BIOINFORMATICS APPLICATIONS NOTE Vol. 22 no. 20 2006, pages 2572–2573
doi:10.1093/bioinformatics/btl410

Databases and ontologies

IndexToolkit: an open source toolbox to index protein databases for high-throughput proteomics

Dequan Li[1,2,*], Wen Gao[1,2], Charles X. Ling[3], Xiaobiao Wang[1,2], Ruixiang Sun[1] and Simin He[1]

[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China, [2]Graduate University of Chinese Academy of Sciences, Beijing 100039, China and [3]Department of Computer Science, The University of Western Ontario, Canada

Received on March 31, 2006; revised on June 1, 2006; accepted on July 22, 2006
Advance Access publication August 31, 2006
Associate Editor: Satoru Miyano

ABSTRACT

Summary: A software package, IndexToolkit, aimed at overcoming the disadvantage of FASTA-format databases for frequent searching, is developed to utilize an indexing strategy to substantially accelerate sequence queries. IndexToolkit includes user-friendly tools and an Application Programming Interface (API) to facilitate indexing, storage and retrieval of protein sequence databases. As open source, it provides a sequence-retrieval developing framework, which is easily extensible for high-speed-request proteomic applications, such as database searching or modification discovering. We applied IndexToolkit to database searching engine pFind to demonstrate its effect. Experimental studies show that IndexToolkit is able to support significantly faster searches of protein database.

Availability: The IndexToolkit is free to use under the open source GNU GPL license. The source code and the compiled binary can be freely accessed through the website http://pfind.jdl.ac.cn/IndexToolkit. In this website, the more detailed information including screenshots and documentations for users and developers is also available.

Contact: dqli@jdl.ac.cn

1 INTRODUCTION

Protein identification is a critical step in most high-throughput proteomics research. A huge number of tandem mass spectrometry (MS/MS) data needs to be searched in protein databases to identify the protein sequences (Aebersold and Mann, 2003). Currently, most popular protein sequence databases, such as Swiss-Prot and IPI (Kersey *et al*., 2004), are represented in the FASTA format, which is a flat text. However, it is inefficient to perform frequent searches directly on such text-based databases since in each search the whole database has to be scanned from the first entry to the last. In general, when a database is frequently searched but less often updated, as in high-throughput protein identification, indexing can greatly improve the overall search speed. In addition, values needed for search, such as the mass values of peptides and fragment ions, can be pre-calculated and indexed to further improve the search performance.

Recently, a software package named DBToolkit (Martens *et al*., 2005) is made available publicly to convert protein sequence database into peptide sequence database to enhance protein identification. However, the output of DBToolkit is still a text file, and it does not support indexing. As far as we know, there is no open-source package to help experimenters index protein/peptide sequence database [note that the open source search engines X!Tandem (Craig and Beavis, 2004) does not provide general-purpose indexing support]. Although commercial search engines such as Mascot and SEQUEST create indexes for their own utilization, they do not provide indexing support for other researchers to develop their own search engine. Although various new peptide-scoring algorithms are being proposed, the lack in the support of efficient database indexes hampers them from practical application.

To solve this problem, we have developed an open source software called IndexToolkit to perform indexing for a protein sequence database. IndexToolkit can help researchers who develop search engines themselves to improve greatly the efficiency of high-throughput protein identification.

2 DESCRIPTION OF INDEXTOOLKIT

Here we briefly describe the implementation, usage and features of IndexToolkit. IndexToolkit consists of two parts: a set of API (Application Programming Interface) functions for application developers and user-friendly software tools for common users (Fig. 1a). API is the heart of IndexToolkit and is written in the C++ language with Standard Template Library (STL). The IndexToolkit API is open as a C++ class library, including a FASTA format parser, digestion and fragmentation simulator, index creator, index loader, index query and other sequence processing functions. Using the classes, developers can execute various operations interacting with indexes. IndexToolkit also supports memory mapping to read the whole or a part of the index database into memory.

Tools in the IndexToolkit package are implemented to help users to manage the index more effectively, such as importing, setting, indexing and retrieving databases. All tools are developed based on the IndexToolkit API and their sources demonstrate how to use API. Users can realize their own powerful tools in two ways: directly use API or modify the source codes of IndexToolkit tools. As an open project, developers can easily add IndexToolkit into their projects and simultaneously provide a high-speed sequence-retrieval programming framework, without knowing details of indexing technology.

*To whom correspondence should be addressed.

