ECL 2.0: Exhaustively Identifying Cross-Linked Peptides with a Linear Computational Complexity

Fengchao Yu[1], Ning Li[1,2], and Weichuan Yu*[1,3]

[1] Division of Biomedical Engineering, The Hong Kong University of Science and Technology, Hong Kong, China.
[2] Division of Life Science, The Hong Kong University of Science and Technology, Hong Kong, China.
[3] Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong, China.

Abstract. Chemical cross-linking coupled with mass spectrometry is a powerful tool to study protein-protein interactions and protein conformations. Two linked peptides were ionized and fragmented to produce a tandem mass spectrum. In such an experiment, a tandem mass spectrum contains ions from two peptides. The peptide identification problem becomes a peptide pair identification problem. Most tools, however, don't search all possible pairs due to the long computational time. Consequently, a significant proportion of all linked peptides are missed. In our earlier work, we developed a tool named ECL to search all pairs of peptides exhaustively. However, compared to tools only pairing a small number of peptides, ECL is still too slow when the database is large.

Here, we propose an advanced version of ECL, named ECL 2.0. It achieves linear time and space complexity by taking advantage of the additive property of a score function. It can exhaustively search tens of thousands of spectra against a database containing thousands of proteins in a few hours. Among another four state-of-the-art tools, ECL 2.0 is much faster than pLink, StavroX, and ProteinProspector, but slower than Kojak which, however, only searches the smallest proportion of all peptide-peptide pairs among the five tools.

Keywords: Cross-Linking, Peptide Identification, Exhaustive Database Search

1 Introduction

The power of chemical cross-linking coupled with mass spectrometry (XL-MS) has been well demonstrated in modeling protein structures and understanding protein-protein interactions [8, 9, 30, 34, 49]. However, identifying cross-linked peptides from XL-MS data is computationally challenging. The time complexity is quadratic with respect to the number of peptides in a database. Thus, exhaustively searching all peptide-peptide pairs is time consuming and resource demanding. For example, there are around 3×10^6 peptides in the homo sapiens (human) database (UniProtKB / Swiss-Prot, 2015-11 release, 20,205 proteins). Supposing the precursor mass tolerance is 10 ppm (parts per million), there will be, on average, around 10^7 peptide-peptide pairs for each spectrum.

Many methods [1, 3, 6, 7, 10, 11, 13, 19, 22, 25, 27–29, 31, 36, 37, 40, 42, 45–47] have been developed. These methods can be classified into two groups. The first group converts searching peptide-peptide pairs into searching two peptides sequentially with the help of special cross-linkers. The second group limits the number of peptide-peptide pairs with heuristic pre-filtering.

Methods in the first group convert the quadratic time complexity into a linear time complexity by using mass-spectrometry-cleavable cross-linkers [15, 17, 32, 33]. This kind of cross-linkers can be broken during dissociation (e.g. collision-induced dissociation (CID)). Huang *et al.* proposed to couple such cross-linkers with three levels of mass spectrometry (i.e. MS1, MS2, and MS3) [15, 17]. Generating three levels of mass spectra requires a longer cycle time. Liu *et al.* [6, 23] proposed to use cross-linker-cleaved signature peaks to infer the masses of two peptide chains. This method

* Email: eeyu@ust.hk

avoids generating three levels of mass spectrometry. However, the cross-linker-cleaved signature peaks may sometimes not be observed, which results in loss of useful data.

xQuest/xProphet [35, 43], pLink [45], ProteinProspector [42], StavroX [7], and Kojak [11] are typical tools in the second group. They only keep a fixed number of peptides to generate peptide-peptide pairs for each experimental spectrum. For example, xQuest/xProphet first uses the top 5000 peptides for pairing. Then, it filters all peptide-peptide pairs with a fast pre-score. Finally, the top 50 peptide-peptide pairs are used for fine scoring. Similarly, pLink, ProteinProspector, and Kojak use the top 500, 1000, and 250 peptides to generate peptide-peptide pairs, respectively. Such a strategy, however, only searches a fraction of all possible peptide-peptide pairs. Let's take the homo sapiens (human) database as an example. There are around 10^7 peptide-peptide pairs for each spectrum. With a rough estimation, pLink, ProteinProspector, and Kojak only search about 1.2%, 5.0%, and 0.3% of all peptide-peptide pairs, respectively. Thus, the non-exhaustive search strategy cannot guarantee to find a spectrum's highest scored peptide-peptide pair. The result is highly variable with respect to database size. Our experiments show that the sensitivity decreases greatly as the size of the database increases. Meanwhile, the corresponding false discovery proportion (FDP) increases considerably.

