RAPID COMMUNICATIONS IN MASS SPECTROMETRY
Rapid Commun. Mass Spectrom. 2007; **21**: 2985–2991
Published online in Wiley InterScience (www.interscience.wiley.com) DOI: 10.1002/rcm.3173

pFind 2.0: a software package for peptide and protein identification via tandem mass spectrometry

Le-heng Wang[1,3,4], De-Quan Li[1,3,4], Yan Fu[1,3], Hai-Peng Wang[1,3,4], Jing-Fen Zhang[1], Zuo-Fei Yuan[1,3,4], Rui-Xiang Sun[1,3], Rong Zeng[2], Si-Min He[1,3*] and Wen Gao[1,3*]

[1]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, P.R. China
[2]Key Lab of Proteomics, Institute of Biochemistry and Cell Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Shanghai 200031, P.R. China
[3]Key Lab of Intelligent Information Processing, Chinese Academy of Sciences, Beijing 100080, P.R. China
[4]Graduate University of Chinese Academy of Sciences, Beijing 100039, P.R. China

Received 19 December 2006; Revised 31 May 2007; Accepted 3 July 2007

This paper describes the pFind 2.0 software package for peptide and protein identification via tandem mass spectrometry. Firstly, the most important feature of pFind 2.0 is that it offers a modularized and customized platform for third parties to test and compare their algorithms. The developers can create their own modules following the open application programming interface (API) standards and then add it into workflows in place of the default modules. In addition, to accommodate different requirements, the package provides four automated workflows adopting different algorithm modules, executing processes and result reports. Based on this design, pFind 2.0 provides an automated target-decoy database search strategy: The user can just specify a certain false positive rate (FPR) and start searching. Then the system will return the protein identification results automatically filtered by such an estimated FPR. Secondly, pFind 2.0 is also of high accuracy and high speed. Many pragmatic preprocessing, peptide-scoring, validation, and protein inference algorithms have been incorporated. To speed up the searching process, a toolbox for indexing protein databases is developed for high-throughput applications and all modules are implemented under a new architecture designed for large-scale parallel and distributed searching. An experiment on a public dataset shows that pFind 2.0 can identify more peptides than SEQUEST and Mascot at the 1% FPR. It is also demonstrated that this version of pFind 2.0 has better usability and higher speed than its previous versions. The software and more detailed supplementary information can both be accessed at http://pfind.ict.ac.cn/. Copyright © 2007 John Wiley & Sons, Ltd.

An important problem arising from proteomics research is to automatically identify peptide and protein sequences via tandem mass spectrometry.[1] The database searching approach addresses this problem by assigning the known peptide sequences in databases to the observed tandem mass spectra.[2] The two most widely used commercial softwares employing the database searching approach are SEQUEST[3] and Mascot.[4] SEQUEST uses a cross-correlation scoring function to evaluate the matching between the spectrum and a peptide, while Mascot uses the probability of a match occurring randomly. Some open-source softwares have also been developed, e.g., X!Tandem.[5] Other similar database searching programs include PepFrag,[6] MS-Tag,[7] PEDANTA,[8] SCOPE,[9] Sonar MS/MS,[10] ProbID,[11] PEP_PROBE,[12] Phenyx,[13] VEMS,[14] PepHMM,[15] and DBDigger.[16]

Although many available database searching tools have been developed, there are still many challenges in the reliability, sensitivity and usability. For example, the target-decoy database strategy has been widely adopted for the estimation of false positive rate (FPR) of peptide identification.[17–19] However, this is usually done manually by users and all existing tools lack an automated module to estimate the FPR. Another problem is the speed of searching high-throughput spectra against huge protein databases. The improvements in the sensitivity of mass spectrometers and the rapid expansion of databases have increased the scope and complexity of searching. Traditional software architecture, i.e., running all tasks in a stand-alone process without any data index, is more and more inadequate.

In our earlier work,[20,21] we developed a database searching software system, named pFind, for automated peptide and protein identification using tandem mass spectra. In this paper, we describe the latest version of the system, pFind 2.0,

*Correspondence to: S.-M. He or W. Gao, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, P.R. China.
E-mails: smhe@ict.ac.cn; wgao@ict.ac.cn
Contract/grant sponsor: National Key Basic Research & Development Program (973) of China; contract/grant number: 2002CB713807.
Contract/grant sponsor: CAS Knowledge Innovation Program.
Contract/grant sponsor: National High Technology Research and Development Program (863) of China; contract/grant numbers: 2007AA02Z315 and 2007AA02Z326.

in which several newly developed or improved algorithms, modules and workflows are incorporated. Firstly, two preprocessing algorithms, a protein inference algorithm and a validation algorithm, have been introduced into pFind. The scoring algorithm is also improved. Experimental comparisons on a public dataset among pFind 2.0, SEQUEST and Mascot demonstrate that pFind 2.0 can obtain more true positive identifications than the other two software tools at the same FPR. Secondly, the pFind 2.0 system incorporates the target-decoy database search strategy for automated FPR estimation. Users can specify a required FPR before searching. Then the system will calculate a threshold that achieves the FPR and filter search results automatically. Finally, we developed a toolbox to index protein databases for high-throughput application and designed all modules under a parallel-processing-oriented architecture for distributing the computational load efficiently among a lot of computers. These developments greatly improve the overall searching speed.

