implemented in GO programming language. To use C programming language based LMDB in Biobtree GO environment lmdb-go binding library has been used. To implement the Web interface Javascript programming language, Vue and Bulma web frameworks has been used. Biobtree workflow consists of three phases, which will be explained in later sections. These phases are named *update*, *generate* and *web* and are controlled by a Biobtree command line interface (CLI)

Update phase

The purpose of the update phase is to retrieve dataset identifiers and special keywords from remote servers or a local disk and produce files that contains identifiers and special keywords with their referred identifiers as keys and values in a sorted order. It is essential that the produced files are sorted to make fast batch inserts to LMDB database in the next phase. The updating phase is started via the Biobtree CLI with the update command. For example, the following command starts the update phase for the hgnc dataset.

```
$ biobtree --d hgnc update
```

Updating reads selected datasets as a stream and saves Biobtree-related data in a series of files. An advantage of reading dataset as streams is that it does not require fully downloading the dataset to the local disk. Datasets can have different formats like XML, JSON, TSV or CSV. Biobtree has specialized parsers for each dataset and parses them to produce its output files. When Biobtree runs the first time it retrieves its configuration, license and web interface files from the source code repository. Configuration files contain Biobtree runtime settings and dataset definitions.

Integrate user dataset

User data can be integrated to Biobtree. This feature creates an alternative for data providers to serve their data. Data should be gzipped and in an xml format compliant with UniProt xml schema definition. After the file path of the data is configured in a Biobtree configuration file, updating starts similarly:

```
$ biobtree --d my_data update
```

Table 1. List of datasets.

Dataset	Description	File Name	Location	Format	Special Keywords
ChEBI	ChEBI reference accession data	database_accession.tsv	ftp.ebi.ac.uk/chebi/Flat_file_tab_delimited/	TSV	-
HGNC	Human gene nomenclature	hgnc_complete_set.json	ftp.ebi.ac.uk/genenames/new/json/	JSON	name,symbol
HMDB	Human metabolome database	hmdb_metabolites.zip	http://www.hmdb.ca/system/downloads/current/	XML	name, synonyms
InterPro	Protein Families	interpro.xml.gz	ftp://ftp.ebi.ac.uk/pub/databases/interpro/current	XML	-
Literature mappings	Literature pmid, pmcid and doi mappings	PMID_PMCID_DOI.csv.gz	ftp://ftp.ebi.ac.uk/pub/databases/pmc/DOI/	CSV	-
Taxonomy	NCBI Taxonomy	taxonomy.xml.gz	ftp://ftp.ebi.ac.uk/pub/databases/taxonomy/	XML	scientificName
Uniparc	UniProt Sequence Archive	uniparc_all.xml.gz	ftp.ebi.ac.uk/pub/databases/uniprot/current_release/uniparc/	XML	-
UniProt reviewed	UniProt Knowledgebase reviewed	uniprot_sprot.xml.gz	ftp.ebi.ac.uk/pub/databases/uniprot/current_release/knowledgebase/complete/	XML	accession
UniProt unreviewed	UniProt Knowledgebase unreviewed	uniprot_trembl.xml.gz	ftp.ebi.ac.uk/pub/databases/uniprot/current_release/knowledgebase/complete/	XML	accession
Uniref50	UniProt sequence clusters	uniref50.xml.gz	ftp.ebi.ac.uk/pub/databases/uniprot/current_release/uniref/uniref50/		-
Uniref90	UniProt sequence clusters	uniref90.xml.gz	ftp.ebi.ac.uk/pub/databases/uniprot/current_release/uniref/uniref90/	XML	-
Uniref100	UniProt sequence clusters	uniref100.xml.gz	ftp.ebi.ac.uk/pub/databases/uniprot/current_release/uniref/uniref100/	XML	-

Updating in multiple computers

Biobtree supports executing the update phase over multiple computers. This is useful when it is necessary to use multiple computer processors at the same time such as with large datasets. The following two commands can be run on different computers with additional *idx* argument to guarantee that the produced files have unique names.

```
$ biobtree --d uniparc -idx 1 update
$ biobtree --d uniref50 -idx 2 update
```

Although Biobtree supports the updating phase occurring over multiple computers, for the next phase all the produced files have to be in a single location.

