

## SOFTWARE

# PubData: search engine for bioinformatics databases worldwide

Bohdan B. Khomtchouk<sup>1\*</sup>, Kasra A. Vand<sup>2</sup>, Thor Wahlestedt<sup>1,3</sup>, Kelly Khomtchouk<sup>4</sup>, Mohammed K. Sayed<sup>5</sup> and Claes Wahlestedt<sup>1</sup>

\*Correspondence:

[b.khomtchouk@med.miami.edu](mailto:b.khomtchouk@med.miami.edu)

<sup>1</sup>Center for Therapeutic Innovation and Department of Psychiatry and Behavioral Sciences, University of Miami Miller School of Medicine, 1120 NW 14th ST, Miami, FL, USA 33136

Full list of author information is available at the end of the article

## Abstract

We propose a search engine and file retrieval system for all bioinformatics databases worldwide. PubData searches biomedical data in a user-friendly fashion similar to how PubMed searches biomedical literature. PubData is built on novel network programming, natural language processing, and artificial intelligence algorithms that can patch into the file transfer protocol servers of any user-specified bioinformatics database, query its contents, retrieve files for download, and adapt to the user's search preferences.

PubData is hosted as a user-friendly, cross-platform graphical user interface program developed using PyQt : <http://www.pubdata.bio>. The methods are implemented in Python, and are available as part of the PubData project at: <https://github.com/Bohdan-Khomtchouk/PubData>.

## Introduction

If there was a data-oriented counterpart to PubMed Central (Roberts, 2001) in today's ever-expanding data world, it would undoubtedly be PubData, a centralized repository dedicated to data in the life sciences. While biomedical literature is typically accessed via search engines such as PubMed or other software tools (Hokamp & Wolfe 2004, Fontaine et al. 2009, States et al. 2009, Lu 2011, Wang et al. 2014, Gou et al. 2015, Squizzato et al. 2015, bioCADDIE 2016, FORCE11 2016), the process of searching biomedical data often entails circuitously accessing various bioinformatics databases on a case-by-case basis, depending on the nature of the data sought by the user. Likewise, the search process often begins at the literature level, whereupon the user retrieves the respective open-access data only once the appropriate literary sources are located. This, in turn, impedes the efficiency of the data search process and prohibits the user from searching the other way around (e.g., finding a list of papers that match the specifics of the data requested by the user).

Although biomedical data repositories such as the Gene Expression Omnibus (Edgar et al. 2002, Barrett et al. 2013) and the Sequence Read Archive (Kodama et al. 2012) offer a fairly comprehensive data search experience, they do not support the search of niche databases, which coincidentally comprise a significant majority of the ever-growing bioinformatics database ecosystem. Likewise, existing resources do not allow the user to search multiple databases simultaneously in a case-specific manner (e.g., search concurrently in Uniprot, Ensembl, and the UCSC Genome Browser, but exclude other databases). In general, as the amount of data relevant

to the biological sciences steadily grows and the number of bioinformatics databases progressively expands, a new initiative is imperative to establish a truly comprehensive electronic archive of data from peer-reviewed literature sources. This kind of resource must be made flexible in searching and intuitively navigating existing data repositories.

To this end, we propose PubData, currently offered as a user-friendly, cross-platform (Mac OS X, Windows, Linux) graphical user interface (GUI) search engine capable of accessing the file transfer protocol (FTP) servers of any bioinformatics database in the world and searching/retrieving their contents. Although the idea of the application of search engine technology to bioinformatics has a rich history (Liebel *et al.* 2004, Liebel *et al.* 2005, Marinescu *et al.* 2005, Morrison *et al.* 2005, Page 2005, Hearst *et al.* 2007, Lewis *et al.* 2012, Mandloi & Chakrabarti 2015, DeFreitas *et al.* 2016, Zhang *et al.* 2016), it has never been attempted at such broad scale. As such, we propose the first search engine designed to search data files in bioinformatics databases worldwide. We aim to provide the scientific community with the ability to conduct a “Google-style” search for biomedical data, thereby taking advantage of a data-oriented resource akin to the literature-oriented resource of PubMed.

PubData can remotely access, search, and retrieve files from the deeply nested directory trees of any major bioinformatics database via a local computer network. By assembling all major bioinformatics databases under the roof of one software program, PubData allows the user to avoid the unnecessary hassle and non-standardized complexities inherent to accessing databases one-by-one using an Internet browser. PubData allows a user to query multiple databases simultaneously for user-specified keywords (e.g., human, cancer, transcriptome), and to manually add support for any additional bioinformatics databases as they come into existence. As such, PubData allows researchers to access, search, view, and download files from the FTP servers of any major bioinformatics database directly from one centralized location. By using only a GUI, PubData allows the user to simultaneously surf multiple bioinformatics FTP servers directly from the comfort of their local computer.