SYSTEM ARCHITECTURE DESCRIPTION

There are four main levels in the architecture of the pFind system: platform level, development kit level, algorithm level and search engine level, as depicted in Fig. 1. Each level consists of many modules, i.e., the algorithm level includes modules of preprocessing, database indexing, scoring, validation and protein inference. Every module depends on the functions of the lower-level modules and meanwhile provides services for the higher-level modules.

In the core of pFind 2.0, many new algorithms and development kits have been added and all existing algorithms have been improved. Modules in the algorithm and search engine level are redesigned under a parallel-processing-oriented architecture which also provides the fault-tolerance ability especially when running on inexpensive commodity computer clusters. Additionally, many pragmatic criteria of software engineering have been adopted during the development process. The heart of the system is written in the standard C++ language with Standard Template Library (STL). Each module has unit test script written with the CppUnit testing framework.[22] We describe the modules, architectures, search engine workflows and applications in detail in the following parts.

Preprocessing module

The preprocessing is necessary to filter bad-quality spectral data and alleviate the computational load. pFind 2.0 adopts two preprocessing strategies for different instruments: the default one only keeps the 200 most intensive peaks from each spectrum; and the other is specially designed for high-resolution data such as Q-TOF spectra.[23,24]

The preprocessing algorithm for high-resolution spectra is based on the strategy of classification. Firstly, it uses a

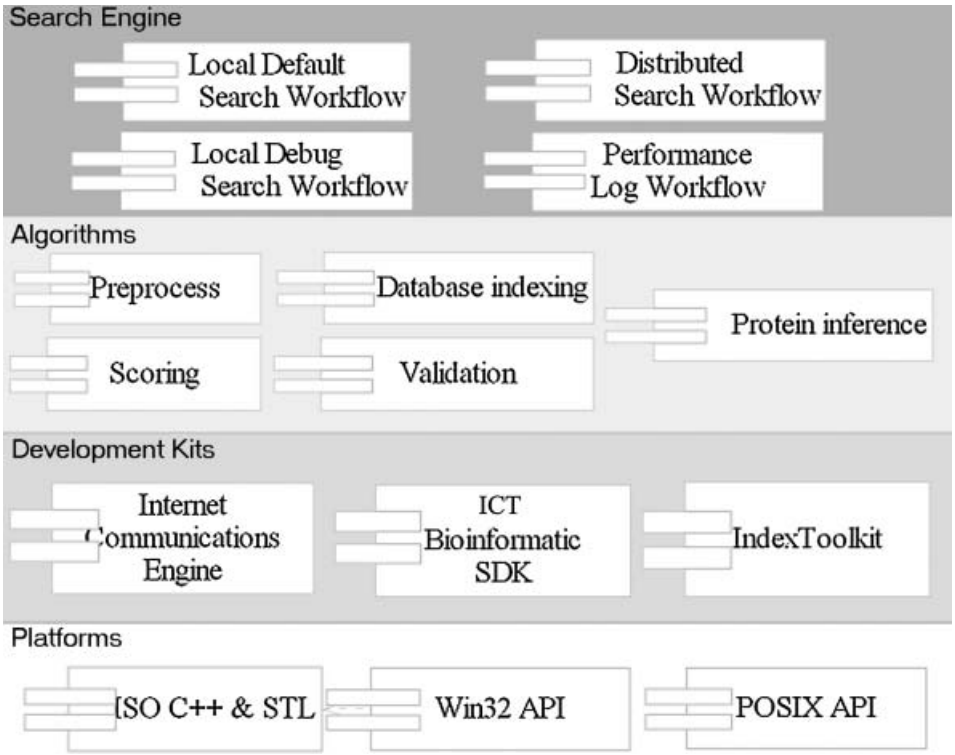

Figure 1. The architecture of pFind 2.0: Win32 is the name of the core set of *Application Programming Interfaces* (APIs) available in the Microsoft Windows operating systems. POSIX (*Portable Operating System Interface for uniX*) is the name of the APIs for software compatible with variants of the Unix operating system. Both Win32 and POSIX are designed for usage by C/C++ programs. ICT Bioinformatics *Software Development Kit* (SDK) is a set of development tools that allows software engineers to create bioinformatics applications.

