BMC Bioinformatics

SOFTWARE

Open Access

Progressive search in tandem mass spectrometry

Yoonsung Joh[1], Kangbae Lee[1], Hyunwoo Kim[2] and Heejin Park[1*]

*Correspondence:
hjpark@hanyang.ac.kr

[1] Department of Computer Science, Hanyang University, Seoul 06978, Republic of Korea
[2] Biomedical Informatics Team, Korea Institute of Science and Technology Information, Daejeon 34141, Republic of Korea

Abstract

Background: High-throughput Proteomics has been accelerated by (tandem) mass spectrometry. However, the slow speed of mass spectra analysis prevents the analysis results from being up-to-date. Tandem mass spectrometry database search requires $O(|S||D|)$ time where S is the set of spectra and D is the set of peptides in a database. With usual values of $|S|$ and $|D|$, database search is quite time consuming. Meanwhile, the database for search is usually updated every month, with 0.5–2% changes. Although the change in the database is usually very small, it may cause extensive changes in the overall analysis results because individual PSM scores such as deltaCn and E-value depend on the entire search results. Therefore, to keep the search results up-to-date, one needs to perform database search from scratch every time the database is updated, which is very inefficient.

Results: Thus, we present a very efficient method to keep the search results up-to-date where the results are the same as those achieved by the normal search from scratch. This method, called progressive search, runs in $O(|S||\Delta D|)$ time on average where ΔD is the difference between the old and the new databases. The experimental results show that the progressive search is up to 53.9 times faster for PSM update only and up to 16.5 times faster for both PSM and E-value update.

Conclusions: Progressive search is a novel approach to efficiently obtain analysis results for updated database in tandem mass spectrometry. Compared to performing a normal search from scratch, progressive search achieves the same results much faster. Progressive search is freely available at: https://isa.hanyang.ac.kr/ProgSearch.html.

Keywords: Mass spectrometry, Protein sequence analysis, Proteomics, Algorithms

Background

Database search in tandem mass spectrometry, usually done by as Sequest [1], Tide [2], Comet [3], Mascot [4], Maxquant [5], MS-GF [6], MSFragger [7], and so on, is quite time consuming: Especially when the number of spectra is large, for example, more than 10 million of spectra [8, 9] and/or the search space is wide such as open search [7].

Meanwhile, protein databases used for search are updated frequently. For example, the most widely used database, Uniprot [10], is updated monthly with 0.5 to 2% changes, which means newly identified protein sequences are inserted and some incorrect

sequences are deleted. Although the change is very small, it may cause changes in the overall analysis results because each spectrum score is calculated relatively based on the entire search results. Therefore, to keep the search results up-to-date, one needs to perform database search from scratch every time the database is updated, which is very inefficient.

Thus, we present a very efficient method to keep the search results up-to-date where the results are the same as those achieved by the normal search from scratch. This method, called progressive search, efficiently minimizes the computation time such that our progressive search is much faster than the normal search from scratch. In this study, we applied our progressive search to Comet which is not only incorporated into widely used proteomics pipelines such as Trans-Proteomics Pipeline [11] and Crux [12] but also a stand-alone open source tandem mass spectrometry database search engine. Our experimental results in Figs. 1 and 2 show that progressive search is 16.5–53.9 times faster than the normal search where the database change is 0.16%, the number of tryptic termini is 1, and the number of missed cleavage is 2.

Implementation

Database separation

First, we compare the old database D_{old} and the new database D_{new} to identify D_{srd}, D_{del}, and D_{ins} where D_{srd} contains the proteins shared by both D_{old} and D_{new}, D_{del} contains the proteins stored in only D_{old}, and D_{ins} contains the proteins stored in only D_{new}. Let R_{old}, R_{new}, and R_{srd} denote the PSM results for D_{old}, D_{new}, and D_{srd}, respectively. Figure 3 shows the case that D_{old} is the set of proteins {A, B, C, D, E} and D_{new} is the set of proteins {B, C, D, E, F}. Thus, D_{srd} is the set of proteins {B, C, D, E}, D_{del} is {A}, and D_{ins} is {F}.

For experimental results, we used UniProtKB database released from January to June 2020 (Fig. 4). On average, 0.07% and 0.67% of the proteins were deleted and inserted every month, respectively. In addition, 0.09% and 0.70% of the amino acids are deleted and inserted every month on average, respectively.

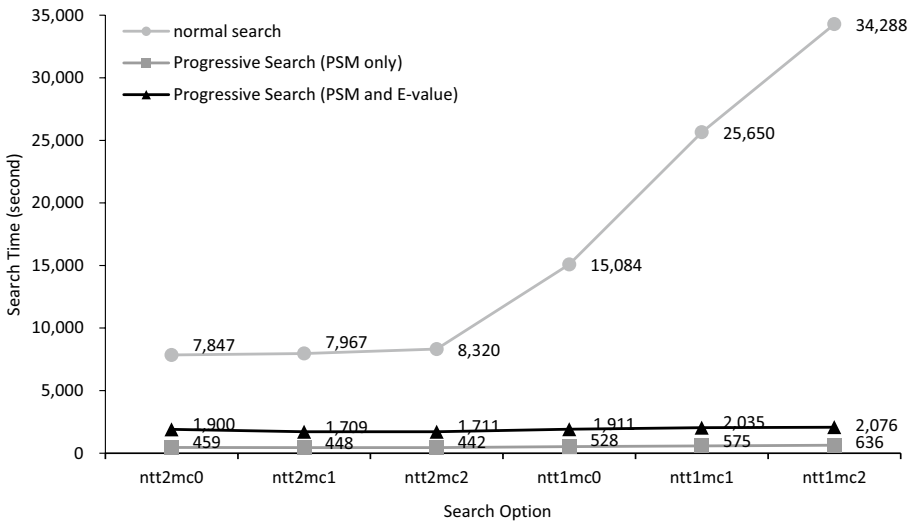


Fig. 1 Search time comparison between the normal search from scratch and the progressive search