In XL-MS, using a noncleavable amine-reactive cross-linker, such as disuccinimidyl suberate (DSS) and bis(sulfosuccinimidyl) suberate (BS3), to link two proteins is a widely used protocol. In order to process XL-MS data using a noncleavable cross-linker, we developed a tool named ECL [48] that can exhaustively search a database in tens of hours. However, the running time of ECL increases quadratically with respect to the increase of the database size. Comparison with the four above-mentioned representative benchmark methods in the second group shows that, although ECL is faster than xQuest and pLink, it is slower than ProteinProspector [42] and Kojak [11] when the database is large.

In this paper, we propose a tool, named ECL 2.0, that can search all peptide-peptide pairs with a linear time and space complexity. Given a file with tens of thousands of MS2 spectra, ECL 2.0 can exhaustively search them against a database containing thousands of proteins in a few hours. ECL 2.0 is much faster than pLink, StavroX, and ProteinProspector. ECL 2.0 is still slower than Kojak. But ECL 2.0 explores 100% of the search space, while Kojak only explores a small fraction, as we just described.

The rest of the paper is organized as follows: Section 2 describe the algorithm of ECL 2.0. Section 3 demonstrates the performance of ECL 2.0 with real data sets. Section 4 concludes the paper.

2 Method

In XL-MS, we first link proteins with a cross-linker. Then, we quench the reaction and digest the proteins. Finally, after digestion, we obtain pairs of linked peptides. By convention [36, 45], two linked peptides are called α chain and β chain, respectively. After dissociation (e.g. CID), there are fragmented ions from two peptide chains. Figure 1 illustrates the ions with marks: green marks indicate linear ions that only contain one chain's ions; red marks indicate cross-linking ions that contain one chain's ions plus a modification containing the cross-linker and the other whole peptide.

Given an experimental spectrum, the objective of identifying cross-linked peptides can be expressed as

$$\begin{aligned} \max_{\mathbf{t}} \quad & s(\mathbf{e}, \mathbf{t}), \\ \text{s.t.} \quad & |m(\mathbf{e}) - m(\mathbf{t})| \leq \tau_1, \end{aligned} \tag{1}$$

where $s(\mathbf{e}, \mathbf{t})$ is a score function, $\mathbf{e}$ is the experimental spectrum, $\mathbf{t}$ is the theoretical spectrum of cross-linked peptides, $m(\bullet)$ is the precursor mass of a spectrum, and τ_1 is the precursor mass tolerance. Existing tools pair two peptide chains to generate the corresponding theoretical spectra, which results in a quadratic time complexity.

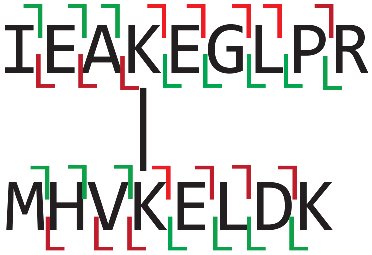


Fig. 1. An illustration of cross-linked peptides. Green marks indicate linear ions and red marks indicate cross-linking ions.

We observe that, with an additive score function [12, 20], the score corresponding to a peptide-peptide pair equals the sum of two scores corresponding to two peptide chains. Other researchers have also made this observation [27, 37], but they did not take advantage of it to reduce the time complexity. Based on this observation, we propose a new algorithm that achieves a linear time complexity. In the following, we first describe what an additive score function is. Then, we show ECL 2.0's algorithm and its computational complexity. Finally, we describe the work-flow of ECL 2.0.