DOI: 10.1002/rcm

Gaussian mixture model (GMM) to estimate the baseline intensity of noise peaks in spectrum. Secondly, a key concept of an isotope pattern vector (IPV) is introduced to characterize the isotope clusters of fragment ions. Then, the algorithm differentiates spectrum peaks based on some features such as the baseline of noise and IPVs of different isotope clusters. Finally, according to these features, a decision tree is constructed to classify the noise or signal peaks and all potential fragment ions are selected.

Database indexing and searching module

In the database searching identification approach, the most frequently invoked but time-consuming step is to find candidate peptides whose masses match the m/z values in spectra within a mass error tolerance. When a database is searched frequently but updated seldom, indexing can greatly improve the overall searching speed in high-throughput protein identification.

To accelerate the query process, we developed an open source project named IndexToolkit to retrieve candidate peptides.[25] The pFind 2.0 system uses IndexToolkit's application programming interface (API) to access the index files of the masses of peptides obtained from theoretical digestion of proteins in databases. With the well-managed index structure for peptide sequences and their mass values, pFind 2.0 is more powerful to process large-scale spectra than its previous versions. We depict the flow chart of IndexToolkit in Fig. 2.

Scoring module

As a fundamental and indispensable part in the database searching approach to peptide identification, the peptide-scoring algorithm compares the theoretical spectrum of a candidate peptide with an experimental spectrum and calculates a score measuring their similarity.

The pFind system uses the KSDP scoring function,[20] a nonlinear extension to the common spectral dot product. The KSDP scoring function makes use of the correlative information among fragment ions to improve the accuracy of peptide identification. Figure 3(a) illustrates this principle.

In pFind 2.0, the scoring algorithm is improved by introducing a refined mass error model, which provides adaptive error windows for ion-peak matches.[26] By visualizing mass errors in various ways, we found that there is a linear correlation between the mass error and the ion mass, and there is approximate log-log linearity between the standard deviation of mass error and the peak intensity. Based on these observations, we model the mass error of a fragment ion by a conditional normal distribution, whose mean and standard deviation (SD) are the functions of ion mass and peak intensity, respectively. This error model utilizes the fact that the more intense a peak is, the more accurate the mass value should be, as shown in Fig. 3(b). It can considerably improve the accuracy of peptide identification. Currently, the scoring module implements these observations by simply using variable mass tolerance windows for different fragment ions. We will work on to make full use of the new mass error model.

Validation module

Although currently there are many proposed algorithms for peptide identification, most of them either lack an effective validation module or only validate the first-ranked peptide, thus leading to a low identification reliability or sensitivity. Two validation algorithms have been developed for pFind 2.0: the default algorithm using expectation values based on

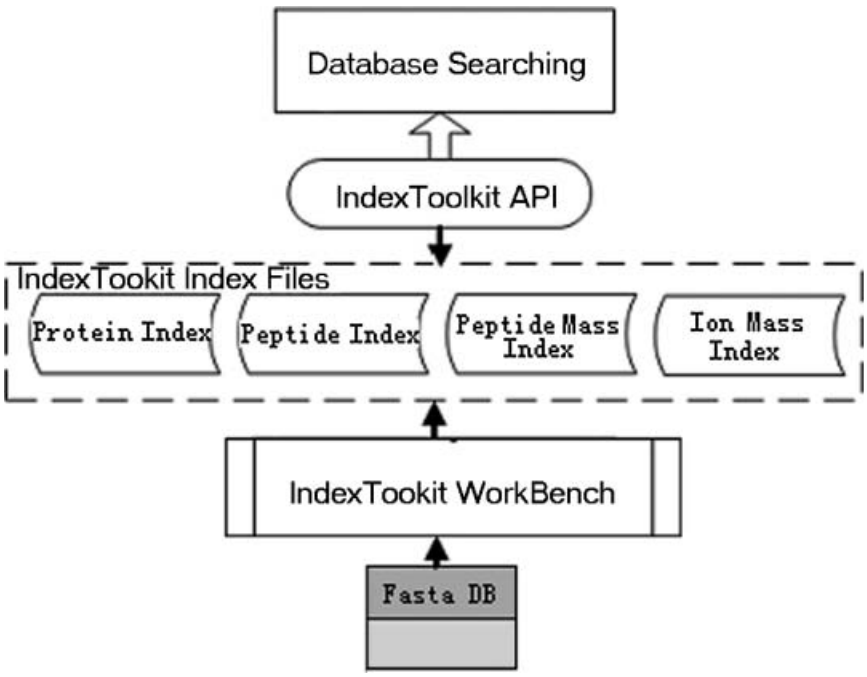

Figure 2. The index toolkit of pFind 2.0: It is a high-throughput sequence-retrieval bridge between proteomics software and FASTA-format database. The workbench tool creates indexes from databases, and APIs are used to access indexes.