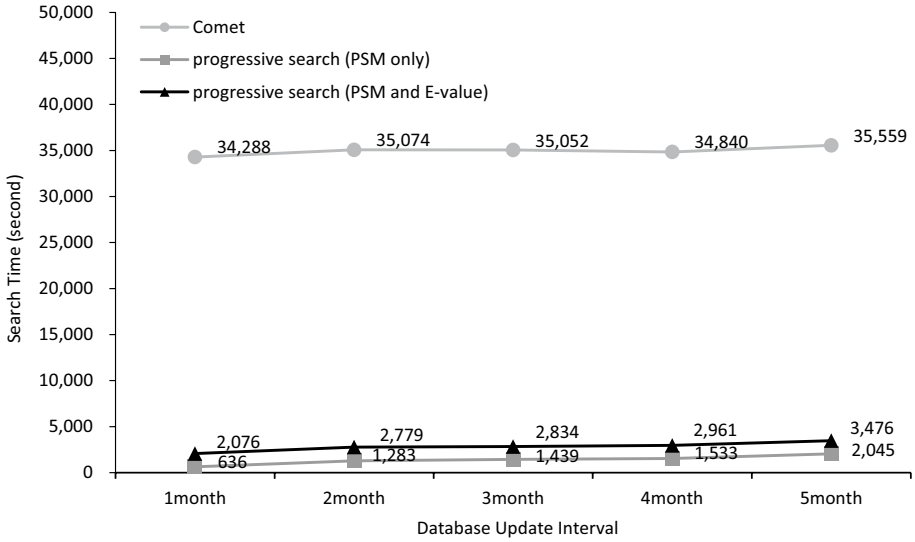


Fig. 2 Search time comparison according to database update intervals for ntt1mc2

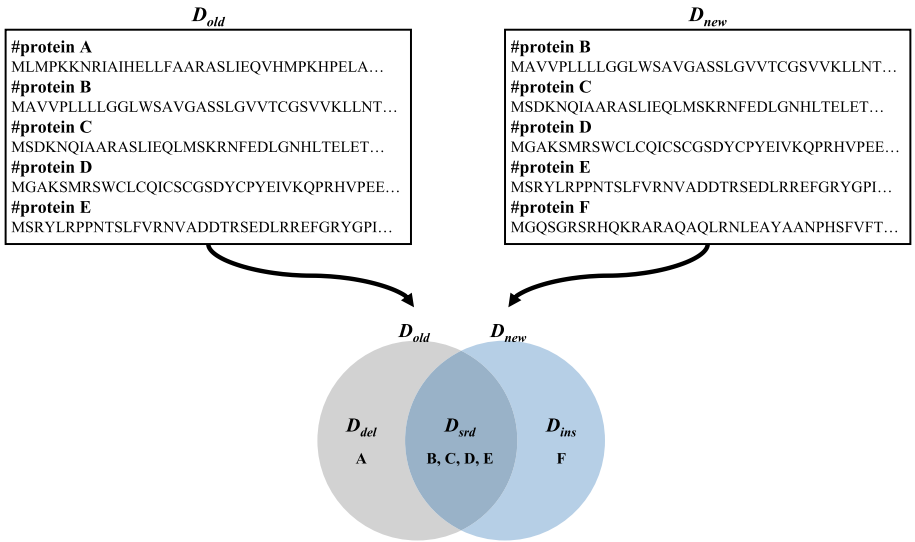


Fig. 3 Database comparison example

Workflow

Progressive search consists of four steps called "deletion", "insertion", "score calculation", and "E-value calculation" (Fig. 5). We explain the progressive search that runs in $O(|S||\Delta D|)$ time on average where S is the set of spectra and ΔD is the difference between the old and the new databases where $|X|$ denotes the number elements in X.

(1) *Deletion* This is the process of obtaining R_{srd} from R_{old}. The R_{srd} is the same as R_{old} except the PSMs whose peptide sequences are from only D_{del}. Those PSMs are deleted and replaced by PSMs obtained by searching D_{srd} for the spectra in the deleted PSMs (S_{del}). For example, PSMs of scans 1, 3 and 6 are updated after deletion in Fig. 5.

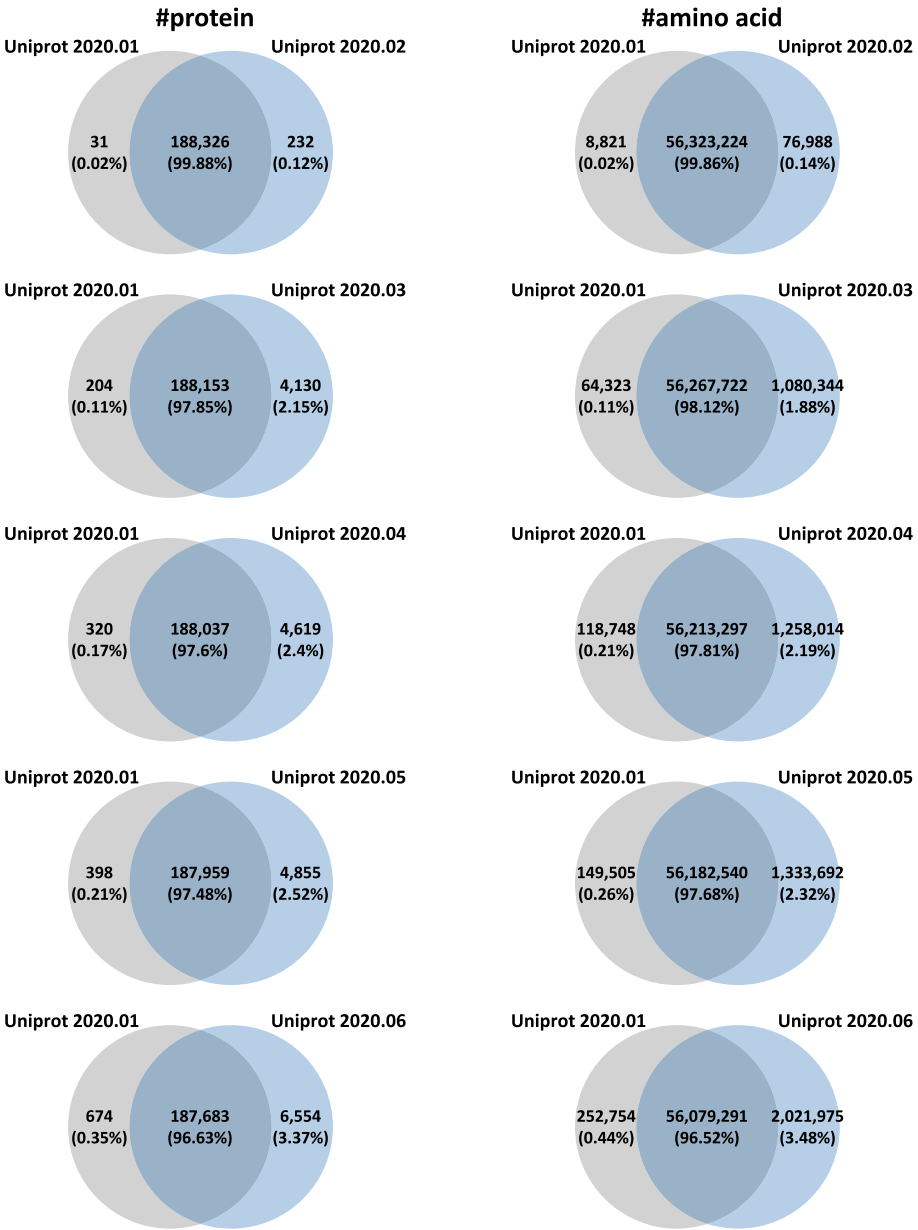
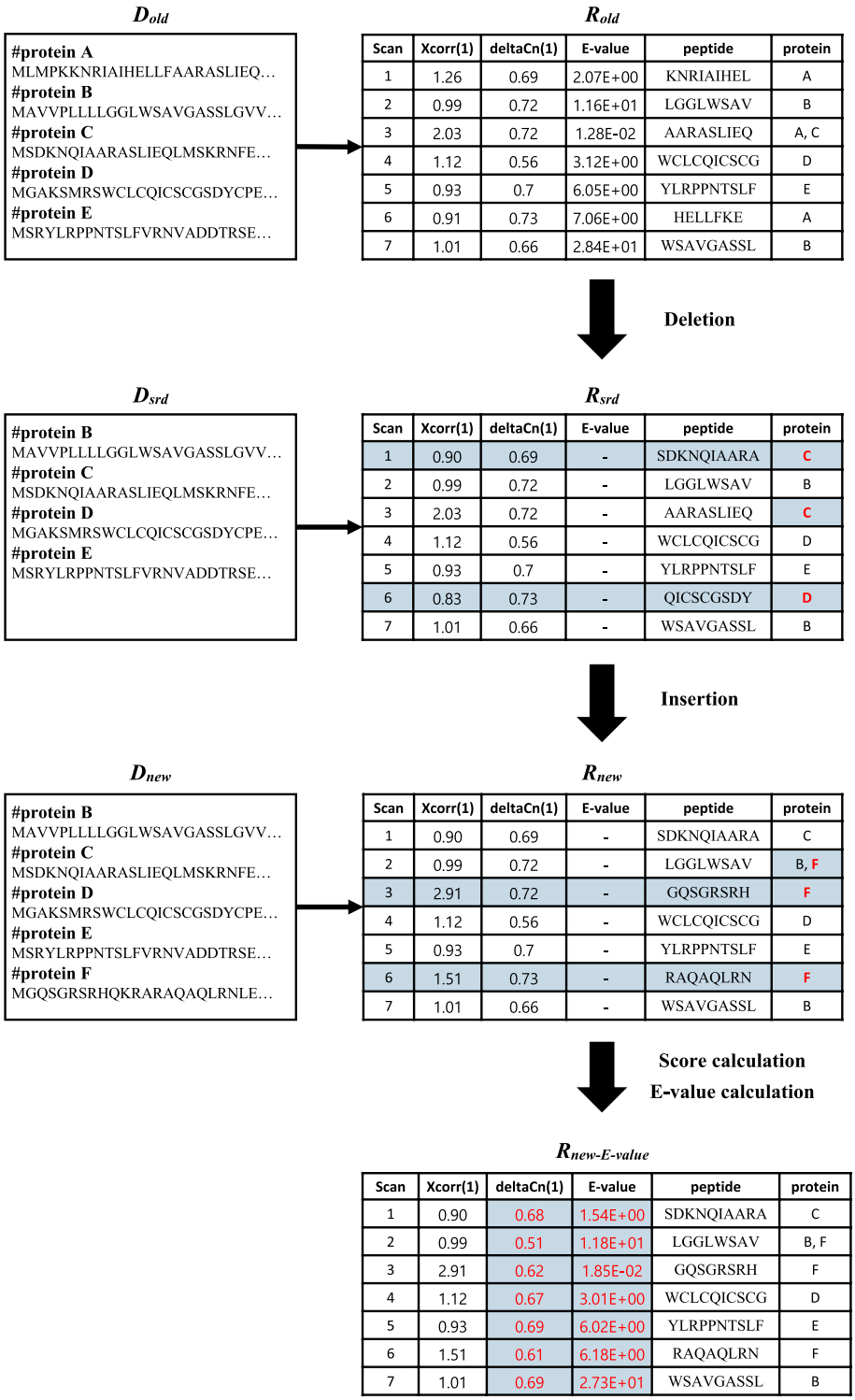