Generate phase

The purpose of this phase is to merge all the files produced in the update phase by keeping the sorted order and generate the final key and values in the generated LMDB database. Keys consist of identifiers and special keywords and values are identifiers that are referred to by these keys with their dataset information. If the values size for each key are above a certain threshold they are saved in pages. The following command starts the generate phase.

```
$ biobtree generate
```

The generated database output is used in next web phase but it can be also used directly. Example source codes for using the database directly can be found in the project github page.

Web phase

The purpose of this phase is to provide web services and a web interface via the produced output of the generate phase. The following command starts web phase

```
$ biobtree web
```

With this command the Biobtree web server is started instantly, serving the REST, gRPC and web interface services.

REST service

To make queries in the produced database, Biobtree provides RESTful endpoints with json-formatted responses. For example, each dataset has a unique identifier and other meta information like name, url template, etc. Dataset-unique identifiers are used in all the services to distinguish the dataset. These meta information is retrieved via the following endpoint:

http://localhost:8888/ws/meta

The following are used to query single or multiple identifiers or special keywords:

http://localhost:8888/ws/?idlist=vav_human

http://localhost:8888/ws/?idlist=vav_human,tpi1,brca2

To make a paging query for a certain identifier:

http://localhost:8888/ws/?id=vav_human&dataset=1&page=1

To make a filtering query based on a dataset:

http://localhost:8888/ws/?id=vav_human&dataset=1&filters=102

To make a paging query with active filtering:

http://localhost:8888/ws/?id=vav_human&dataset=1&filters=102&page=1

gRPC service

RESTful services are often json-based and are very convenient in json-based applications. But for non–json-based applications, it requires an extra process of serialization and deserialization. To address this, Biobtree provides a gRPC service with same functionality as its RESTful service. Sample codes for using gRPC in different languages can be found on the project github page. The following is snippet of Biobtree gRPC service definitions:

```
service BiobtreeService {

rpc Get     (BiobtreeGetRequest)     returns (BiobtreeGetResponse);
rpc GetPage (BiobtreeGetPageRequest) returns (BiobtreeGetPageResponse);
rpc Filter  (BiobtreeFilterRequest)  returns (BiobtreeFilterResponse);
rpc Meta    (BiobtreeMetaRequest)    returns (BiobtreeMetaResponse);

}
```

Web interface

The Web interface allow user to visualize the produced database via the RESTful service. Once the web phase is started it is accessed via the browser from the following address:

http://localhost:8888/ui

The Web interface provides searching of multiple identifiers and special keywords, visualizing and filtering results, and executing bulk queries. On the result page for each result, it provides a url to access the main website where the data are originally produced. Figure 1 and Figure 2 show the main and result page of the web interface.

Operation

Biobtree executable file is available from GitHub page for Windows, MacOS and Linux operating systems. For default datasets and configuration Biobtree uses up to 4 GB of memory. For large datasets it is advised to use computer which has large RAM space such as 16 GB to speed up finishing update and generate phases. RAM usage for these phases can be managed from configuration file via *kvgenChunkSize* and *batchSize* variables. For CPU Biobtree uses all available CPU powers when it needed. This default behaviour can be restricted via CLI with *maxcpu* argument.