DOI: 10.1002/rcm

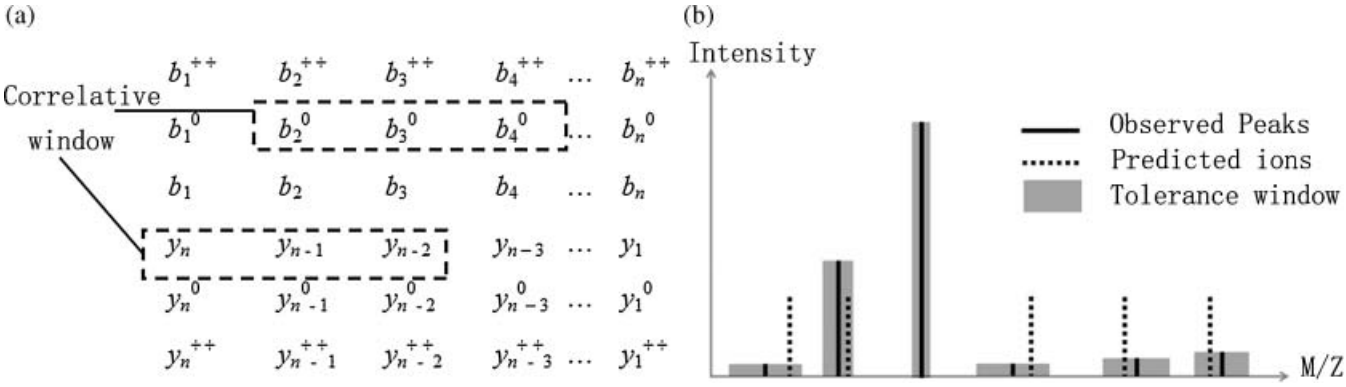

Figure 3. (a) Fragment ion matrix and correlative window. (b) Variable mass tolerance window.

survival functions (E-value),[27] and the novel effective algorithm called pepReap.[28]

pepReap re-ranks peptide candidates generated by the up-stream scoring module using some features characterizing the match quality between peptide and spectrum, such as correlations between ions, the mass matching errors of fragment, peptide ions. Using a support vector machine (SVM), this algorithm can give explicit yes-or-no results. Therefore, it can serve as not only a scoring module, but also as a validation module. The pepReap algorithm currently suits ion trap spectra. We are still making efforts to improve its performance.

Composite target-decoy database search strategy for automated FPR estimation

To estimate more effectively the FPR of peptide-spectral matches, the target-decoy database strategy has been widely used. This strategy is based on the principle that incorrect matches have an equal probability of being derived from either the target or the decoy database.[17–19]

In this strategy, the composite databases contain protein sequences in both forward and reverse orientation. After scoring, the candidate peptides are filtered by some criteria (e.g., E-value[27]). Any surviving peptide derived from the reverse database is defined as a false positive. It can be assumed that the number of false positives is the same for both forward (target) and reverse (decoy) sequences. Therefore, the overall number of false positives can be estimated by doubling the number of peptides found from the reverse sequences, and the FPR can be estimated as FPR $= 2 * \#R/(\#R + \#F)$, where #R is the number of peptides identified from the reverse sequences, and #F is the number of peptides identified from the forward sequences.

Elegant as it is, the target-decoy database strategy has not been incorporated in most software packages; people usually use this strategy manually. pFind 2.0 provides a fully automated method. Firstly, the database index files contain all candidate peptides digested from composite protein sequences in both forward and reverse orientations. After searching, each spectrum matches a candidate peptide best. pFind 2.0 sorts these matching results by their validation scores, calculates the FPRs in turn and finds the validation score threshold that achieves the preset FPR, 1% by default, to filter them. Then, the surviving peptides will be assembled into protein identifications by the protein inference algorithm.

Architecture for distributed and parallel processing

It is time consuming to search thousands of spectra against huge peptide/protein databases in a stand-alone process. One solution is to distribute the computational load efficiently among a lot of computers.

pFind 2.0 has an architecture designed for large-scale parallel and distributed searching. We implement the cluster architecture based on tools of Internet Communications Engine (Ice), an object-oriented middleware platform.[29] Ice provides tools, APIs, and library support for building large-scale parallel and distributed applications.

A pFind cluster consists of a single master node and a lot of slave nodes. The master node assigns search tasks to particular slave nodes and manages a registry service maintaining information, such as the network locations of the slave nodes and the progresses of search tasks, while the slave nodes are responsible for starting and monitoring the tasks assigned to them. If one of the slave nodes encounters problems, the master node will carry the search tasks to the other slave nodes. Considering that the master node consumes little processor time, it also can run on the same computer with a slave node.