2.1 Additive Score Function

Given a spectrum, we can digitize the whole m/z range into bins based on the MS2 m/z tolerance: $i = \frac{m}{\tau_2} - o$, where i is the index of the digitized bin, m is an m/z value, τ_2 is the MS2 m/z tolerance, and o is an offset. For the i-th bin, the corresponding intensity can be obtained as $v_i = \sum_{m=(i+o)\times\tau_2}^{(i+1+o)\times\tau_2} p_m$, where v_i is the i-th value in the digitized vector and p_m is the peak intensity whose m/z value is m. If there is no peak at the location m, $p_m = 0$. Then, we have the following definition:

Definition 1. *Given a digitized experimental spectrum* $\mathbf{e}$ *and a digitized theoretical spectrum* $\mathbf{t}$*, an additive score function reads*

$$s(\mathbf{e}, \mathbf{t}) = \sum_j \sum_i f(g_i(\mathbf{e}), t_j), \tag{2}$$

where $g_i(\mathbf{e})$ *is a measure of the* i*-th bin in the experimental spectrum,* t_j *is the* j*-th value in* $\mathbf{t}$*, and* $f(g_i(\mathbf{e}), t_j)$ *is a score term.*

Roughly speaking, there are two types of score functions in linear/cross-linked peptides identification tools [5, 11, 14, 18, 21, 27, 35, 41–45]:

1. Dot-product-based score functions, such as the dot product [18, 21, 42], intensity summation [35, 43], XCorr [5, 11, 14, 27, 35, 43], and the kernel spectral dot product (KSDP) [44, 45].
2. Probability based score functions, such as the match-odds [35, 43] and log-odds function [41].

The dot product, intensity summation and XCorr are additive score functions. For example, XCorr can be expressed as:

$$\begin{aligned} XCorr(\mathbf{e},\mathbf{t}) &= \sum_i e_i \times t_i - \frac{1}{150}\sum_i \sum_{\delta=-75,\delta\neq 0}^{75} e_{i+\delta} \times t_i \\ &= \sum_i (e_i - \frac{1}{150}\sum_{\delta=-75,\delta\neq 0}^{75} e_{i+\delta}) \times t_i \\ &= \sum_i g_i(\mathbf{e}) \times t_i, \end{aligned} \tag{3}$$

where δ is an m/z offset and $g_i(\mathbf{e}) = e_i - \frac{1}{150}\sum_{\delta=-75,\delta\neq 0}^{75} e_{i+\delta}$. Here, we assume that there are no overlapping peaks in the theoretical spectrum. The power of XCorr has been demonstrated in many methods [11, 12, 16, 35, 38]. Thus, we use it as the score function of ECL 2.0.

According to Figure 1, peaks forming a theoretical spectrum are from four sources: linear ions and cross-linking ions from α chain; linear ions and cross-linking ions from β chain. Correspondingly, a score of two linked peptides can be expressed as

$$\begin{aligned} s(\mathbf{e},\mathbf{t}) &= \sum_j \sum_i f(g_i(\mathbf{e}), t_j) \\ &= \sum_{j_\alpha} \sum_i f(g_i(\mathbf{e}), t_{j_\alpha}) + \sum_{j_\beta} \sum_i f(g_i(\mathbf{e}), t_{j_\beta}) \\ &= s(\mathbf{e},\mathbf{t}_\alpha) + s(\mathbf{e},\mathbf{t}_\beta), \end{aligned} \tag{4}$$

where j_α is a bin index corresponding to the peak from α chain, j_β is a bin index corresponding to the peak from β chain, $\mathbf{t}_\alpha$ is a digitized theoretical spectrum containing peaks from α chain only, and $\mathbf{t}_\beta$ is a digitized theoretical spectrum containing peaks from β chain only. Let's call $s(\mathbf{e},\mathbf{t}_\alpha)$ and $s(\mathbf{e},\mathbf{t}_\beta)$ chain scores for convenience. With Equation (4), Equation (1) can be expressed as

$$\begin{aligned} \max_{\mathbf{t}} \quad & s(\mathbf{e},\mathbf{t}_\alpha) + s(\mathbf{e},\mathbf{t}_\beta), \\ \text{s.t.} \quad & |m(\mathbf{e}) - m(\mathbf{t}_\alpha) - m(\mathbf{t}_\beta) - m_x| \leq \tau_1, \end{aligned} \tag{5}$$

where m_x is the mass of the cross-linker.