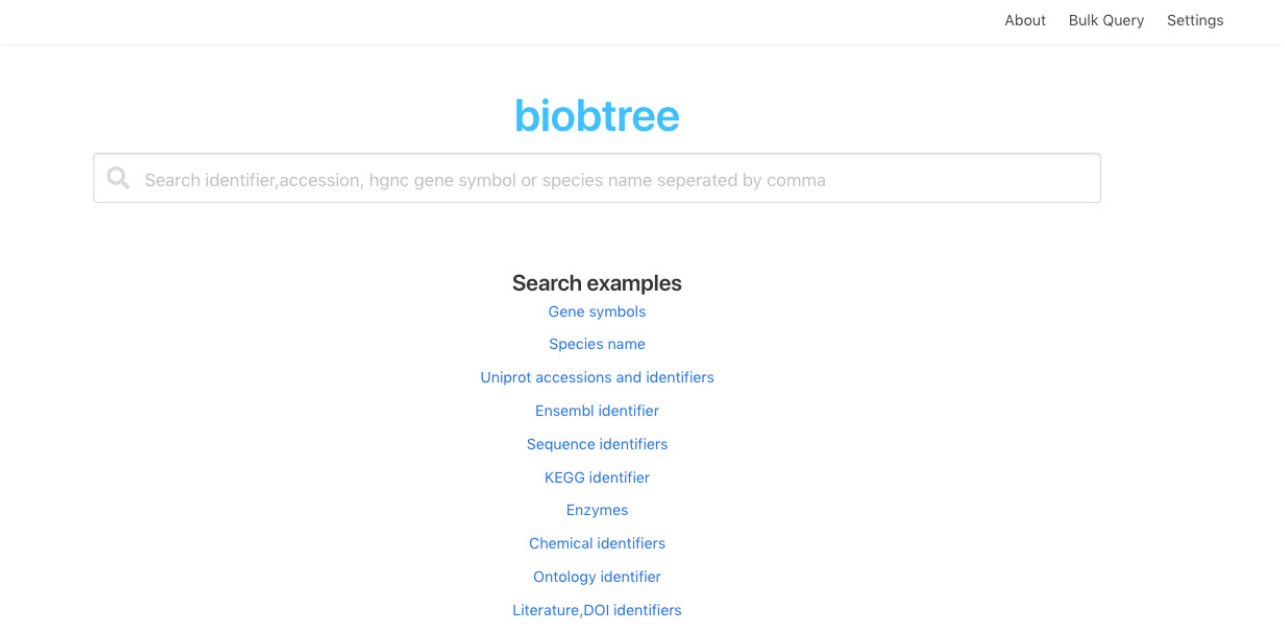

Figure 1. Web interface main page.