## Results & Discussion

PubData (Figure 1) has been written in the Python programming language (Python Software Foundation, 2016) and the user-interface has been designed using PyQt, which is the Python binding of the cross-platform GUI toolkit Qt (PyQt, 2016). In order to download or access biological files of interest, users typically connect to FTP servers. This could be very time-consuming and the method has inherent flaws. For instance, it is impossible to search files based on a property, or search concurrently in multiple databases.

In PubData, we have resolved these issues. Now a user can search a specific keyword in several databases and see relevant results within seconds. PubData utilizes techniques in natural language processing (NLP) in order to retrieve information in an intuitive way. For instance, querying words that are semantically related (e.g. homo and human and mankind) leads to similar search results that are returned to a user within a short period of time. For implementation of the NLP module in PubData, we used the Natural Language Toolkit (NLTK, 2016).

The FTP (File Transfer Protocol) is a standard network protocol used to transfer computer files between a client and server on a computer network. Hence, in using FTP, there is no access to the entire directory tree. While traversing the directory tree on a server using FTP, users need to send a series of requests to a server (e.g. changing a path, downloading a file, retrieving the list of files in a directory, etc) and, as the size of the database gets larger, the transaction will inevitably take a lot of time (sometimes several hours). But since crawling/searching through a directory tree is the most time-consuming (and computationally expensive) procedure, we have designed PubData to perform this process locally as a background process rather than through FTP (Figure 2).

To achieve this, we implemented a search module using the Breadth First Search (BFS) algorithm and a homebrewed FTP module, FTPWalker, and threaded this function through a parallelized and concurrent algorithm that divides the subdirectories within the root amongst multiple processes (based on available processors). Then, each process automatically divides the subdirectories between multiple threads that use FTPWalker in order to crawl through an FTP server's entire directory tree. Since all we needed to deal with were filenames and their respective paths, we traversed all the FTP databases using the aforementioned method only once, and subsequently preserved the paths and filenames in a local SQLite database. The advantage of this approach is that at search time we can just search in this local database, and after extracting the relative paths we can simply access file directories by one single request, which makes all the computational processes extremely fast. The only drawback to this approach was that this method overlooked new files when FTP servers were updated. We circumvented this issue by providing the users with an update mechanism, so that users can manually update the databases whenever they like via a set of update buttons.

To efficiently crawl through an entire FTP server directory tree, a series of computational innovations in the form of parallelized algorithms were developed in PubData (e.g., the FTPTraverse class and launcher.py). Although multithreading improves the performance of the application, there is no precise formula for calculating the algorithmic time complexity, since it depends on multiple factors. For instance, two threads can gain a speedup of up to 2X on-average, and four threads up to 3X. However, the potential speedup is bound to the available physical threads of the CPU and is dependent on balancing the computational workload. Specifically, we built a function called "find\_leading" that can find all directories and their respective subdirectories within the root directory. This function returns the leading directories when it encounters more than one directory with the corresponding path. Then the "main-walker" module divides the directories within root amongst the available processors by calling the "main\_run" function from the "traverse" module. Finally, this function borrows methods from the "find\_leading" function in order to find the subdirectories and partition them between threads (based on available ones) by calling the "traverse\_branch" function, which itself uses the "ftp-walker" module for traversing the FTP server. This module uses the BFS algorithm and Python's "ftplib" module in order to be adapted for traversing an FTP directory tree. As such, performance gains depend mostly on the number of directories within the root and the number of subdirectories within them (i.e., maximum depth). In general,

executing two threads gives an increase in performance of up to 2X the original execution time. Using four threads, provides an increase of 2.5X to 3X the original execution time. Hence, if the number of CPUs is  $C$  and the regular execution time is  $S$ , the new execution time would approximately be  $S/C \times 3$  to  $S/C \times 2.5$ .

PubData provides users with efficient implementations of the following features:

- Selecting a server name and connecting to a corresponding FTP database in order to manually explore the database from root.
- Modification of the server list, include the ability to add new servers and deleting/editing the existing ones.
- Selecting an arbitrary number of local servers and searching among them.
- Automatically searching through all the servers.
- Suggesting the most frequently used words at search time, via a built-in recommender system (Figure 3).
- Match semantically related words via an NLTK general WordNet and a manual WordNet.
- Downloading any accessed files.
- Providing the metafile for each server, so that users can see the corresponding metafiles for each server after opening a specific path.