2.2 Searching Cross-Linked Peptides

Equation (5) implies that, given an experimental spectrum, we can first calculate all chain scores independently (Algorithm 1). Then, we can add pairs of chain scores whose total mass is in a certain range. The time complexity of adding all possible pairs of scores is quadratic with respect to the number of chain scores. In this section, we propose a digitization-based algorithm to achieve linear time complexity. We first describe the procedure of searching cross-linked peptides given an experimental spectrum. Then, we analyze this procedure's time and space complexity.

Given a database, we first *in silicon* digest all proteins into n peptide chains. All the peptide chains are sorted based on their masses. Then, we split the whole mass range into multiple intervals. The width of the intervals w is much smaller than the precursor mass tolerance τ_1. Here, we set the width w to 0.001 Da. The number of intervals is equal to $b = \frac{m_{max} - m_{min}}{w} + 1$, where b is the number of intervals, m_{max} is the maximal considered peptide chain mass, and m_{min} is the minimal considered peptide chain mass. All of these values are pre-fixed before the database search. Finally,

Algorithm 1 Calculating chain scores. Without loss of generality, we use ions fragmented from CID. It can be easily applied to other dissociation methods by changing b/y-ion to a/x-ion or c/z-ion. $\mathbf{b}$ is a vector of b-ion masses from the peptide chain, and $\mathbf{y}$ is a vector of y-ion masses from the peptide chain. We assume that the difference of any two masses is larger than the MS2 m/z tolerance. x is the mass of the cross-linker, m_c is the mass of the peptide chain, $\mathbf{e}$ is the digitized experimental spectrum, m_e is the mass of the experimental spectrum, τ_2 is the MS2 m/z tolerance, and o is the offset in digitization.

```
procedure ChainScore($\mathbf{b}, \mathbf{y}, x, m_c, \mathbf{e}, m_e, \tau_2, o$)
    $\mathbf{c} \leftarrow vector[len(\mathbf{e})]$
    $s \leftarrow 0$

    for $i \leftarrow 1, len(\mathbf{b})$ do                ▷ fill bins corresponding to b-ions
        if $i < x$ then
            $\mathbf{c}[\lfloor b_i/\tau_2 + o \rfloor] \leftarrow 1$
        else
            $\mathbf{c}[\lfloor (b_i + m_e - m_c)/\tau_2 + o \rfloor] \leftarrow 1$
        end if
    end for

    for $i \leftarrow 1, len(\mathbf{y})$ do                ▷ fill bins corresponding to y-ions
        if $i > x$ then
            $\mathbf{c}[\lfloor y_i/\tau_2 + o \rfloor] \leftarrow 1$
        else
            $\mathbf{c}[\lfloor (y_i + m_e - m_c)/\tau_2 + o \rfloor] \leftarrow 1$
        end if
    end for

    for $j \leftarrow 1, len(\mathbf{c})$ do                ▷ calculate the chain score
        for $i \leftarrow 1, len(\mathbf{e})$ do
            $s \leftarrow s + f(g_i(\mathbf{e}), \mathbf{c}[j])$    ▷ $f(g_i(\mathbf{e}), \mathbf{c}[j])$ is the score function
        end for
    end for

    return $s$
end procedure
```

we assign all peptide chains into different intervals based on their masses, and all peptide chains in the same interval are treated as having the same mass.

Given an experimental spectrum, we calculate the chain scores with respect to all possible peptide chains using Algorithm 1 and assign them to the corresponding intervals. According to Section 2.1, the highest score must come from one of the following situations:

- Two peptide chains are from different intervals: the highest score is equal to the sum of the two top chain scores in two different intervals.
- Two peptide chains are from the same interval: the highest score is equal to two times the top chain score in the interval.