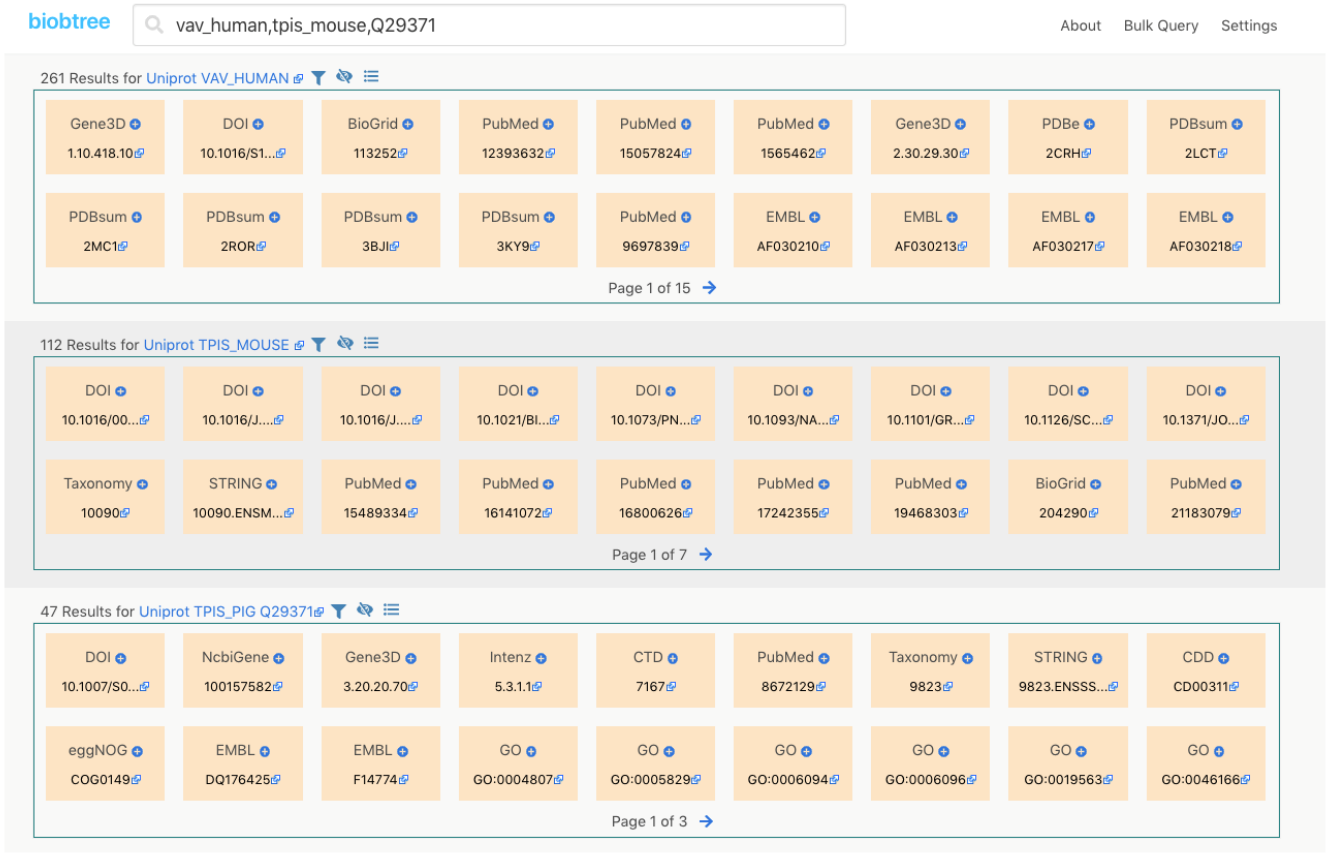

Figure 2. Web interface result page.

Benchmarks

Software benchmarks should often be taken with a grain of salt, especially where the input data is large, because benchmarks results can be affected by many factors like the processor, operating system, storage, network speed, input data, application parameters etc. Considering these, the purpose of Biobtree benchmarks is mainly to show overall capabilities and resource usage of Biobtree. Table 2 shows the benchmark details and results. Benchmarks were primarily computed at DigitalOcean London datacentres using their CPU optimized droplets with block storage volumes. Hundred thousand sample query which used in the benchmarks can be found on the project GitHub page.

Discussion

The benchmarks show that Biobtree[1] has produced LMDB database output in relatively acceptable times. Let's discuss how Biobtree behaves if UniProt provides tens of times larger data. Clearly, more disk space would have been needed.

If enough disk space is provided, the next obstacle would have happened during the update phase, because currently UniProt provides a single gzip compressed file for each dataset and Biobtree reads each file as a stream from the beginning to end. Gzip does not allow a random access to a file unless there are checkpoints defined. This characteristic of gzip prevents the processing of a single large file in a split manner and utilizes more computing resources if available. Two solutions can address the issue. The first could be for UniProt to allow its datasets to be capable of parallel processing, like splitting and compressing files inside a tar archive. The second solution would be to implement a new functionality in Biobtree and save and decompress these files to local disk and make parallel access to decompressed files.

A further obstacle would have happened during the generate phase, since we would have obtained more files from update phase. The generate phase struggles to merge all these files and could cause much longer output generation times. To address this obstacle, a new phase could be implemented that runs before the generate phase and merges files coming from the update phase.

Table 2. Benchmarks results.

