Bioinformatics, 31(24), 2015, 4026–4028
doi: 10.1093/bioinformatics/btv490
Advance Access Publication Date: 26 August 2015
Applications Note

Gene expression

HiXCorr: a portable high-speed X_{Corr} engine for high-resolution tandem mass spectrometry

Hyunwoo Kim[1], Hosung Jo[1], Heejin Park[2,*] and Eunok Paek[2]

[1]Department of Electronics and Computer Engineering and [2]Department of Computer Science and Engineering, Hanyang University, Seoul, Korea

*To whom correspondence should be addressed.

Associate Editor: Ziv Bar-Joseph

Received on March 26, 2015; revised on August 12, 2015; accepted on August 14, 2015

Abstract

Summary: Peptide identification is an important problem in proteomics. One of the most popular scoring schemes for peptide identification is X_{Corr} (cross-correlation). Since calculating X_{Corr} is computationally intensive, a lot of efforts have been made to develop fast X_{Corr} engines. However, the existing X_{Corr} engines are not suitable for high-resolution MS/MS spectrometry because they are either slow or require a specific type of CPU. We present a portable high-speed X_{Corr} engine for high-resolution tandem mass spectrometry by developing a novel algorithm for calculating X_{Corr}. The algorithm enables X_{Corr} calculation 1.25–49 times faster than previous algorithms for 0.01 Da fragment tolerance. Furthermore, our engine is easily portable to any machine with different types of CPU because it is developed in C language. Hence, our X_{Corr} engine will expedite peptide identification by high-resolution tandem mass spectrometry.

Availability and implementation: Available at http://isa.hanyang.ac.kr/HiXCorr/HiXCorr.html.

Contact: hjpark@hanyang.ac.kr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Proteomics (Wilkins *et al.*, 1997) is the study of proteins, particularly expression, structures, functions and interactions of proteins. Because proteins play important roles in a human body, correct protein (sequence) identification (Steen *et al.*, 2004) is very important. High-throughput protein identification is generally done by cleaving a protein into peptides, getting tandem mass (MS/MS) spectra of the peptides and analyzing the spectra to identify peptide sequences.

SEQUEST (Eng *et al.*, 1994) is one of the most widely used computer programs for peptide identification from MS/MS spectrum analysis. It compares an experimental spectrum with theoretical spectra computationally created from sequences in peptide database, and finds the theoretical spectrum most similar to the experimental spectrum. To measure the similarity between the theoretical and experimental spectra, SEQUEST uses a sophisticated scoring scheme X_{Corr} (cross-correlation).

However, calculating X_{Corr} can be very slow and consumes most of the running time of SEQUEST. Thus, a lot of efforts have been made to overcome this speed issue. The original SEQUEST used fast Fourier transform algorithm (Cormen *et al.*, 2001) to make the X_{Corr} calculation faster. Later, Crux (Eng *et al.*, 2008) improved the calculation speed of X_{Corr} by using a precomputation table, which is also used in modern SEQUEST and TurboSEQUEST. Faster X_{Corr} calculation is performed by Tide (Diament and Noble, 2011). It was optimized for x86 machine by including the x86 assembly code. Later, a portable Tide was developed in C language with exact *P*-value computation capability. (Hobert and Noble, 2014). To distinguish these two Tide versions, we will call the earlier version with x86 assembly code *Tide-x86* and the later portable version *Tide-C*. Modern processors have multicores and support multithreading. Comet (Eng *et al.*, 2013), an open-source MS/MS search tool by X_{Corr}, supported multithreading for X_{Corr} calculation. Thus, the more processors and cores a machine has, the faster the Comet runs.

Nowadays, more and more spectra are being acquired by high-resolution mass spectrometers. For example, Q-Exactive Orbitrap hybrid mass spectrometers (Thermo Scientific, Bremen, Germany)

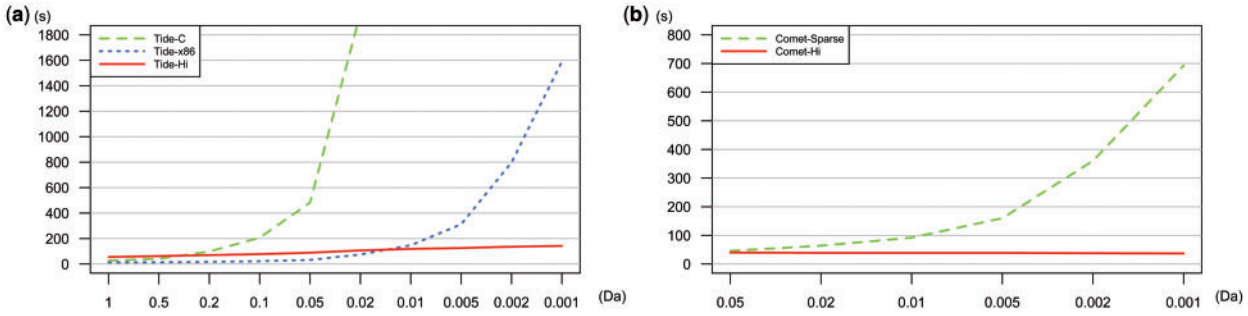

Fig. 1. (**a**) Compares the total running times of Tide-C, Tide-x86 and Tide-Hi and (**b**) compares the total running times of Comet-Sparse and Comet-Hi. The MS/MS data were generated by the Clinical Proteomic Tumor Analysis Consortium (NCI/NIH) and are explained in detail in the Supplementary Data

generate massive MS/MS high-resolution spectra whose fragment ion mass accuracy is within 0.01 Da. In addition, ultra high-resolution spectra whose fragment ion mass accuracy is <0.01 Da are expected to be generated in the near future. For high-resolution MS/MS spectra, calculating X_{Corr} becomes much slower and consumes most of the running time of peptide identification program. For example, the X_{Corr} engines in Tide-x86 and Tide-C run 6.6 and 20 times slower, respectively, when the fragment tolerance is 0.01 Da than when the tolerance of 0.1 Da (Fig. 1a and Supplementary Table S1). Comet shows similar behavior as the resolution gets higher (Fig. 1b, Supplementary Table S2, and Supplementary Fig. S1).