Workflows and interfaces

Compared to the previous versions of pFind, the most remarkable progress of pFind 2.0 is its customization and modularization.

To accommodate different requirements, we design different workflows adopting different algorithm modules, executing processes and result reports. The search engine of pFind 2.0 supports four workflows: the local default search workflow in a stand-alone process, the distributed search workflow for parallel cluster as described above, the debug workflow which maintains the runtime information of search engine for developing and testing purposes, and the performance workflow which logs the time information of every module consumed for performance benchmark testing.

pFind 2.0 has become a platform on which third parties can develop their own algorithms modules. For example,

DOI: 10.1002/rcm

Table 1. Some pragmatic tools of pFind 2.0

Name	Function descriptions
pLabel	pLabel can label various ions appearing in the spectra, graphically display matching peaks in different colors between experimental and theoretical spectra, and record this information in a file.
pFormat	pFormat can convert MS/MS files from one format into another (e.g., mzXML to DTA).
pQMass	pQMass can automatically analyze Q-TOF/QSTAR spectra, graphically display noise or signal peaks, show the information of IPVs, and save the preprocessing result.
pBatch	pBatch can load a set of pFind search tasks and execute them in turn.
pBuild	pBuild can analyze the search results, draw the curve of the FPR, find the threshold that achieves the specified FPR, filter the results, group them into proteins and output. It also can graphically display peaks and label matching ions.

following the open API standards, one can build a plug-in implementing a unique preprocessing algorithm and add it into pFind 2.0 workflows; even the executable files of other search engines, e.g., SEQUEST and X!Tandem, can also be used as the scoring module of pFind 2.0 in place of the default KSDP algorithm.

On the other hand, many modules of pFind 2.0 have their own interfaces and applications, which can be incorporated into other software systems. For instance, the input/output module has its application, pFormat, which can convert MS/MS files from one format into the other format; the preprocessing module also can be used independently as the spectra preprocessor of SEQUEST or Mascot; and the database indexing module, an open source project named IndexToolkit,[25] is also applied to other database search tools, such as X!Tandem.

As shown in Table 1, many pragmatic applications have been incorporated into pFind 2.0. All applications of pFind 2.0 provide a compact and user-friendly interface, as shown in Fig. 4, whose input/output data formats are compatible with the previous versions of pFind. The new spectral data standard format, mzXML,[30] is also supported.

PERFORMANCE EVALUATION

The experimental data, that came from a previously reported dataset,[19] were obtained after analyzing five trypsin-digested gel regions of the yeast proteome in triplicate using LTQ and QSTAR (Q-TOF) mass spectrometers. Since the target-decoy database search estimation method described above is instrument- and algorithm-independent, we can compare the performances of different software tools by comparing their FPR estimation curves.

Trypsin and up to two missed cleavage sites are specified for theoretically digesting the proteins in database. The matching tolerance for the precursor and the fragment ion in the LTQ spectra are set to 3 and 1 Da, respectively; while for QSTAR spectra these are set to 0.2 and 0.2, respectively.