-	Benchmark-1	Benchmark-2	Benchmark-3	Benchmark-4	Benchmark-5	Benchmark-6	Benchmark-7	Benchmark-8	Benchmark-9
Description	Runs all the phases and 100K query for default datasets at macbook laptop	Runs default dataset update phase at digital ocean.	Runs uniref50 update phase at digital ocean.	Runs uniref90 update phase at digital ocean.	Runs uniref100 update phase at digital ocean.	Runs uniprot unreviewed update phase at digital ocean.	Runs uniparc update phase at digital ocean.	Runs generate phase for all data at digital ocean.	Runs 100K query against all data at digital ocean.
Operating System	macOS High Sierra	Ubuntu 16.04.5 x64	Ubuntu 16.04.5 x64	Ubuntu 16.04.5 x64	Ubuntu 16.04.5 x64	Ubuntu 16.04.5 x64	Ubuntu 16.04.5 x64	Ubuntu 16.04.5 x64	Ubuntu 16.04.5 x64
CPU	4	8	8	8	8	8	8	8	8
RAM	16GB	16GB	16GB	16GB	16GB	16GB	16GB	16GB	16GB
Update Phase Average CPU Usage	309%	556%	201%	201%	204%	303%	351%	-	-
Update Phase Average RAM Usage	9%	84%	39%	36%	36%	36%	37%	-	-
Update Phase Elapsed Duration	13m	5m	1h24m	2h3m	2h43m	8h27m	8h37m	-	-
Update Phase Files Size	838MB	823MB	3GB	2.8GB	2.4GB	25GB	38GB	-	-
Generate Phase Average CPU Usage	141%	-	-	-	-	-	-	164%	-
Generate Phase Average RAM Usage	19%	-	-	-	-	-	-	90%	-
Generate Phase Elapsed Duration	18m	-	-	-	-	-	-	8h5m	-
Generate Phase Files Size	3.8GB	-	-	-	-	-	-	392GB	-
Web Phase Average CPU Usage While Running 100K Query	113%	-	-	-	-	-	-	-	3%
Web Phase Average RAM Usage While Running 100K Query	2%	-	-	-	-	-	-	-	36%
100K Consequent Query Average Response Time Queries Sent Inside Same Machine	0.3ms	-	-	-	-	-	-	-	3ms
100K Consequent Query Average Response Time Queries Sent From Outside Network and Machine	-	-	-	-	-	-	-	-	20ms
Total Keys	60M	-	-	-	-	-	-	3.19B	-
Total Values	131M	-	-	-	-	-	-	17.07B	-

Limitations

Although duplicate values for each key are discarded during the update phase for each dataset, the generate phase could rarely produce duplicate values. The user needs to discard these duplicate records manually if these are created. Another limitation is when querying the special keywords, they need to be fully specified including all space characters.

Future work

The limitations can be addressed and new functionalities added in the future. For instance, different bioinformatics datasets like Ensembl[8] or ENA[9] can be integrated. Another feature would be sorting result values based on a certain criterion.

Data availability

All data underlying the results are available as part of the article and no additional source data are required.

Software availability

All source codes and binaries available at: https://www.github.com/tamerh/biobtree.

Archived source code at time of publication: https://doi.org/10.5281/zenodo.2547047[1].

Grant information

The author(s) declared that no grants were involved in supporting this work.