Fig. 4 Differences between various Uniprot versions. We used different Uniprot database versions from January to June 2020 for database comparison. The databases were compared based on the number of proteins and amino acids

(2) *Insertion* This is the process of obtaining R_{new} from R_{srd}. We search D_{ins} for all the spectra to find PSMs. Then the found PSMs are compared with the PSMs in R_{srd}. The PSMs with better scores are selected and stored in R_{new}. For example, PSMs of scans 2, 3 and 6 are updated after insertion in Fig. 5.

(3) *Score calculation* This is the process of calculating deltaCn values in R_{new}. deltaCn is a score representing the difference between Xcorr values. Since we got the Xcorr of R_{new} through previous steps, we can calculate the deltaCn of R_{new} in this step.



Scan	Xcorr(1)	deltaCn(1)	E-value	peptide	protein
1	1.26	0.69	2.07E+00	KNRIAIHEL	A
2	0.99	0.72	1.16E+01	LGGLWSAV	B
3	2.03	0.72	1.28E-02	AARASLIEQ	A, C
4	1.12	0.56	3.12E+00	WCLCQICSCG	D
5	0.93	0.7	6.05E+00	YLRPPNTSLF	E
6	0.91	0.73	7.06E+00	HELLFKE	A
7	1.01	0.66	2.84E+01	WSAVGASSL	B

Scan	Xcorr(1)	deltaCn(1)	E-value	peptide	protein
1	0.90	0.69	-	SDKNQIAARA	C
2	0.99	0.72	-	LGGLWSAV	B
3	2.03	0.72	-	AARASLIEQ	C
4	1.12	0.56	-	WCLCQICSCG	D
5	0.93	0.7	-	YLRPPNTSLF	E
6	0.83	0.73	-	QICSCGSDY	D
7	1.01	0.66	-	WSAVGASSL	B

Scan	Xcorr(1)	deltaCn(1)	E-value	peptide	protein
1	0.90	0.69	-	SDKNQIAARA	C
2	0.99	0.72	-	LGGLWSAV	B, F
3	2.91	0.72	-	GQSGRSRH	F
4	1.12	0.56	-	WCLCQICSCG	D
5	0.93	0.7	-	YLRPPNTSLF	E
6	1.51	0.73	-	RAQAQLRN	F
7	1.01	0.66	-	WSAVGASSL	B

Scan	Xcorr(1)	deltaCn(1)	E-value	peptide	protein
1	0.90	0.68	1.54E+00	SDKNQIAARA	C
2	0.99	0.51	1.18E+01	LGGLWSAV	B, F
3	2.91	0.62	1.85E-02	GQSGRSRH	F
4	1.12	0.67	3.01E+00	WCLCQICSCG	D
5	0.93	0.69	6.02E+00	YLRPPNTSLF	E
6	1.51	0.61	6.18E+00	RAQAQLRN	F
7	1.01	0.69	2.73E+01	WSAVGASSL	B

Fig. 5 Workflow overall

(4) *E-value calculation* This is the process of calculating E-values in R_{new}. Note that the E-value of every PSM may be invalid even if only one of all PSMs has been changed. Since E-value calculation requires all PSM information that has not been output by

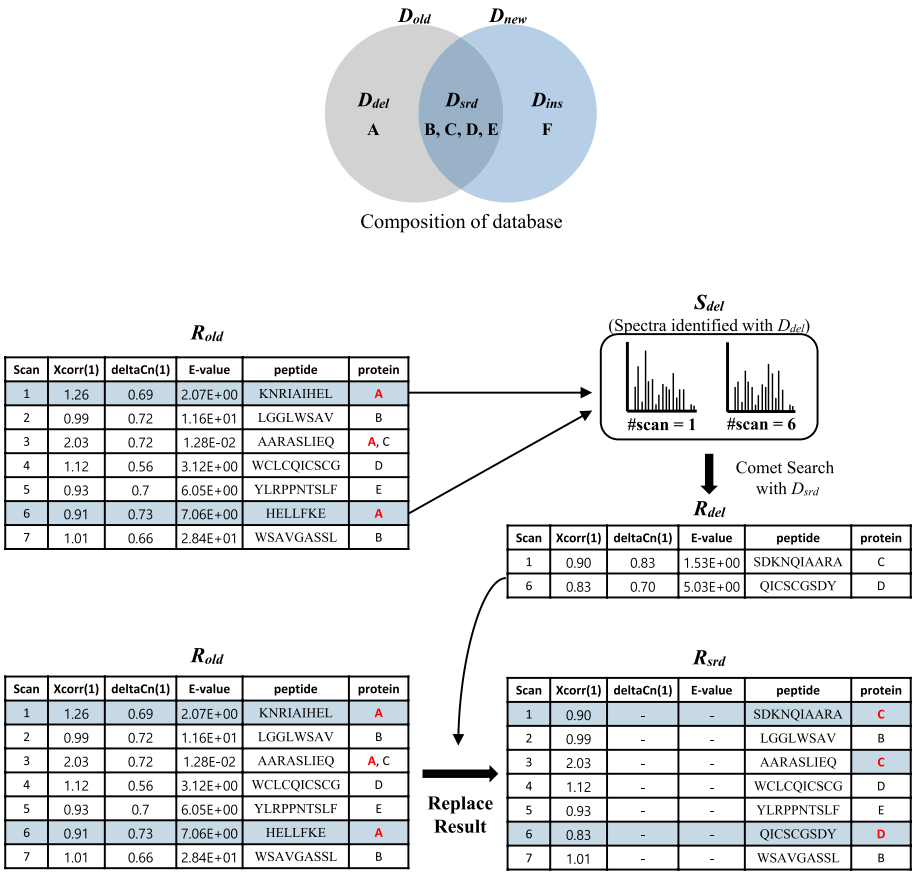
the original Comet, we built "Comet-E", a modified version of Comet, to address the E-value correction.