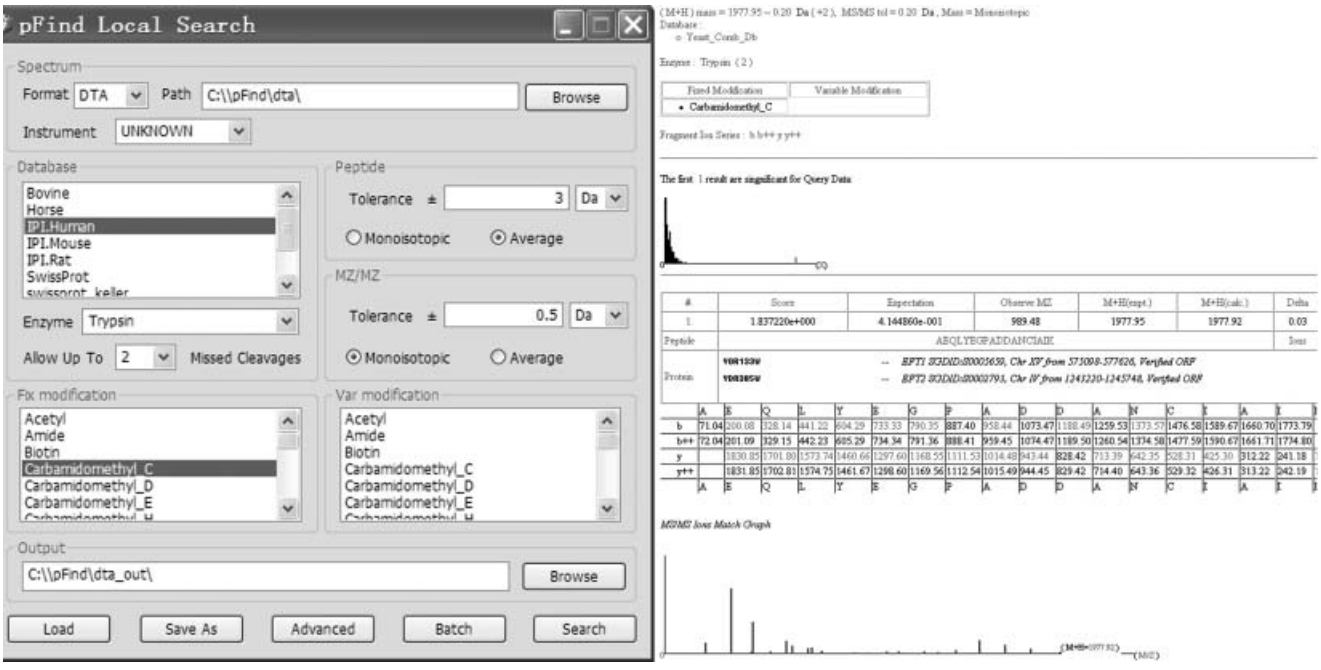

Figure 4. The user interface of pFind 2.0: (a) the local stand-alone process version and (b) the search result of a spectrum using XML format.

DOI: 10.1002/rcm

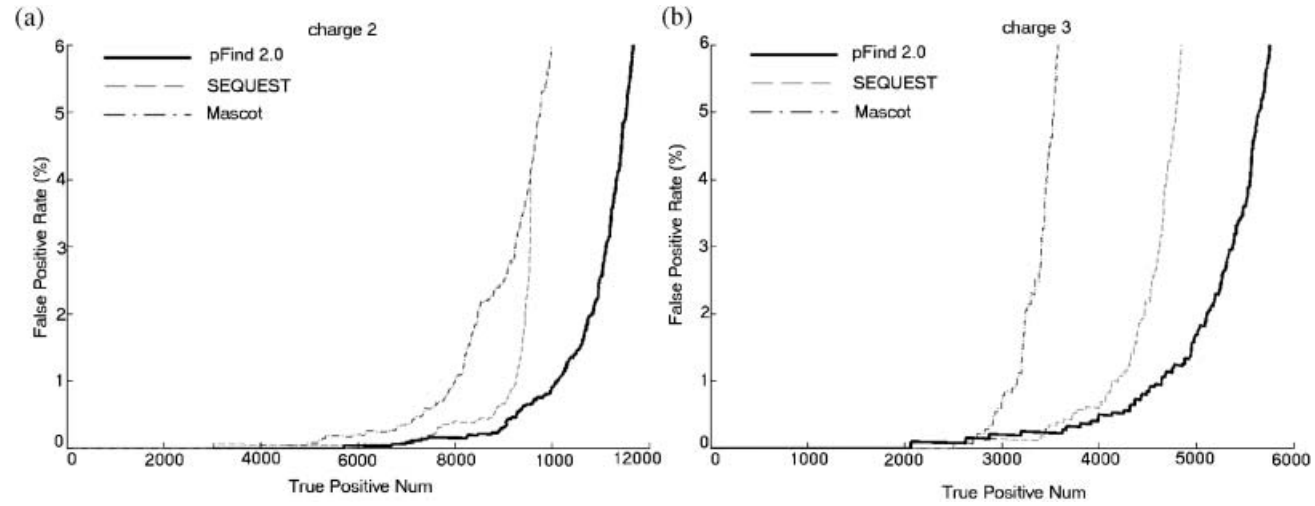

Figure 5. LTQ identification accuracy: (a) +2 charge precursors and (b) +3 charge precursors. The vertical axis stands for false positive rate (FPR) and the horizontal axis stands for true positive identification number (TP).