Users will end up with a list of paths that contain files matching their queries, either semantically or literally (Figure 1). Then users can open a direct connection to a selected path and get access to a file directory on the server and download the files.

In general, to search a database, users enter a keyword and see two types of results:

- The keyword appears exactly in the file or directory names.
- The result is semantically and scientifically related to the keyword.

For semantic analysis, we used two kinds of WordNets. The first one is a general WordNet provided by the NLTK library, which only gives us some common relative results for a word. Here are some examples of synonyms given from the NLTK WordNet:

Sample input: “human”

Output: set(['human\_being', 'homo', 'human', 'man'])

Sample input: “RNA”

Output: set(['RNA', 'ribonucleic\_acid'])

The NLTK corpora are general purpose and their coverage might not be adequate for an application in the realm of biology. Due this point and the fact that still there is no complete and free WordNet for biology, we created a new one. Here is the way that we created this biological WordNet:

We started by parsing two biological encyclopedias/dictionaries (Rittner & McCabe 2004, Singleton 2010). For all the words within these books, we extracted all the nouns from their corresponding description and then we refined those words by removing any general and irrelevant nouns. For example, here is a word alongside its relative nouns, as determined by PubData:

“adenylyl cyclase”: [“monophosphate”, “cytoplasm”, “signal”, “cAMP”, “molecule”, “adeno”, “enzyme”, “receptor”, “membrane”, “plasma”]

## Future Perspectives

We plan to continue improving the NLP aspects of PubData by adding more semantic analyses, as applied to query words, and improving the accuracy of the recommender system. We also plan to create a web-based version of PubData, allowing for platform-independent, web-browser accessibility for biologists.

## Conclusion

We provide access to a user-friendly graphical user interface program designed to access, search, retrieve, and download data files from any bioinformatics database in the world. We have gathered all the bioinformatics databases worldwide in PubData and let scientists search them quickly and efficiently. In addition, the use of semantic analysis has made the search engine retrieval system more intelligent and useful via a built-in recommender system. Since we realize that PubData is a community-driven project, we have made available the source code on GitHub with the vision that this encourages scientists and developers to help us improve the application. To encourage open-source contributions to PubData, we plan to determine future co-authorship on subsequent PubData publications by the number and quality of Github commits from the developer community.

### Competing interests

The authors declare that they have no competing interests.

### Author's contributions

BBK conceived the study. BBK and KAV wrote the code. BBK, KAV, TW, KK, and MKS tested the code. TW created the website. CW provided project resources and participated in the management of the source code and its coordination. BBK wrote the paper. All authors read and approved the final manuscript.

### Acknowledgements

BBK wishes to acknowledge the financial support of the United States Department of Defense (DoD) through the National Defense Science and Engineering Graduate Fellowship (NDSEG) Program: this research was conducted with Government support under and awarded by DoD, Army Research Office (ARO), National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a.

### Author details

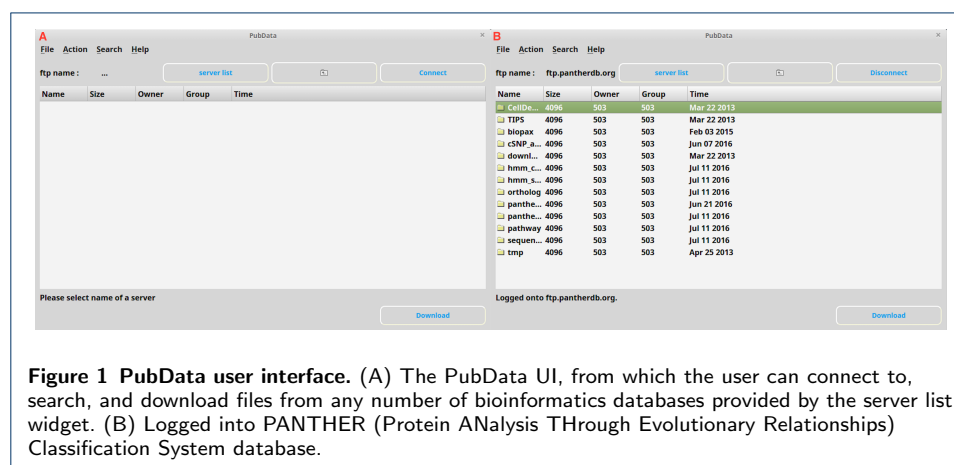
<sup>1</sup>Center for Therapeutic Innovation and Department of Psychiatry and Behavioral Sciences, University of Miami Miller School of Medicine, 1120 NW 14th ST, Miami, FL, USA 33136. <sup>2</sup>kasraavand@gmail.com. <sup>3</sup>Ransom Everglades School, 3575 Main Highway, Coconut Grove, FL, USA 33133. <sup>4</sup>Department of Microbiology and Immunology, University of Miami Miller School of Medicine, 1600 NW 10th Ave, Miami, FL, USA 33136. <sup>5</sup>Department of Electrical and Computer Engineering, University of Miami, 1251 Memorial Drive, Coral Gables, FL, USA 33146.