Detailed explanations are given in the following subsections: Deletion, Insertion, Score calculation, and E-value calculation.

Deletion

Algorithm description: The main purpose of deletion is converting R_{old} into R_{srd}. Each spectrum in R_{old} was identified by either D_{del} or D_{srd} (Fig. 6, Composition of database). Recall that S_{del} denote the set of spectra identified by only D_{del} and let S_{srd} denote the set of spectra identified by D_{srd}. While the PSMs for S_{srd} remain as they are, the PSMs for S_{del} should be replaced by the PSMs obtained by searching D_{srd} for S_{del}. In the example in Fig. 6, among the PSMs for scans 1–7, only the PSMs for scans 1 and 6 are identified with only D_{del} (= {A}). Thus, S_{del} consists of spectra in scans 1 and 6. (Note that the PSM for scan 3 belongs to S_{srd} because its peptide AARASLIEQ exists in both proteins A and C and thus all we have to do is to delete A from the protein list of scan 3.) We search D_{srd} (= {B, C, D, E}) for S_{del} and the new results replace the old results of S_{del}.

Time complexity: Since only D_{srd} is searched for S_{del}, the time complexity is $O(|S_{del}|\cdot|D_{srd}|)$. We show that $O(|S_{del}|\cdot|D_{srd}|)$ is reduced to $O(|S|\cdot|D_{del}|)$ on average where



Scan	Xcorr(1)	deltaCn(1)	E-value	peptide	protein
1	1.26	0.69	2.07E+00	KNRIAIHEL	A
2	0.99	0.72	1.16E+01	LGGLWSAV	B
3	2.03	0.72	1.28E-02	AARASLIEQ	A, C
4	1.12	0.56	3.12E+00	WCLCQICSCG	D
5	0.93	0.7	6.05E+00	YLRPPNTSLF	E
6	0.91	0.73	7.06E+00	HELLFKE	A
7	1.01	0.66	2.84E+01	WSAVGASSL	B

Scan	Xcorr(1)	deltaCn(1)	E-value	peptide	protein
1	0.90	0.83	1.53E+00	SDKNQIAARA	C
6	0.83	0.70	5.03E+00	QICSCGSDY	D

Scan	Xcorr(1)	deltaCn(1)	E-value	peptide	protein
1	1.26	0.69	2.07E+00	KNRIAIHEL	A
2	0.99	0.72	1.16E+01	LGGLWSAV	B
3	2.03	0.72	1.28E-02	AARASLIEQ	A, C
4	1.12	0.56	3.12E+00	WCLCQICSCG	D
5	0.93	0.7	6.05E+00	YLRPPNTSLF	E
6	0.91	0.73	7.06E+00	HELLFKE	A
7	1.01	0.66	2.84E+01	WSAVGASSL	B

Scan	Xcorr(1)	deltaCn(1)	E-value	peptide	protein
1	0.90	-	-	SDKNQIAARA	C
2	0.99	-	-	LGGLWSAV	B
3	2.03	-	-	AARASLIEQ	C
4	1.12	-	-	WCLCQICSCG	D
5	0.93	-	-	YLRPPNTSLF	E
6	0.83	-	-	QICSCGSDY	D
7	1.01	-	-	WSAVGASSL	B

Fig. 6 Deletion example

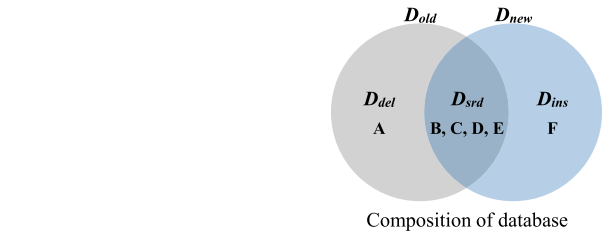
S is the set of total spectra $S_{del} \cup S_{srd}$. If we assume that PSMs were randomly selected from D_{old} in general (this assumption is verified in Results), the ratio $|S_{del}|/|S|$ is similar to the ratio $|D_{del}|/|D_{old}|$. Thus, $|S_{del}|$ is approximately the same as $|S| \cdot |D_{del}|/|D_{old}|$ and the time complexity can be expressed as $O(|S| \cdot |D_{del}| \cdot |D_{srd}|/|D_{old}|)$. Furthermore, since $|D_{srd}|/|D_{old}| \leq 1$, the time complexity is reduced to $O(|S| \cdot |D_{del}|)$ on average.

Insertion

The main purpose of insertion is converting R_{srd} into R_{new}. Each spectrum in R_{new} is identified by either D_{ins} or D_{srd}. First, we search D_{ins} for the set S. Let R_{ins} denote the search result. For each spectrum, we replace its PSM in R_{srd} by its PSM in R_{ins} if the Xcorr of the PSM in R_{ins} is higher than that in R_{srd}. In Fig. 7, we search D_{ins} (= {F}) for all the spectra and get R_{ins}. Since only the PSMs of scans 3 and 6 in R_{ins} have higher Xcorr values (2.91 and 1.51) than those in R_{srd} (2.03 and 0.83), R_{new} is obtained by replacing the PSMs of scans 3 and 6 in R_{srd} with those in R_{ins}. (Note that the scan 2 result of R_{ins} is the same as R_{srd} because its peptide LGGLWSAV exists in both proteins B and F and thus all we have to do is to add F to the protein list of scan 2.) Since the main part of insertion is to search D_{ins} for the set S, the time complexity is $O(|S| \cdot |D_{ins}|)$.

Score calculation

After the deletion and insertion, all PSMs with their Xcorr scores have been updated for D_{new}. Now, the deltaCn values which are defined as follows should be recalculated.



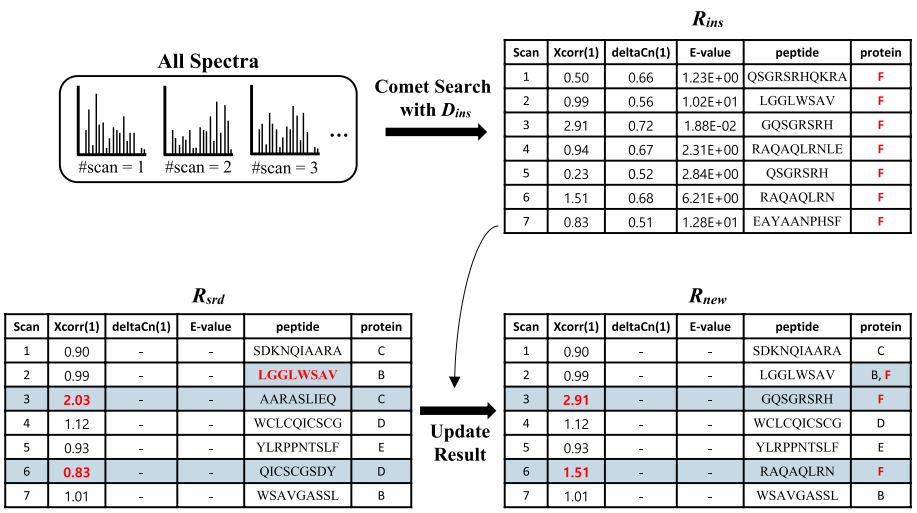


Fig. 7 Insertion example

$$\text{deltaCn}(i) = 1 - \text{Xcorr}(i+1) \,/\, \text{Xcorr}(i)$$

where Xcorr(i) denote the i-th largest PSM score for a spectrum. Thus, recalculating deltaCn takes $O(|S|)$ $(= O(|\text{PSM}|))$ time in the worst case. In addition, when deltaCn is updated, there are two subtleties to consider as follows.