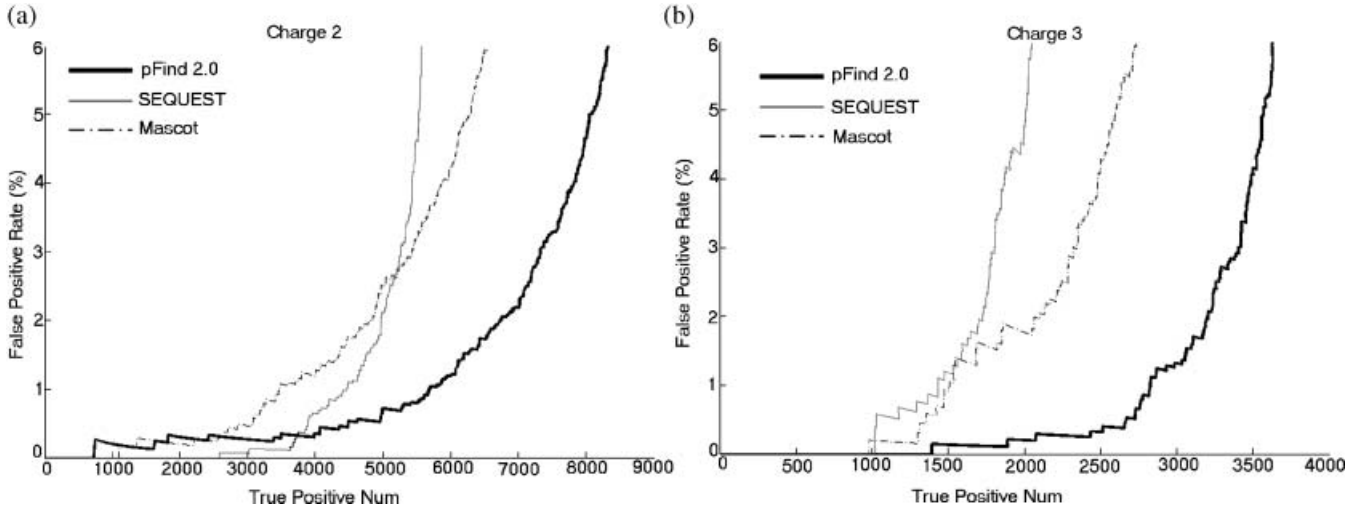

Figure 6. QSTAR identification accuracy: (a) +2 charge precursors and (b) +3 charge precursors. The vertical axis stands for false positive rate (FPR) and the horizontal axis stands for true positive identification number (TP).

Predicted fragment ion types include b, b^{++}, b^0, y, y^{++}, and y^0 (a superscript '++' indicates double charge while single charge is as default. A superscript '0' indicates a neutral loss of H_2O). For LTQ spectra, the default preprocessing algorithm, the KSDP scoring algorithm with mass tolerance model and the default validation algorithm using expectation values are adopted, while for QSTAR spectra the high-resolution preprocessing algorithm, the KSDP scoring algorithm and the default validation algorithm are chosen. In addition, more supplemental information, like the detailed searching parameters of Mascot 2.1.03, SEQUEST 2.7 and pFind 2.0, can be found on the webpage.[31]

Figures 5 and 6 show the searching performance of pFind 2.0 compared to SEQUEST and Mascot, from which it can be observed that pFind 2.0 can achieve a higher true positive identification number than other two softwares at nearly 1% FPR. It demonstrates that pFind 2.0 achieves higher performance in terms of identification accuracy on both LTQ and QSTAR spectra.

Figure 7 illustrates the running time of pFind 2.0 compared to pFind 1.0. It demonstrates that the speed of the search engine in pFind 2.0 is improved markedly.

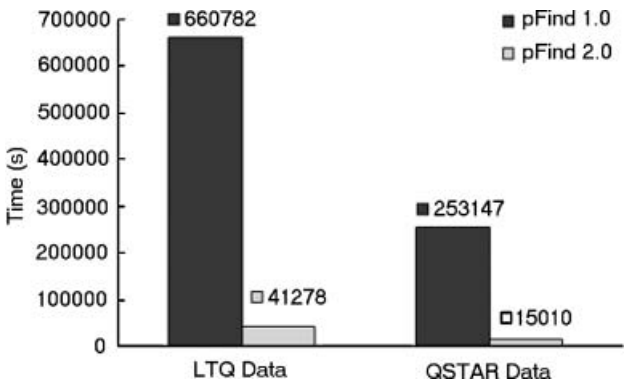

Figure 7. Running time of pFind 1.0/2.0.

DOI: 10.1002/rcm

CONCLUSIONS

This paper describes our recent work on pFind 2.0, a software system for peptide and protein identification via tandem mass spectra. Firstly, many new modules have been designed and most existing algorithms are revised and refined in pFind 2.0. Secondly, an *automatic* estimation of FPR is provided, based on the target-decoy database search strategy. Finally, a toolbox is incorporated to index protein databases for high-throughput application. In addition, the system has been implemented under a new architecture for large-scale parallel and distributed database searching, which provides fault-tolerance ability when running on an inexpensive commodity cluster.

As a result, pFind 2.0 has become a platform on which third parties can develop their own algorithms modules. Four different workflows of search engine and applications can be chosen for different search requirements. The speed of the database search engine for peptides and proteins in pFind 2.0 is improved greatly compared with the previous versions. Furthermore, experiments show that pFind 2.0 has better identification accuracy than previous versions of pFind and other systems, e.g., SEQUEST and Mascot.