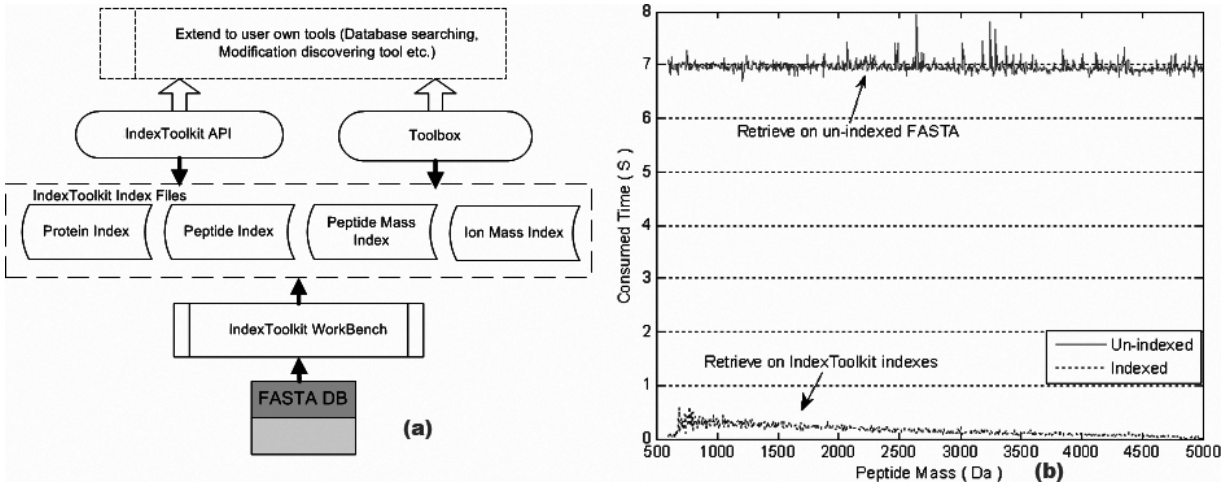

Fig. 1. The overview of IndexToolkit. (**a**) IndexToolkit is a high-throughput sequence-retrieval bridge between proteomics software and FASTA-format database. The Workbench tool creates indexes from databases, and API and other open-source tools are used to access indexes and support to future developing. (**b**) Performance evaluation using IPI Rat database. We search mass value from 600 to 5000 Da with mass tolerance 1 Da and plot the consumed time run on the un-indexed and indexed database.

Using Workbench tool in the package, an input FASTA database is transformed into an index database containing a series of binary indexes (shown in Fig. 1a): two types of entry index (protein entry and peptide entry) and two types of mass index (peptide mass and ion mass). The former two types help to quickly locate the specific position to read fixed segments in the peptide and protein sequence.

The latter two types, typically designed for database searching and modification discovering (Tang *et al*., 2005), index a protein database by two types of sorted masses: the calculated no-modification masses of the peptides obtained from digestion of the proteins and the masses of theoretical no-modification MS/MS fragment ions such as b-ions of each of the peptide. The detailed data structure and generating process of the index are provided on the web pages named 'Format' and 'Q&A'.

Once the protein database has been fully indexed, searching database and scoring an MS/MS spectrum is extremely rapid. To illustrate the improvement, the IPI RAT database, which is ∼21 MB containing 33 379 protein entries, is indexed by IndexToolkit. As we know, a frequent but time-consuming step for database searching is to find candidates whose mass fall within a mass error tolerance of the specified mass value. We evaluate the performance of using IndexToolkit by comparing the time running on un-indexed and indexed database (Fig. 1b). Figure 1b shows that searching with the indexes of IndexToolkit is many times faster than directly searching the raw FASTA database.

3 DISCUSSION

The CPU and memory usage may be high when indexing, but once indexing is completed, the future queries will usually use much less system resources to achieve a much higher performance. Index is a proper data structure to swiftly find all results satisfied specified criteria. However, it is not suitable for all cases. Because the cost to update indexes may be higher than improvements of querying performance, it is not recommended to index if the database changes frequently. In addition, large databases require large amount of disk space for storing their indexes. For IndexToolkit, the peptide-mass indexes may use ∼3 times the disk space as the original database, and the ion-mass indexes may use ∼30 times as the disk space. Nevertheless, we still suggest databases searching software run on the indexed database for high-throughput protein identification, as disk space is very cheap.

In sum, IndexToolkit can improve greatly the speed of protein database search engines, and is recommended for high-throughput protein identification. We have applied IndexToolkit to pFind (Li *et al*., 2005), which is a recently developed search engine with a novel scoring function for peptide identification. The new version of pFind integrated with IndexTookit is about five times faster than the version without indexing. In our future work we will integrate IndexToolkit into other open-source search engines such as X!Tandem, and include more supporting database formats and additional tools for supporting protein identification.

ACKNOWLEDGEMENTS

The authors thank Yan Fu, Yonggang Wei, Leheng Wang and Haipeng Wang for valuable discussions. This work was supported by National '973' Project of China (No. 2002CB713807).

Conflict of Interest: none declared.

REFERENCES

Aebersold,R. and Mann,M. (2003) Mass spectrometry-based proteomics. *Nature*, **422**, 198–207.

Craig,R. and Beavis,R.C. (2004) TANDEM: matching proteins with tandem mass spectra. *Bioinformatics*, **20**, 1466–1467.

Kersey,P.J. *et al*. (2004) The International Protein Index: an integrated database for proteomics experiments. *Proteomics*, **4**, 1985–1988.

Li,D. *et al*. (2005) pFind: a novel database-searching software system for automated peptide and protein identification via tandem mass spectrometry. *Bioinformatics*, **21**, 3049–3050.

Martens,L. *et al*. (2005) DBToolkit: processing protein databases for peptide-centric proteomics. *Bioinformatics*, **21**, 3584–3585.

Tang,W.H. *et al*. (2005) Discovering known and unanticipated protein modifications using MS/MS database searching. *Anal. Chem.*, **77**, 3931–3946.