(i) **Increment of the parameter *num_output_lines* by 1**

In order to calculate deltaCn(i), not only Xcorr(i) but also Xcorr($i+1$) is required. Since the parameter *num_output_lines* of Comet determines the number of Xcorr values in the output, *num_output_lines* should be $n+1$ if deltaCn(i)'s for $i \leq n$ are to be calculated by progressive search (Comet-P or Comet-E). Even though Comet just outputs n lines, it always calculates the Xcorr values for PSMs of all ranks, and thus incrementing *num_output_lines* by 1 rarely affects the total running time.

(ii) **Xcorr precision refinement in the output**

In Comet, the internal data type of Xcorr is double but the Xcorr values in the output of Comet are rounded to the fourth decimal place as shown in Table 1. Thus, the deltaCn(1) calculated by Comet is different from the deltaCn(1) calculated by Xcorr(1) and Xcorr(2) values from the output of Comet as explained in the legend of Table 1. Hence, the Xcorr values in the output of Comet-P/Comet-E are rounded to the seventh decimal place so that the deltaCn calculated by the output of Comet-P/Comet-E is the same as that calculated by Comet.

E-value calculation

The purpose of "E-value calculation" is converting R_{new} into $R_{new\text{-}E\text{-}value}$. For example, we explain how to calculate E-values of Comet. We built "Comet-E", a modified version of Comet, to address the E-value correction. Comet-E has two more features than the original Comet. First, it can output the histogram of Xcorr values which was just an intermediate data structure used to calculate E-values in Comet. Second, it can take a histogram of Xcorr values as input and calculate E-values based on the histogram. Let His(R) denote a histogram for a result set R. We calculate His(R_{new}) as follows: First, we run Comet-E to acquire histograms His(R_{del}) and His(R_{ins}). Then, His(R_{new}) is calculated by "$\text{His}(R_{old}) - \text{His}(R_{del}) + \text{His}(R_{ins})$" where His($R_{old}$) was already produced earlier by Comet-E. Finally, His(R_{new}) is given as input to Comet-E and it recalculates the E-value. Then, R_{new} is converted into $R_{new\text{-}E\text{-}value}$. Detailed explanations are given

Table 1 Xcorr precision refinement

	Comet (internal)	Comet (output)	Comet-P/Comet-E (output)
Xcorr(1)	0.0171654	0.0172	0.0171654
Xcorr(2)	0.0324375	0.0324	0.0324375
deltaCn(1)	0.4708162	0.4708	0.4708

Note that 0.4708, deltaCn(1) in the Comet (output) column, is achieved by rounding 0.4708162 (= 1 − 0.0171654/0.0324374) which is the deltanCn(1) in the Comet (internal) column. Thus, if deltaCn(1) is calculated from 0.0172 to 0.0324 which are the Xcorr values in the Comet (output) column, it becomes 0.4691 (= 1 − 0.0172/0.0324) which is different from 0.4708. Hence, the output of Xcorr values of Comet-P/Comet-E is rounded to the 7th decimal place so that deltaCn(1) is calculated by 1 − (0.0171654/0.0324374), which amount to 0.4708162 and then it is rounded to 0.4708

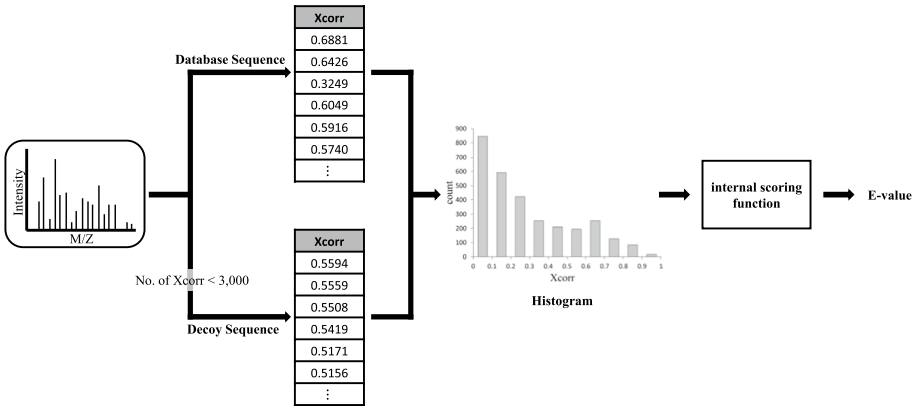


Fig. 8 E-value calculation of Comet

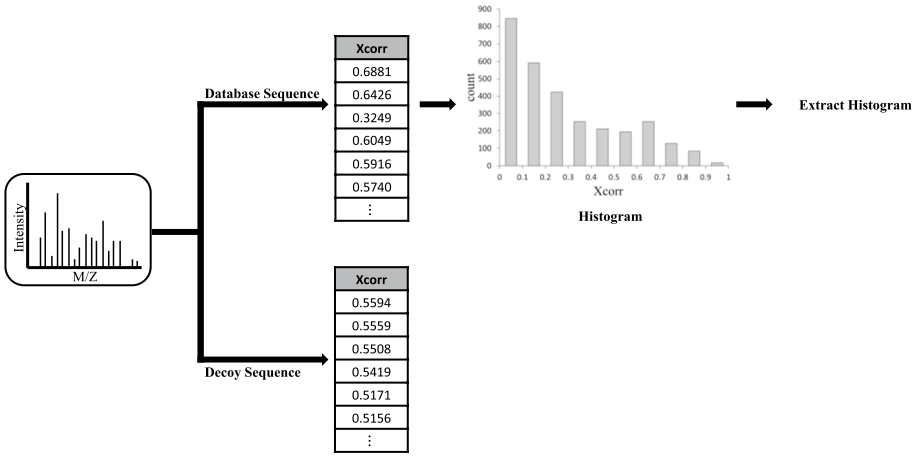


Fig. 9 Histogram output by Comet-E

in the following subsections i), ii), and iii). Subsection i) explains the E-value calculation by Comet and subsections ii) and iii) explain the two new features of Comet-E.

(i) E-value calculation by Comet

Comet calculates the E-value for each spectrum based on Xcorr values for all candidate peptides (Fig. 8). Comet needs at least 3000 Xcorr values for each spectrum to calculate its E-value. Comet uses decoy peptides predefined in Comet if the number of Xcorr values is less than 3,000. Then, Comet calculates the histogram of the Xcorr values for each spectrum. The histogram is used to calculate the E-value of each spectrum by the internal scoring function of Comet.

(ii) Comet-E (output)

Comet-E can output the histogram of Xcorr values for each spectrum (Fig. 9). The histogram consists of Xcorr values for the sequences in the database only, excluding decoy sequences. Unlike Comet, Comet-E outputs the histogram table for every spectrum as a.txt file. Histograms are created with a bin width of 0.1, and has an average

of 10 bin counts per spectrum. So, the histogram information (Xcorr counts for all bins) for each spectrum can be represented using only about 20 numbers.

(iii) **Comet-E (E-value recalculation)**

Given His(R_{new}) as input, Comet-E can calculate the E-values of R_{new} (Fig. 10). Note that His(R_{new}) is calculated by "His(R_{old}) − His(R_{del}) + His(R_{ins})". This calculation is performed for each bin. If there is no output for a bin among histograms, its frequency is assigned to 0. Note that His(R_{old}) was produced by Comet-E when R_{old} was generated and His(R_{del}) and His(R_{ins}) are produced by Comet-E when R_{del} and R_{ins} are generated, respectively. The time complexity of E-value calculation is O($|S|$) because it is regardless of the size of database difference and only proportional to the number of spectra.