### References

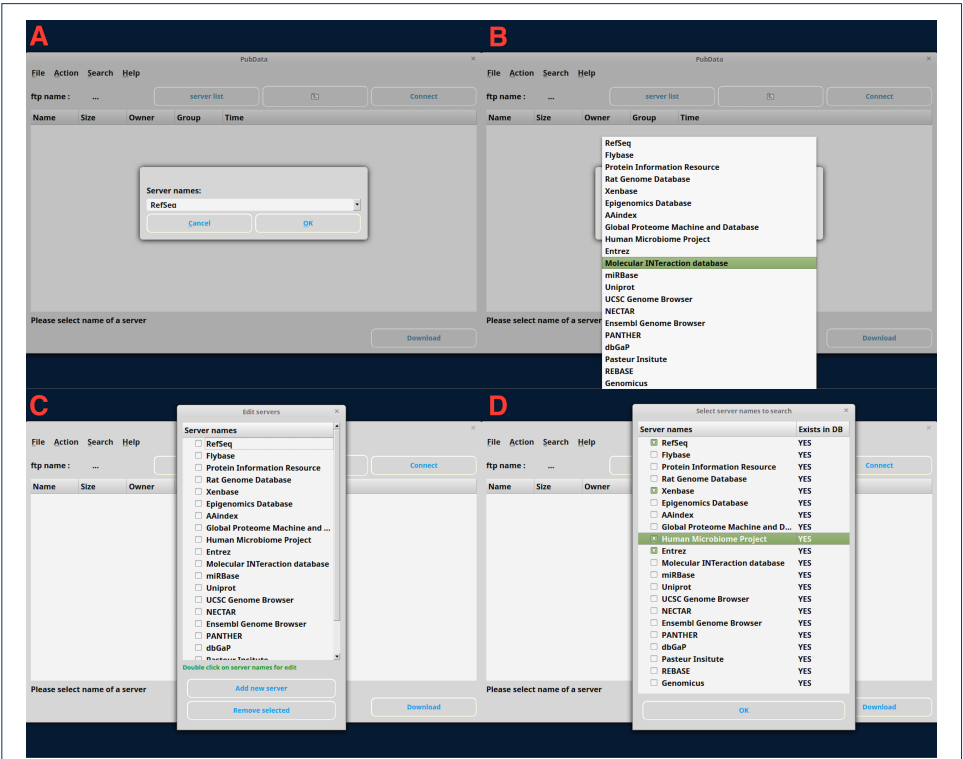
1. Barrett T, Wilhite SE, Ledoux P, Evangelista C, Kim IF, Tomashevsky M, Marshall KA, Phillippy KH, Sherman PM, Holko M, Yefanov A, Lee H, Zhang N, Robertson CL, Serova N, Davis S, Soboleva A: *NCBI GEO: archive for functional genomics data sets—update*. Nucleic Acids Research. 2013, 41 (Database issue): D991–995.
2. bioCADDIE: biomedical and healthCare Data Discovery Index Ecosystem. 2016. <https://biocaddie.org>.
3. Fontaine JF, Barbosa-Silva A, Schaefer M, Huska MR, Muro EM, Andrade-Navarro MA: *MedlineRanker: flexible ranking of biomedical literature*. Nucleic Acids Research. 2009, 37 (Web Server issue): W141–146.
4. FORCE11: The Future of Research Communications and e-Scholarship. 2016. <https://www.force11.org>.
5. DeFreitas T, Saddiki H, Flaherty P: *GEMINI: a computationally-efficient search engine for large gene expression datasets*. BMC Bioinformatics. 2016, 17:102.
6. Edgar R, Domrachev M, Lash AE: *Gene Expression Omnibus: NCBI gene expression and hybridization array data repository*. Nucleic Acids Research. 2002, 30(1): 207–210.