Our future work will focus on improving the performance of algorithms and workflows further. More mass spectrometer types, input/output data formats and operating systems will be supported. The local searching version of pFind 2.0 is released through the website[31] and more detailed information can be found there.

Acknowledgements

This work was supported by the National Key Basic Research & Development Program (973) of China under Grant No. 2002CB713807, CAS Knowledge Innovation Program, and the National High Technology Research and Development Program (863) of China under Grant Nos. 2007AA02Z315 and 2007AA02Z326. We thank Cui Zou, Hao Chi, You Li, JinJin Cai, Xiaobiao Wang, Jun Miao, Bo Cao, Yonggang Wei and Yi Zhao for valuable discussions.

REFERENCES

1. Aebersold R, Mann M. *Nature* 2003; **422**: 198.
2. Xu C, Ma B. *Drug Discov. Today* 2006; **11**: 595.
3. Eng JK, McCormack AL, Yates JR III. *J. Am. Soc. Mass. Spectrom.* 1994; **5**: 976.
4. Perkins DN, Pappin DJ, Creasy DM, Cottrell JS. *Electrophoresis* 1999; **20**: 3551.
5. Craig R, Beavis RC. *Bioinformatics* 2004; **20**: 1466.
6. Fenyo D, Qin J, Chait BT. *Electrophoresis* 1998; **19**: 998.
7. Clauser KR, Baker P, Burlingame AL. *Anal. Chem.* 1999; **71**: 2871.
8. Pevzner PA, Dancik V, Tang CL. *J. Comput. Biol.* 2000; **7**: 777.
9. Bafna V, Edwards N. *Bioinformatics* 2001; **17** (Suppl 1): S13.
10. Field HI, Fenyo D, Beavis RC. *Proteomics* 2002; **2**: 36.
11. Zhang N, Aebersold R, Schwikowski B. *Proteomics* 2002; **2**: 1406.
12. Sadygov RG, Yates JR III. *Anal. Chem.* 2003; **75**: 3792.
13. Colinge J, Masselot A, Giron M, Dessingy T, Magnin J. *Proteomics* 2003; **3**: 1454.
14. Matthiesen R, Bunkenborg J, Stensballe A, Jensen ON, Welinder KG, Bauw G. *Proteomics* 2004; **4**: 2583.
15. Wan Y, Yang A, Chen T. *Anal. Chem.* 2006; **78**: 432.
16. Tabb DL, Narasimhan C, Strader MB, Hettich RL. *Anal. Chem.* 2005; **77**: 2464.
17. Peng J, Elias JE, Thoreen CC, Licklider LJ, Gygi SP. *J. Proteome Res.* 2003; **2**: 43.
18. MacCoss MJ. *Curr. Opin. Chem. Biol.* 2005; **9**: 88.
19. Elias JE, Haas W, Faherty BK, Gygi SP. *Nat. Methods* 2005; **2**: 667.
20. Fu Y, Yang Q, Sun R, Li D, Zeng R, Ling CX, Gao W. *Bioinformatics* 2004; **20**: 1948.
21. Li D, Fu Y, Sun R, Ling CX, Wei Y, Zhou H, Zeng R, Yang Q, He S, Gao W. *Bioinformatics* 2005; **21**: 3049.
22. Available: http://cppunit.sourceforge.net/cppunit-wiki/FrontPage.
23. Zhang J, He S, Cai J, Cao X, Sun R, Fu Y, Zeng R, Gao W. *Genomics, Proteomics Bioinformatics* 2005; **3**: 231.
24. Zhang J, Gao W, Cai J, He S, Zeng R, Chen R. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 2005; **2**: 217.
25. Li D, Gao W, Ling CX, Wang X, Sun R, He S. *Bioinformatics* 2006; **22**: 2572.
26. Fu Y, Gao W, He SM, Sun RX, Zhou H, Zeng R. *Pacific Symp. Biocomputing* 2007; **12**: 421.
27. Fenyö D, Beavis RC. *Anal. Chem.* 2003; **75**: 768.
28. Wang HP, Fu Y, Sun RX, He SM, Zeng R, Gao W. *Pacific Symp. Biocomputing* 2006; **11**: 303.
29. Henning M. *IEEE Internet Computing* 2004; **8**: 66.
30. Pedrioli PG, Eng JK, Hubley R, Vogelzang M, Deutsch EW, Raught B, Pratt B, Nilsson E, Angeletti RH, Apweiler R, Cheung K, Costello CE, Hermjakob H, Huang S, Julian RK, Kapp E, McComb ME, Oliver SG, Omenn G, Paton NW, Simpson R, Smith R, Taylor CF, Zhu W, Aebersold R. *Nat. Biotechnol.* 2004; **22**: 1459.
31. Available: http://pfind.ict.ac.cn/pfind/pFind2.htm.

DOI: 10.1002/rcm