Results

We measured and compared the running times of Comet, Comet-P (progressive Comet with PSM update only), and Comet-E (progressive Comet with both PSM and E-value update). The databases used were the SwissProt and TrEMBL human protein databases provided by UniProt. And tandem mass spectrometry (MS/MS) spectra for HEK293 cells [13] were used as an input, and the total number of spectra was 1,121,149. We compared them in different parameter settings: In subsection i), we show the results when the difference between D_{old} and D_{new} is fixed and the numbers of tryptic termini (ntt) and missed cleavages (mc) change. In subsection ii), we show the results when the difference between D_{old} and D_{new} changes and ntt and mc are fixed. The search results of Comet, Comet-P, and Comet-E remain consistent for both PSM and peptide levels (Fig. 11).

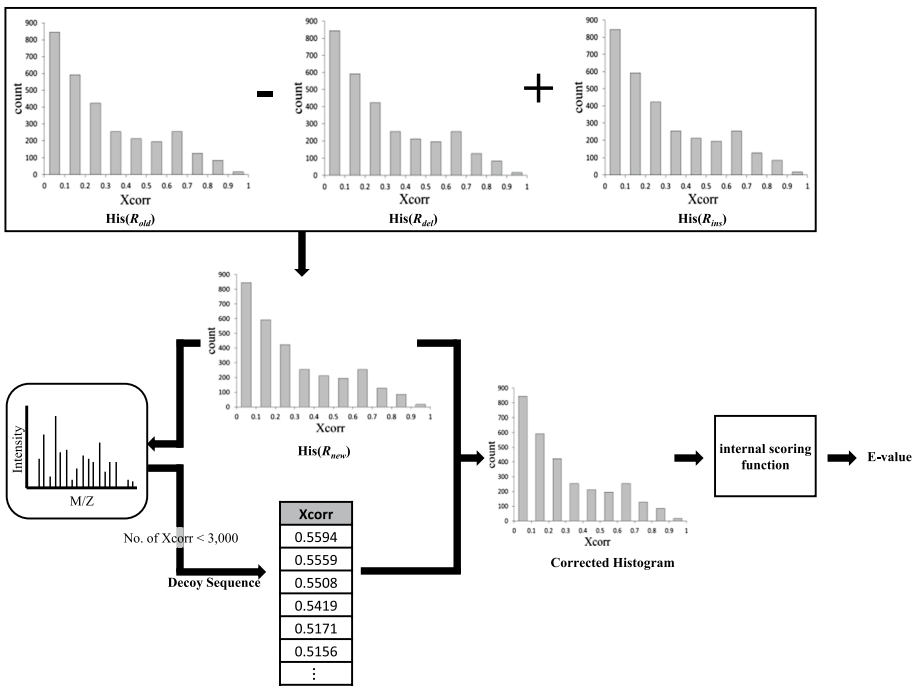


Fig. 10 E-value recalculation by Comet-E

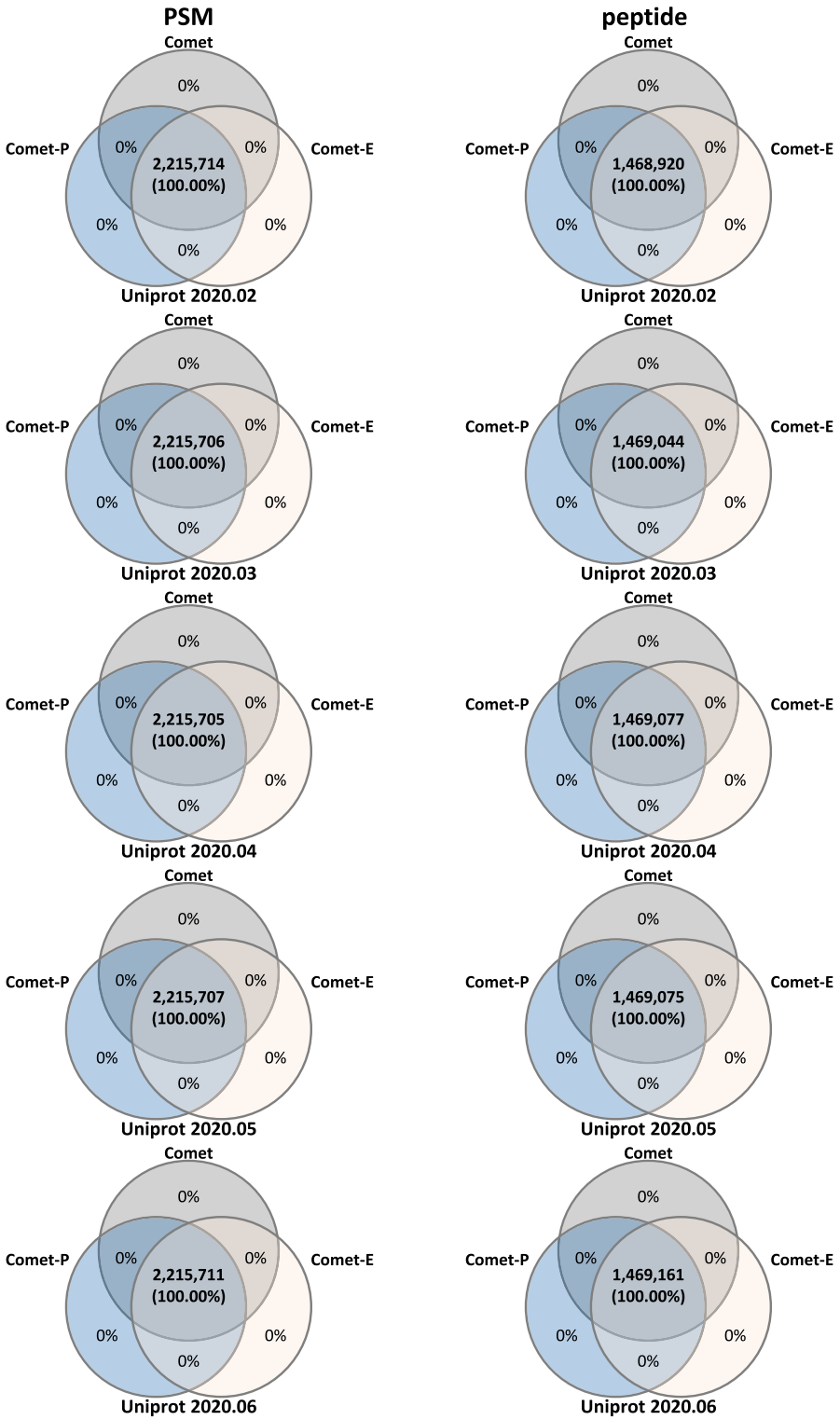


Fig. 11 Consistency between Comet and Progressive Search results at the PSM and peptide level. For comparison, Progressive Search used the results analyzed using databases updated from Uniprot 2020.01 to Uniprot 2020.02, Uniprot 2020.03, Uniprot 2020.04, Uniprot 2020.05, and Uniprot 2020.06 versions. Comparisons were made for Comet, Comet-E and Comet-P for ntt1mc2

The entire experiments were carried out on a Linux PC with an Intel(R) Xeon(R) octa-core CPU E5-2609 v3 @ 1.90 GHz and 36 GB of RAM. The Linux version is Ubuntu 12.04.5 LTS and the compiler is GNU C compiler 6.5.0. All experiments were performed by a single thread.