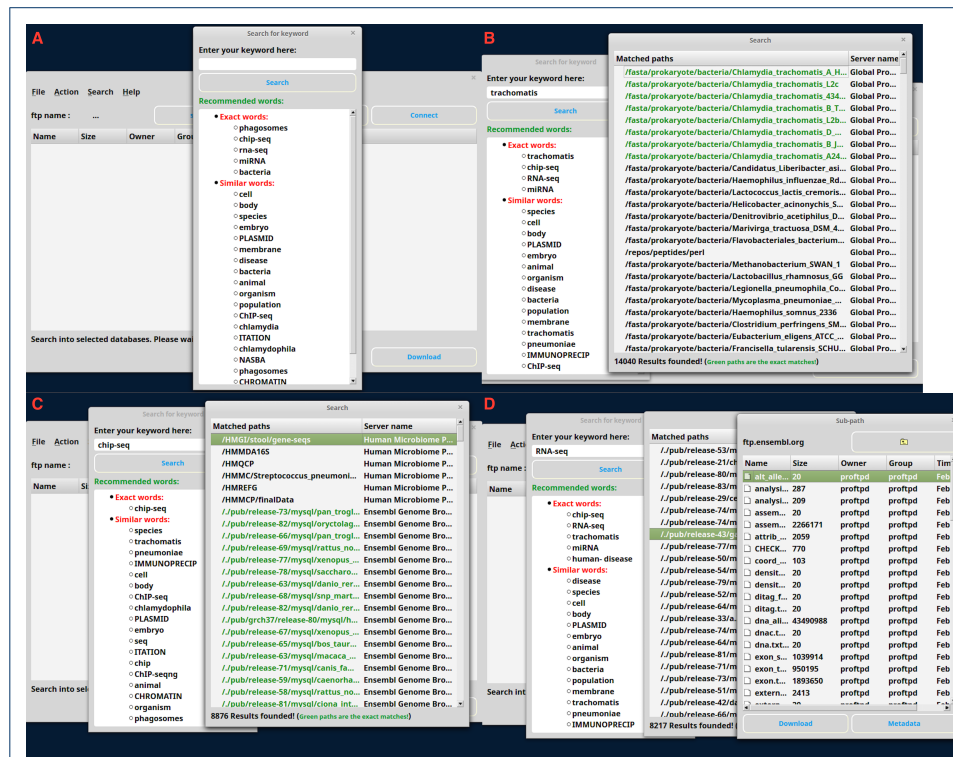
7. Gou Y, Graff F, Kilian O, Kafkas S, Katuri J, Kim JH, Marinos N, McEntyre J, Morrison A, Pi X, Rossiter P, Talo F, Vartak V, Coleman LA, Hawkins C, Kinsey A, Mansoor S, Morris V, Rowbotham R, Chaplin D, MacIntyre R, Patel Y, Ananiadou S, Black WJ, McNaught J, Rak R, Rowley A: *Europe PMC: a full-text literature database for the life sciences and platform for innovation*. Nucleic Acids Research. 2015, 43 (Database issue): D1042–1048.
8. Hearst MA, Divoli A, Guturu H, Ksikes A, Nakov P, Wooldridge MA, Ye J: *BioText Search Engine: beyond abstract search*. Bioinformatics. 2007, 23(16): 2196–2197.
9. Hokamp K, Wolfe KH: *PubCrawler: Keeping up comfortably with PubMed and GenBank*. Nucleic Acids Research. 2004, 32 (Web Server issue): W16–W19.
10. Jouper K, Nordin H: *Performance analysis of multithreaded sorting algorithms*. 2015. Thesis no: BCS-2015-05, Dept. Computer Science & Engineering. Blekinge Institute of Technology SE–371 79 Karlskrona, Sweden.
11. Kodama Y, Shumway M, Leinonen R, International Nucleotide Sequence Database Collaboration: *The Sequence Read Archive: explosive growth of sequencing data*. Nucleic Acids Research. 2012, 40 (Database issue): D54–56.
12. Lewis S, Csordas A, Killcoyne S, Hermjakob H, Hoopmann MR, Moritz RL, Deutsch EW, Boyle J: *Hydra: a scalable proteomic search engine which utilizes the Hadoop distributed computing framework*. BMC Bioinformatics. 2012, 13:324.
13. Liebel U, Kindler B, Pepperkok R: *'Harvester': a fast meta search engine of human protein resources*. Bioinformatics. 2004, 20(12): 1962–1963.
14. Liebel U, Kindler B, Pepperkok R: *Bioinformatic "Harvester": A Search Engine for Genome-Wide Human, Mouse, and Rat Protein Resources*. Methods in Enzymology. 2005, 404: 19–26.
15. Lu Z: *PubMed and beyond: a survey of web tools for searching biomedical literature*. Database (Oxford). 2011, baq036.
16. Mandloi S, Chakrabarti S: *PALM-IST: Pathway Assembly from Literature Mining—an Information Search Tool*. Scientific Reports. 2015, 5:10021.
17. Marinescu VD, Kohane IS, Riva A: *MAPPER: a search engine for the computational identification of putative transcription factor binding sites in multiple genomes*. BMC Bioinformatics. 2005, 6:79.
18. Morrison JL, Breitling R, Higham DJ, Gilbert DR: *GeneRank: Using search engine technology for the analysis of microarray experiments*. BMC Bioinformatics. 2005, 6:233.
19. Natural Language Toolkit. NLTK 3.0 documentation. 2016. <http://www.nltk.org>.
20. Page RDM: *A Taxonomic Search Engine: Federating taxonomic databases using web services*. BMC Bioinformatics. 2005, 6:48.
21. Python Software Foundation. Python Language Reference, version 2.7. Available at <http://www.python.org>.
22. PyQt. <https://wiki.python.org/moin/PyQt>.
23. Rittner D, McCabe TL: *Encyclopedia of Biology (Science Encyclopedia)*. Facts on File, 2004.
24. Roberts RJ: *PubMed Central: The GenBank of the published literature*. Proceedings of the National Academy of Sciences. 2001, 98(2): 381–382.
25. Singleton P: *Dictionary of DNA and Genome Technology, Second Edition*. John Wiley & Sons, Inc. 2010.
26. Squizzato S, Park YM, Buso N, Gur T, Cowley A, Li W, Uludag M, Pundir S, Cham JA, McWilliam H, Lopez R: *The EBI Search engine: providing search and retrieval functionality for biological data from EMBL-EBI*. Nucleic Acids Research. 2015, 43 (Web Server issue): W585–W588.
27. States DJ, Ade AS, Wright ZC, Bookvich AV, Athey BD: *MiSearch adaptive pubMed search tool*. Bioinformatics. 2009, 25(7): 974–976.
28. Wang JZ, Zhang Y, Dong L, Li L, Srimani PK, Yu PS: *G-Bean: an ontology-graph based web tool for biomedical literature retrieval*. BMC Bioinformatics. 2014, 15 (Suppl 12): S1.
29. Zhang Y, Cao X, Zhong S: *GeNemo: a search engine for web-based functional genomic data*. Nucleic Acids Research. 2016, 44 (Web Server issue): W122–W127.