The existing X_{Corr} engines run slower for high-resolution spectra because they require more memory as the resolution gets higher: They create an O (m/f)-sized mass bin array for X_{Corr} calculation where m is the precursor mass and f is the fragment ion mass accuracy. For example, for a low-resolution spectrum whose precursor mass is 1000 Da and fragment tolerance is 1 Da, they create an array whose size is around 1000. However, for a high-resolution spectrum whose precursor mass is 1000 Da and fragment tolerance is 0.01 Da, they create an array whose size is around 100 000. Comet suggested a partial solution for this. When it runs with "use_sparse_matrix=1" in the parameter file, it first creates a huge mass bin array and then compresses the array. We will call this *Comet-Sparse*.

2 Results

In this article, we present a portable hi-speed X_{Corr} engine, which does not create a mass bin array altogether, instead, calculates X_{Corr} directly from the peak list. Thus, it runs in $O(p)$ time where p is the number of peaks in a spectrum, while all the previous engines are based on X_{Corr} algorithms running in $O(m/f)$ time where m is the precursor mass and f is the fragment tolerance (pseudocodes are available in the Supplementary Data).

We compared our X_{Corr} engine with previous engines on a machine with an Intel Core i7-3770K CPU (3.50 GHz) and 32 GB RAM under the CentOS 6.6 operating system and the GNU C compiler 4.4.7. First, we implanted our X_{Corr} engine into Tide-C and named it Tide-Hi. We compared Tide-Hi, with Tide-C, and Tide-x86. Since Tide-x86 does not calculate the exact P-value, we compared them without exact P-value calculation. Figure 1a and Supplementary Table S1 show that Tide-Hi is 49 times faster than Tide-C in X_{Corr} calculation and 45 times faster in total running time when the fragment tolerance is 0.01 Da. The running time gap between Tide-Hi and Tide-C gets bigger as the resolution gets higher. Tide-Hi is even 1.25 times faster than Tide-x86 in both X_{Corr} calculation and total running time for 0.01 Da fragment tolerance. (Note that Tide-Hi is developed in C language and Tide-x86 includes x86 assembly code.) Second, we implanted our X_{Corr} engine into Comet-Sparse and named it Comet-Hi. (Comet without sparse option requires much more memory to run on high-resolution data.) Figure 1b and Supplementary Table S2 show that Comet-Hi runs 2.4 times faster than Comet-Sparse for 0.01 Da fragment tolerance when eight threads were enabled. The gap between Comet-Hi and Comet-Sparse also gets bigger as the resolution gets higher when eight threads were used. Supplementary Figure S1 shows similar patterns for one, two and four threads.

3 Conclusion

We present a portable high-speed X_{Corr} engine for high-resolution tandem mass spectrometry by developing a novel algorithm, which enables X_{Corr} calculation 1.25–49 times faster than before for 0.01 Da fragment tolerance. When the fragment tolerance is 0.001 Da, our engine runs 1000 times faster than Tide-C's X_{Corr} engine, 20 times faster than Comet-Sparse's and 11 times faster than Tide-x86's X_{Corr} engine (Fig. 1 and Supplementary Data). Furthermore, our engine is easily portable to almost every machine because it is developed in C. Optimizing our engine for x86 machines by embedding an x86 machine code can be a future research topic. Since X_{Corr} score is widely used in peptide identification, this article may be useful for the community. Finally, we did not trade correctness for efficiency. Our X_{Corr} engine calculates the same X_{Corr} score as Tide and Comet do (Supplementary Theorem 2).

Funding

This work was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2012-0006999) and also by the National Research Foundation of Korea [NRF-2012M3A9B9036676, NRF-2014R1A2A1A11054147, NRF-2012M3A9D1054452].

Conflict of Interest: none declared.

References

Cormen,T.H. *et al.* (2001) *Introduction to Algorithms*, 2nd edn. MIT Press, Cambridge, MA.

Diament,B.J. and Noble,W.S. (2011) Faster SEQUEST searching for peptide identification from tandem mass spectra. *J. Proteome Res.*, **10**, 3871–3879.

Eng,J.K. *et al*. (1994) An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. *J. Am. Soc. Mass Spectrom.*, **5**, 976–989.

Eng,J.K. *et al*. (2008) A fast SEQUEST cross correlation algorithm. *J. Proteome Res.*, **7**, 4598–4602.

Eng,J.K. *et al*. (2013) Comet: an open-source MS/MS sequence database search tool. *J. Proteomics*, **13**, 22–24.

Hobert,J.J. and Noble,W.S. (2014) Computing exact p-values for a cross-correlation shotgun proteomics score function. *J. Mol. Cell. Proteomics*, **13**, 2467–2479.

Steen,H. *et al*. (2004) The ABC's (XYZ's) of peptide sequencing. *Nat. Rev. Mol. Cell Biol.*, **5**, 699–711.

Wilkins,M.R. *et al*. (1997) *Proteome Research: New Frontiers in Functional Genomics*, 1st ed. Springer, New York.