(i) **Changing the numbers of tryptic termini and missed cleavages**

Table 2A shows the running time results when D_{old} and D_{new} are fixed to Uniprot 2020.01 and Uniprot 2020.02, respectively and ntt changes from 0 to 1 and mc changes from 0 to 2. Note that the difference between D_{old} (Uniprot 2020.01) and D_{new} (Uniprot 2020.02) is 0.16% (Fig. 4 #amino acid). Table 2A shows not only the overall running times of Comet, Comet-P, and Comet-E, but also breaks down the overall running times of Comet-E into the running times of individual modules (database separation, deletion, insertion, and E-value calculation). Note that the running time of Comet-P is the sum of the running times of all individual modules except the E-value calculation. For example, look at the leftmost column ntt2mc0. In this case, Comet, Comet-P, and Comet-E take 7846.9, 459.1, and 1900.1 s, respectively. The 459.1 s which is the running time of Comet-P is the sum of 3.7 s (database separation), 244.1 s (deletion), and 211.3 s (insertion). The 1900.1 s which is the running time of Comet-E is the sum of 459.1 s (Comet-P) and 1441.0 s (E-value calculation).

Table 2B shows the statistics of the running time results in Table 2A. The running time ratio rows show the ratios of individual running times to the running time of Comet. Look at the leftmost column ntt2mc0 again. Since the running time of Comet-P is 459.1 s and that of the original Comet is 7846.9, the ratio is $459.1/7846.9 = 0.0585 = 5.85\%$. Since the ratio is 5.85%, Comet-P is 17.09 (=(1/5.85)*100) times faster than Comet which is shown just below 5.85% in the table. The speedup of Comet-P is between 17.09 (ntt2mc0) and 53.92 (ntt1mc2) and the speedup of Comet-E is between 4.13 (ntt2mc0) and 16.52 (ntt1mc2). Hence, the more nontryptic termini and missed cleavages there are, the bigger the speedup is.

Finally, it should be noted that the E-value calculation time does not change a lot as the nontryptic termini or missed cleavages change. It is between 1261.4 and 1459.9 s as shown in the last row of Table 2A. It may seem strange on a first look but it is reasonable because the time complexity of E-value calculation is just $O(|S|)$ which means it is regardless of the size of database difference.

(ii) **Changing database update interval**

Table 3 shows the running times and their statistics of Comet, Comet-P, and Comet-E when ntt and mc are fixed to 1 and 2, respectively and the database update interval changes from 1 to 5 months. In this experiment, D_{old} is fixed to Uniprot 2020.01 and D_{new} changes appropriately from Uniprot 2020.02 to Uniprot 2020.06. The ratio $|D_{del}|/|D_{new}|$ increases from 0.02% to 0.44% and the ratio $|D_{ins}|/|D_{new}|$ also increases from 0.14% to 3.48% as the database update interval increases as shown in the last two rows in Table 3B. Recall that the time complexities of deletion and insertion are $O(|S|\cdot|D_{del}|)$ and $O(|S|\cdot|D_{ins}|)$, respectively. Thus, their running times are expected to increase as the database update interval increases. As expected, the measured running

Table 2 Summary of the running times for various search parameter settings

	ntt2mc0	ntt2mc1	ntt2mc2	ntt1mc0	ntt1mc1	ntt1mc2
A (unit: second)						
1. Comet search	7846.9	7967.0	8320.1	15,084.1	25,650.1	34,287.5
2-a. Comet-P (without E-value)	459.1	447.9	442.0	528.0	575.3	635.9
2-b. Comet-E (with E-value)	1900.1	1709.3	1711.2	1911.1	2035.3	2075.7
2.1. Database separation	3.7	3.3	3.1	3.3	3.0	3.0
2.2. Deletion	244.1	228.2	223.1	261.5	280.2	344.3
2.2.1. Extract spectra	14.8	15.9	14.4	16.6	14.2	15.6
2.2.2. Comet-P/E search	72.5	57.9	58.4	78.5	91.8	158.7
2.2.3. Update results	156.9	154.3	150.3	166.4	174.2	170.0
2.3. Insertion	211.3	216.4	215.7	263.1	292.1	288.6
2.3.1. Comet-P/E search	168.9	168.6	162.9	192.6	216.1	213.7
2.3.2. Update results	42.4	47.8	52.8	70.5	76.0	74.9
2.4. E-value calculation	1441.0	1261.4	1269.3	1383.1	1459.9	1439.8
B						
The running time ratio						
Comet-P	5.85% (× 17.09)	5.62% (× 17.79)	5.31% (× 18.82)	3.50% (× 28.57)	2.24% (× 44.59)	1.85% (× 53.92)
Comet-E	24.21% (× 4.13)	21.45% (× 4.66)	20.57% (× 4.86)	12.67% (× 7.89)	7.93% (× 12.60)	6.05% (× 16.52)
Deletion	3.11%	2.86%	2.68%	1.73%	1.09%	1.00%
Insertion	2.69%	2.72%	2.59%	1.74%	1.14%	0.84%
E-value calculation	18.36%	15.83%	15.26%	9.17%	5.69%	4.20%
Database size ratio						
$\|D_{del}\|/\|D_{new}\|$*	0.02% (8821 / 56,400,212)					
$\|D_{ins}\|/\|D_{new}\|$*	0.14% (76,988 / 56,400,212)					
Deleted PSMs ratio ($\|S_{del}\|/\|S\|$)	0.02% (221 / 1,121,149)	0.02% (249 / 1,121,149)	0.02% (261 / 1,121,149)	0.02% (222 / 1,121,149)	0.02% (221 / 1,121,149)	0.02% (226 / 1,121,149)

(A) Running time comparison between Comet, Comet-P, and Comet-E in various search parameter settings. (B) Statistics of the running times. *The numerator and the denominator are the numbers of amino acids in target database only. If they are the numbers in both target and decoy databases, they are doubled and thus the ratio is still the same

time of deletion (resp. insertion) increases from 344.3 to 598.3 s (resp. from 288.6 to 1443.9) as the database update interval increases as shown in Table 3A. When it comes to the E-value calculation, since its time complexity is $O(|S|)$, its running time is regardless of the database update interval. It is between 1394.5 and 1496.7 s. Conclusively, the speedup of Comet-P is between 53.92 (1 month) and 17.39 (5 months) and the speedup of Comet-E is between 16.52 (1 month) and 10.23 (5 months) as shown in Table 3B.

Table 3 Summary of the running times for several database update intervals

	1 month	2 months	3 months	4 months	5 months
A (unit: second)					
1. Comet search	34,287.5	35,073.6	35,052.1	34,840.2	35,559.4
2-a. Comet-P (without E-value)	635.9	1282.6	1439.5	1532.8	2045.3
2-b. Comet-E (with E-value)	2075.7	2779.3	2834.0	2960.5	3476.1
2.1. Database separation	3.0	3.0	3.0	3.2	3.1
2.2. Deletion	344.3	410.0	459.0	498.8	598.3
2.2.1. Extract spectra	15.6	15.8	15.0	16.3	15.7
2.2.2. Comet-P/E search	158.7	184.9	199.5	216.2	251.7
2.2.3. Update results	170.0	209.3	244.5	266.3	330.8
2.3. Insertion	288.6	869.6	977.5	1030.8	1443.9
2.3.1. Comet-P/E search	213.7	787.8	893.9	946.9	1354.8
2.3.2. Update results	74.9	81.8	83.6	83.9	89.1
2.4. E-value calculation	1439.8	1496.7	1394.5	1427.8	1430.8
B					
The running time ratio					
Comet-P	1.85% (× 53.92)	3.66% (× 27.35)	4.11% (× 24.35)	4.40% (× 22.73)	5.75% (× 17.39)
Comet-E	6.05% (× 16.52)	7.92% (× 12.62)	8.09% (× 12.37)	8.50% (× 11.77)	9.78% (× 10.23)
Deletion	1.00%	1.17%	1.31%	1.43%	1.68%
Insertion	0.84%	2.48%	2.79%	2.96%	4.06%
E-value calculation	4.20%	4.27%	3.98%	4.10%	4.02%
Database size ratio					
$\lvert D_{del}\rvert/\lvert D_{new}\rvert$*	0.02% (8821 / 56,400,212)	0.11% (64,323 / 57,348,066)	0.21% (118,748 / 57,471,311)	0.26% (149,505 / 57,516,232)	0.44% (252,754 / 58,101,266)
$\lvert D_{ins}\rvert/\lvert D_{new}\rvert$*	0.14% (76,988 / 56,400,212)	1.88% (1,080,344 / 57,348,066)	2.19% (1,258,014 / 57,471,311)	2.32% (1,333,692 / 57,516,232)	3.48% (2,021,975 / 58,101,266)
Deleted PSMs ratio ($\lvert S_{del}\rvert/\lvert S\rvert$)	0.02% (226 / 1,121,149)	0.10% (1121 / 1,121,149)	0.15% (1717 / 1,121,149)	0.21% (2410 / 1,121,149)	0.31% (3426 / 1,121,149)