## Figures



**Figure 1** PubData user interface. (A) The PubData UI, from which the user can connect to, search, and download files from any number of bioinformatics databases provided by the server list widget. (B) Logged into PANTHER (Protein ANALYSIS Through Evolutionary Relationships) Classification System database.





**Figure 3** PubData recommender system. (A) Collaborative filtering recommender system based on previous search history. (B) Keyword search returns color-coded exact matches of the specific keyword to the specific file path. (C and D) Both exact matches and similar matches are returned by the search engine, allowing the user to pinpoint the full path destination to a desired group of files that meet the user's search criteria.

A

PubData

[File](#) [Action](#) [Search](#) [Help](#)

ftp name : ...

[server list](#)[Connect](#)

Name	Size	Owner	Group	Time

Please select name of a server

[Download](#)

B

PubData

[File](#) [Action](#) [Search](#) [Help](#)

ftp name : ftp.pantherdb.org

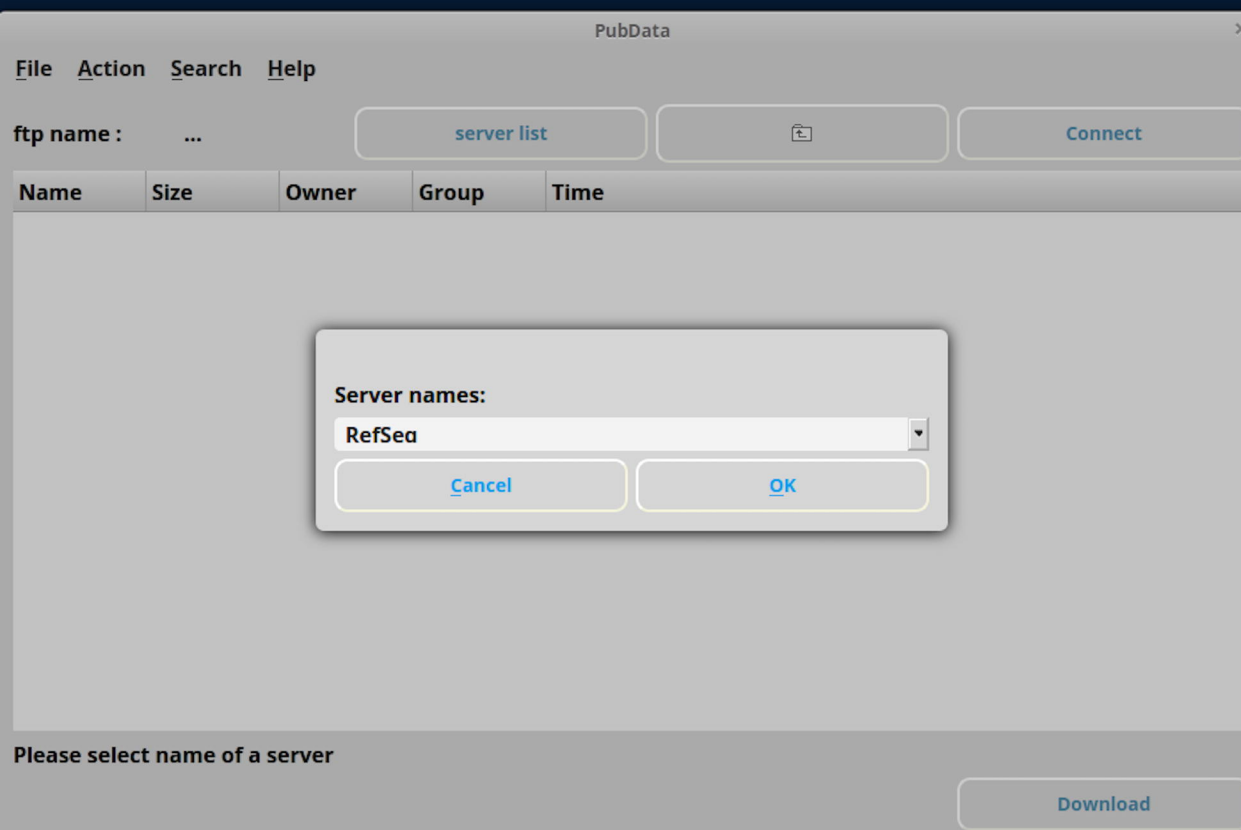
[server list](#)[Disconnect](#)

Name	Size	Owner	Group	Time
CellDe...	4096	503	503	Mar 22 2013
TIPS	4096	503	503	Mar 22 2013
biopax	4096	503	503	Feb 03 2015
cSNP_a...	4096	503	503	Jun 07 2016
downl...	4096	503	503	Mar 22 2013
hmm_c...	4096	503	503	Jul 11 2016
hmm_s...	4096	503	503	Jul 11 2016
ortholog	4096	503	503	Jul 11 2016
panthe...	4096	503	503	Jun 21 2016
panthe...	4096	503	503	Jul 11 2016
pathway	4096	503	503	Jul 11 2016
sequen...	4096	503	503	Jul 11 2016
tmp	4096	503	503	Apr 25 2013

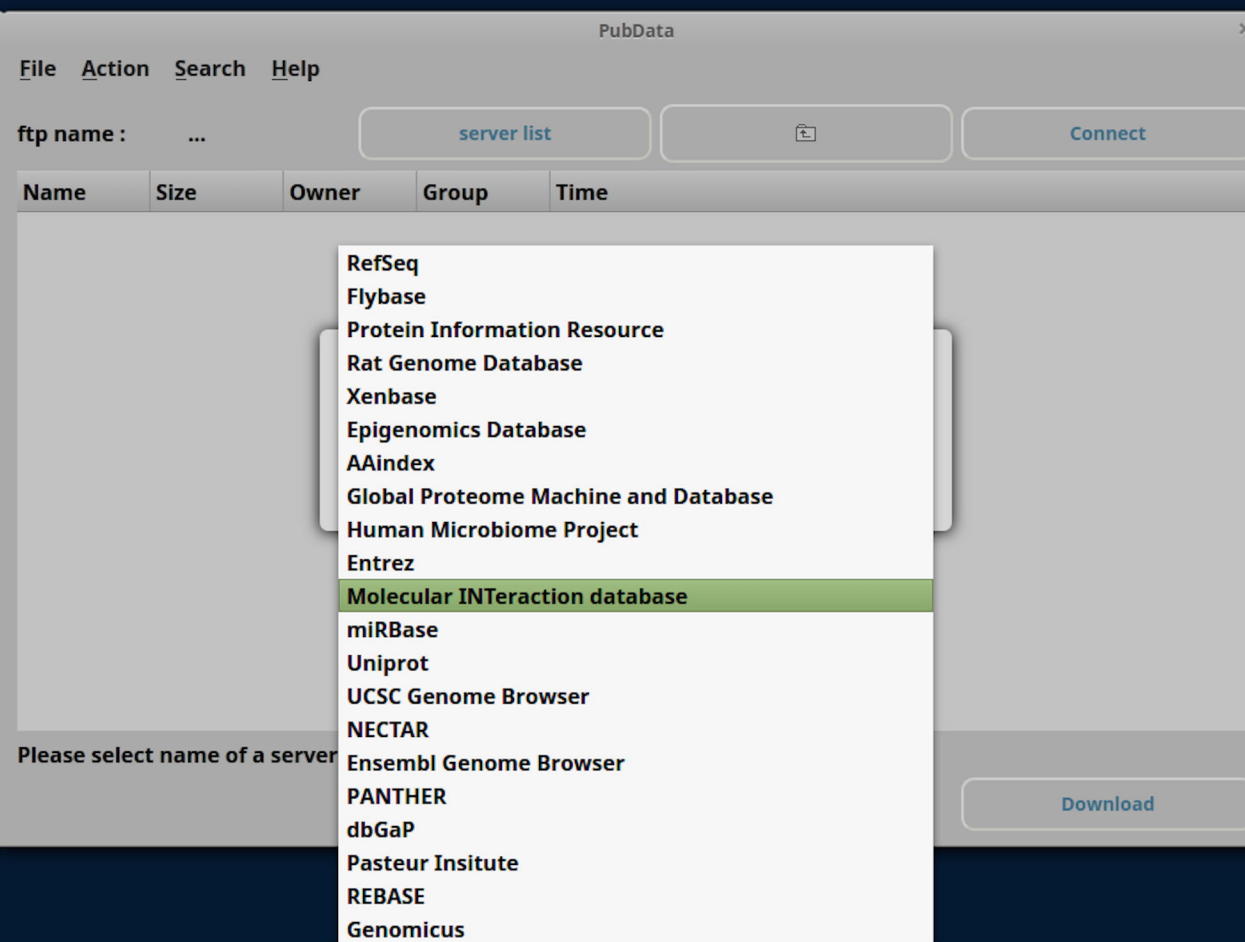
Logged onto ftp.pantherdb.org.