References

1. Gür T: **tamerh/biobtree: biobtree v1.0.0-rc2 (Version v1.0.0-rc2).** *Zenodo.* 2019.
http://www.doi.org/10.5281/zenodo.2547047
2. Hastings J, Owen G, Dekker A, *et al.*: **ChEBI in 2016: Improved services and an expanding collection of metabolites.** *Nucleic Acids Res.* 2016; **44**(D1): D1214–D1219.
PubMed Abstract | Publisher Full Text | Free Full Text
3. Yates B, Braschi B, Gray KA, *et al.*: **Genenames.org: the HGNC and VGNC resources in 2017.** *Nucleic Acids Res.* 2017; **45**(D1): D619–625.
PubMed Abstract | Publisher Full Text | Free Full Text
4. Wishart DS, Feunang YD, Marcu A, *et al.*: **HMDB 4.0: the human metabolome database for 2018.** *Nucleic Acids Res.* 2018; **46**(D1): D608–17.
PubMed Abstract | Publisher Full Text | Free Full Text
5. Mitchell AL, Attwood TK, Babbitt PC, *et al.*: **InterPro in 2019: improving coverage, classification and access to protein sequence annotations.** *Nucleic Acids Res.* 2019; **47**(D1): D351–D360.
PubMed Abstract | Publisher Full Text | Free Full Text
6. Europe PMC Consortium: **Europe PMC: a full-text literature database for the life sciences and platform for innovation.** *Nucleic Acids Res.* 2015; **43**(Database issue): D1042–D1048.
PubMed Abstract | Publisher Full Text | Free Full Text
7. The UniProt Consortium: **UniProt: a worldwide hub of protein knowledge.** *Nucleic Acids Res.* 2019; **47**(D1): D506–D515.
PubMed Abstract | Publisher Full Text | Free Full Text
8. Zerbino DR, Achuthan P, Akanni W, *et al.*: **Ensembl 2018.** *Nucleic Acids Res.* 2018; **46**(D1): D754–D761.
PubMed Abstract | Publisher Full Text | Free Full Text
9. Harrison PW, Alako B, Amid C, *et al.*: **The European Nucleotide Archive in 2018.** *Nucleic Acids Res.* 2019; **47**(D1): D84–D88.
PubMed Abstract | Publisher Full Text | Free Full Text

Open Peer Review

Current Peer Review Status: ✗ ?

Version 1

Reviewer Report 15 April 2019

https://doi.org/10.5256/f1000research.19605.r46335

?

Samuel Lampa
Department of Pharmaceutical Biosciences, Uppsala University, Uppsala, Sweden

The article describes a commandline tool, Biobtree, that is claimed to allow to process relations between bioinformatics datasets based on various characteristics such as identifiers and keywords.

The manuscript describes the tool in a clear way technically, making it quite clear what it does in technical terms, and how it is supposed to be used.

Also, I was able to install and run the tool in a simple way on my laptop (i5 CPU, 8GB RAM and 10-15 GB free hard drive, Xubuntu 16.04 64 bit) without problems. It provides a simple but good looking and easy to use web interface.

I'm seeing at least two major issues with the tool and manuscript though, that needs being thoroughly addressed to make them acceptable.

Main problem 1: Visualization?

Firstly, the title claims that the tool does visualization of the database produced by the tool. Perhaps I'm missing something, but I have not found any visualization in the tool apart from a form of search hit result listings. I don't think this is enough to be called "visualization". Especially as it is unclear how the current form of output is supposed to be used in a concrete biological usecase. With the current wording, I would expect something more graphical, like a graphviz-like graph view of dataset relations.

Suggested edits to make the tool and paper acceptable:

- Provide graphical visualization beyond results listings (or explain how to show them, if I have missed them), or else remove "visualization" from the title and other places.
- Use this/these visualizations in the use cases/demonstrators discussed above, to explain how they contribute to solving concrete biological problems.

Main problem 2: Lack of context and discussion of biological relevance

The first and main problem with the manuscript is that it does not provide a clear enough description of

what *biological* problem it is solving. Nor does it provide an overview of existing tools and solutions in this field. Right now, the manuscript only states what the tool can do in technical terms. It somehow reads like a (well written) user guide or README file, but not yet a scientific paper. To help potential new users understand why they might need this tool, it needs to be put in context and compared with other existing tools.

In my view, the manuscript needs the following points thoroughly addressing to be acceptable:

1. In the introduction: Elaborate on the field of mapping/visualising dataset relations, mentioning relevant existing similar tools, what are the typical problems, and what particular problem Biobtree solves.
2. Explain a few examples of *biological* problems that can be solved with this tool, or type of tool.
3. E.g. in the results: Provide at least one, and optimally two or three potentially simple, but relevant, biological demonstrators or use cases, that can be addressed with the tool. Provide complete instructions on how to re-run this or these demo(s) and provide outputs for this/these in terms of figures or diagrams and how these were produced. In this way, both reviewers and users can make sure that they understand how to operate the tool.
4. In the discussion: Connect back to the explained problem the tool is addressing, and explain how the problem was solved, again reinstating the relevance of this specific tool compared to other existing tools, and what improvement it provides to the end user trying to solve biological problems, exemplified by the demonstrators or use cases.