(A) Running time Comparison between Comet, Comet-P, and Comet-E with several database update intervals. (B) Statistics of the running times. *The numerator and the denominator are the numbers of amino acids in target database only. If they are the numbers in both target and decoy databases, they are doubled and thus the ratio is still the same

Conclusions

Progressive search is a novel approach to efficiently obtain analysis results for updated database in tandem mass spectrometry. Its running time is $O(|S||\Delta D|)$ on average and thus it is up to 53.9 times faster than the normal search from scratch for PSM update only (including the update of PSM scores such as Xcorr and DeltaCn) and up to 16.5 times faster for both PSM and E-value update for the intervals up to 5 months. We also discovered our Progressive search is effective even for longer intervals. Comet-P and Comet-E are 2.5 and 4 times faster than normal search, respectively, even with the interval of 34 months (July 2019 and May 2022 databases) (data not shown). The PSMs and E-values achieved by progressive search are the same as those achieved by the normal search from scratch. In addition, we verified that repeated use of progressive search does not increase the differences in deltaCn values due to rounding. We compared the results from searches for 3-month intervals (between Jan. 2020 and Apr. 2020) with results from 3 repeated searches for 1-month interval (between Jan. 2020 and Feb. 2020, between Feb. 2020 and Mar. 2020, and between Mar. 2020 and Apr. 2020). The deltaCn values were the same in both results although progressive search was used multiple times. This study demonstrates the applicability of Progressive search for efficient tandem mass spectrometry database search. Use of this approach can be extended to a variety of public search tools, including Comet.

Availability and requirements

Project name: progressive search.

Project home page: https://isa.hanyang.ac.kr/ProgSearch.html

Operating system(s): Linux.

Programming language: Java, C++

Other requirements: JDK 1.8 or higher.

License: Apache License V2.0

Any restrictions to use by non-academics: as stipulated by Apache License V2.0

Abbreviations

S	Set of spectra
D	Set of peptides in a database
D_{new}	New database
D_{old}	Old database
D_{srd}	Database which contains the proteins shared by both D_{old} and D_{new}
D_{del}	Database which contains the proteins stored in only D_{new}
D_{ins}	Database which contains the proteins stored in only D_{new}
R_{new}	PSM results for D_{new}
R_{old}	PSM results for D_{old}
R_{srd}	PSM results for D_{srd}
R_{del}	PSM results for D_{del}
R_{ins}	PSM results for D_{ins}

Acknowledgements

Not applicable

Author contributions

YJ, KL, HK, and HP designed the study. YJ and KL performed computing experiments. All authors wrote, read and approved the final manuscript.

Funding

This work was supported by the National Research Foundation of Korea grant funded by the Korea government (Ministry of Science and ICT) (No. 2018R1A5A7059549 and No. 2021M3H9A2030520), and by the Korea Institute of Science

and Technology Information (KISTI) and Korea Bio Data Station (K-BDS) with computing resources including technical support.

Availability of data and materials
Experiments were carried out with the August 2019 version of Comet and can be obtained through Comet website: http://comet-ms.sourceforge.net. The database used in this current study are publicly available in the UniProt website: https://www.uniprot.org. The databases used were the SwissProt and TrEMBL human protein databases provided by UniProt. We measured the performance of Progressive Search using tandem mass spectrometry (MS/MS) spectra for HEK293 cells [13]. The HEK293 24-fraction MS/MS dataset was used in the experiment, and the total number of spectra was 1,121,149.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 4 August 2022 Accepted: 3 March 2023
Published online: 14 March 2023

References

1. Eng JK, McCormack AL, Yates JR. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. J Am Soc Mass Spectrom. 1994;5(11):976–89.
2. Diament BJ, Noble WS. Faster SEQUEST searching for peptide identification from tandem mass spectra. J Proteome Res. 2011;10(9):3871–9.
3. Eng JK, Jahan TA, Hoopmann MR. Comet: an open-source MS/MS sequence database search tool. Proteomics. 2013;13(1):22–4.
4. Perkins DN, Pappin DJ, Creasy DM, Cottrell JS. Probability-based protein identification by searching sequence databases using mass spectrometry data. ELECTROPHORESIS Int J. 1999;20(18):3551–67.
5. Cox J, Mann M. MaxQuant enables high peptide identification rates, individualized ppb-range mass accuracies and proteome-wide protein quantification. Nat Biotechnol. 2008;26(12):1367–72.
6. Kim S, Gupta N, Pevzner PA. Spectral probabilities and generating functions of tandem mass spectra: a strike against decoy databases. J Proteome Res. 2008;7(8):3354–63.
7. Kong AT, Leprevost FV, Avtonomov DM, Mellacheruvu D, Nesvizhskii AI. MSFragger: ultrafast and comprehensive peptide identification in mass spectrometry–based proteomics. Nat Methods. 2017;14(5):513–20.
8. Frank AM, Bandeira N, Shen Z, Tanner S, Briggs SP, Smith RD, Pevzner PA. Clustering millions of tandem mass spectra. J Proteome Res. 2008;7(01):113–22.
9. Griss J, Perez-Riverol Y, Lewis S, Tabb DL, Dianes JA, Del-Toro N, Rurik M, Walzer M, Kohlbacher O, Hermjakob H. Recognizing millions of consistently unidentified spectra across hundreds of shotgun proteomics datasets. Nat Methods. 2016;13(8):651–6.
10. Wu CH, Apweiler R, Bairoch A, Natale DA, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, et al. The Universal protein resource (UniProt): an expanding universe of protein information. Nucleic Acids Res. 2006;34:D187-191.
11. Deutsch EW, Mendoza L, Shteynberg D, Farrah T, Lam H, Tasman N, Sun Z, Nilsson E, Pratt B, Prazen B. A guided tour of the trans-proteomic pipeline. Proteomics. 2010;10(6):1150–9.
12. McIlwain S, Tamura K, Kertesz-Farkas A, Grant CE, Diament B, Frewen B, Howbert JJ, Hoopmann MR, Käll L, Eng JK. Crux: rapid open source protein tandem mass spectrometry analysis. J Proteome Res. 2014;13(10):4488–91.
13. Chick JM, Kolippakkam D, Nusinow DP, Zhai B, Rad R, Huttlin EL, Gygi SP. A mass-tolerant database search identifies a large proportion of unassigned spectra in shotgun proteomics as modified peptides. Nat Biotechnol. 2015;33(7):743–9.

Publisher's Note