[Download](#)

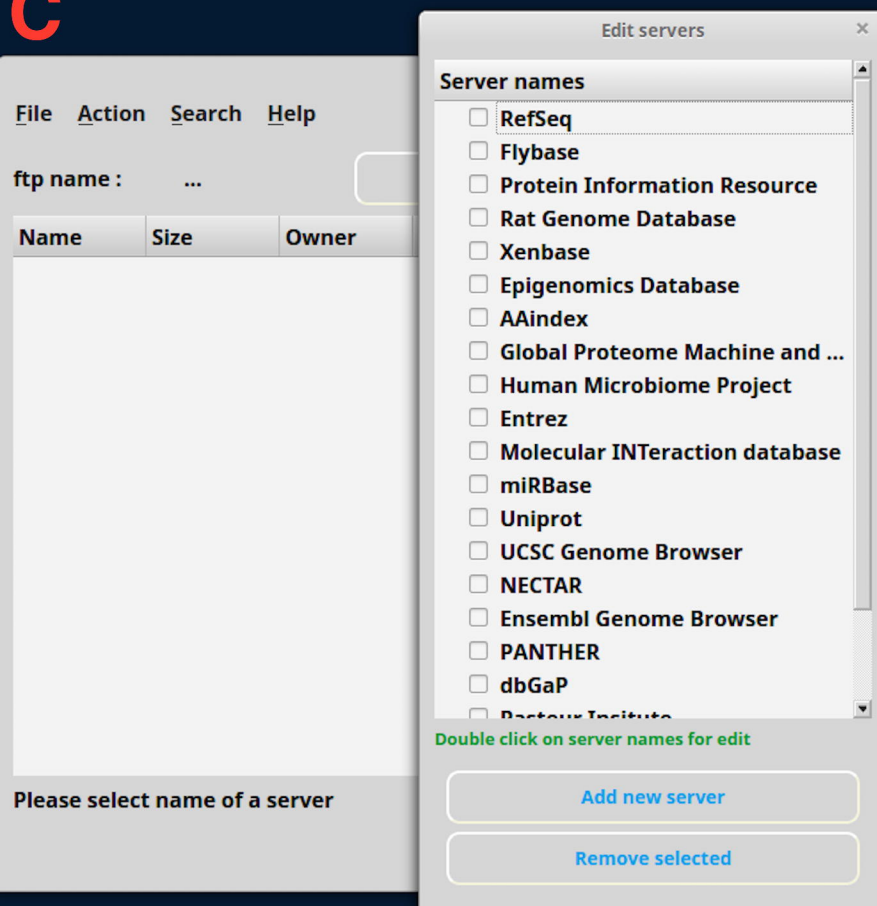
A



B



C



D