Language issues

The manuscript also contains quite a number of language issues. I'm listing a few language suggestions below as examples, but further language proofing or editing is highly recommended, to make sure there are not more of these:

1. **Methods** section:
   "in GO programming language" --> "in the Go programming language"
   (Note the "the" and that only G is uppercase in "Go").
2. **Update phase** section:
   "to LMDB" -> "to the LMDB"
3. **Update phase** section:
   "Updating reads selected datasets as a stream"
   I don't understand this sentence. Please language-check it.
4. "is used in next" -> "is used in the next"
5. **Generate phase** section:
   "the project github page" -> "the project's GitHub page"
6. **Web phase** section:
   "starts web phase" -> "starts the web phase"
7. **Web interface** section:
   "The Web interface allow user" -> "The web interface allows the user"
8. **Operation** section:
   "Biobtree executable" -> "The biobtree executable"

Is the rationale for developing the new software tool clearly explained?

Partly

Is the description of the software tool technically sound?

Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Scientific workflow tools, Cheminformatics, Semantic web.

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Reviewer Report 07 March 2019

https://doi.org/10.5256/f1000research.19605.r45074

Maxim N. Shokhirev
Razavi Newman Integrative Genomics and Bioinformatics Core, Salk Institute for Biological Studies, La Jolla, CA, USA

While it is important to create a consistent and queryable database of biological identifiers, it is unclear what advances this tool brings to the field. For example, how does this tool compare to other queryable database tools such as mygene.info, or BioMart? The paper will greatly benefit from a comparison to these and other such tools.

I downloaded and ran the tool but it seems I can't get through the update phase when I run ./biobtree update (It seems to hang after uniprot_reviewed finishes) without any other messages. When I rerun using biobtree --d uniprot_reviewed update it finishes but there is an error:

Error while reading file-> .//out/index/0_13.938476000.gz
panic: gzip: invalid header

I tried running generate and web after that regardless, but couldn't get it to work:

panic: mdb_txn_commit: MDB_BAD_TXN: Transaction must abort, has a child, or is invalid

Error while reading meta information file which should be produced with generate command. Please make sure you did previous steps correctly.

The author needs to debug/test their code to ensure that it can be used by others.

Is the rationale for developing the new software tool clearly explained?
Partly

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
No

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: Bioinformatics

I confirm that I have read this submission and believe that I have an appropriate level of expertise to state that I do not consider it to be of an acceptable scientific standard, for reasons outlined above.

Author Response 10 Mar 2019

Tamer Gur, EMBL European Bioinformatics Institute, UK

Thank you for reviewing the article. I agree that there are several similar tools exist with different dataset and functionalities such as Biomart and mygene.info. However, this tool can still complement them for following main reasons.

- Biobtree can work in local machine. This can be especially useful when large number of requests needs to be performed. For instance currently similar Uniprot tool documentation suggests either split the requests or download underlying data when number of requests are above 50K. These types of limitations for bulk requests are sensible for fair usage of a public service and can be more suitable with Biobtree type locally runnable tool.
- Users custom dataset can be integrated.
- Tool provides new intuitive web interface.

In relation to reported errors, I have added demo of tool in case such errors happen again. I have also added integration test which runs periodically on Linux, MacOS and Windows operating systems via Azure DevOps platform. These tests can be accessed publicly and test and demo

links can be found at github page. Based on these tests, it seems that tool is working as expected. I believe that hanged process is a specific issue or bug which I am happy to resolve if I have more information. The rest of problems which have been reported are most probably due to the prematurely exited hanged process. Either starting in a new folder or passing --clean parameter can solve the issue.

Competing Interests: No competing interests were disclosed.