Thus, we only need to keep the top-scored peptide chain and the chain score in each interval during the calculation of $s(\mathbf{e}, \mathbf{t})$. Given a peptide chain in the i-th interval, another peptide chain must be in the interval satisfying $|(i + j) \times w - m(\mathbf{e}) + m_x| \in [-\tau_1, \tau_1]$. Here, we ignore the rounding error because w (i.e. 0.001 Da) is much smaller than τ_1 (e.g. 0.05 Da) in practice. Algorithm 2 shows the pseudo code of identifying cross-linked peptides given a set of peptide chains and an experimental spectrum.

Algorithm 2 Identifying cross-linked peptides with linear time and space complexity. $\{\mathbf{b}_i\}$ is a set of b-ion mass vectors from all peptide chains, $\{\mathbf{y}_i\}$ is a set of y-ion mass vectors from all peptide chains, $\{x_i\}$ is a set of link-site indexed corresponding to all peptide chains, $\{m_{c_i}\}$ is a set of peptide chain masses, $\mathbf{e}$ is a digitized experimental spectrum, m_e is the mass of the experimental spectrum, m_x is the mass of the cross-linker, τ_1 is the precursor mass tolerance, τ_2 is the MS2 m/z tolerance, and o is the offset in digitization.

```
procedure SEARCH($\{\mathbf{b}_i\}, \{\mathbf{y}_i\}, \{x_i\}, \{m_{c_i}\}, \mathbf{e}, m_e, m_x, \tau_1, \tau_2, o$)    ▷ $i \in [1, n]$
    $\mathbf{s}_c \leftarrow vector[\lceil \max(\{m_{C_i}\})/\tau_1 \rceil]$
    $s \leftarrow 0$
    $c_1 \leftarrow -1$
    $c_2 \leftarrow -1$

    for $i \leftarrow 1, |\{m_{c_i}\}|$ do    ▷ calculate chain scores and assign them to ranges
        $s \leftarrow$ ChainScore($\mathbf{b}_i, \mathbf{y}_i, x_i, m_{c_i}, \mathbf{e}, m_e, \tau_2, o$)
        if $\mathbf{s}_c[\lfloor m_{c_i}/\tau_1 \rfloor] < s$ then
            $\mathbf{s}_c[\lfloor m_{c_i}/\tau_1 \rfloor] \leftarrow s$
        end if
    end for

    for $i \leftarrow 1, \lceil \max(\{m_{c_i}\})/0.001 \rceil/2$ do    ▷ pair peptide peptide pairs
        for $j \leftarrow (m_e - m_x - m_{c_i} - \tau_1)/0.001 - i, (m_e - m_x - m_{c_i} + \tau_1)/0.001 - i$ do
            if $\mathbf{s}_c[i] + \mathbf{s}_c[j] > s$ then
                $s \leftarrow \mathbf{s}_c[i] + \mathbf{s}_c[j]$
                $c_1 \leftarrow i$
                $c_2 \leftarrow j$
            end if
        end for
    end for

    return $s, c_1, c_2$
end procedure
```

In the following, we analyze the time and space complexity of Algorithm 2. The time complexity of mass range splitting and peptide chains assignment is

$$\mathcal{O}(n). \tag{6}$$

Without loss of generality, we suppose that the time and space complexity of calculating a chain score is independent of the number of peptides. The time complexity of calculating all chain scores and assigning them to intervals is

$$\mathcal{O}(n). \tag{7}$$

The time complexity of finding pairs of peptide chains, summing chain scores, and keeping the highest-scored pair is

$$\mathcal{O} = \left(\frac{m_{max} - m_{min}}{w} \cdot \frac{\tau_1}{w} \right). \tag{8}$$

Combining Equation (6), (7), and (8), we obtain the total time complexity:

$$\mathcal{O}\left(n + \frac{(m_{max} - m_{min})\tau_1}{w^2} \right) = \mathcal{O}(n). \tag{9}$$

Because m_{max}, m_{min}, τ_1, and w are pre-fixed and independent of the database size, the total time complexity is linear with respect to the number of peptide chains. It is easy to see that the space complexity is also linear: $\mathcal{O}(n + (m_{max} - m_{min})/w)$.

2.3 The Work-Flow of ECL 2.0

Figure 2 shows the work-flow of ECL 2.0. It takes a data file and a protein database file as inputs. After digitizing spectra, digesting protein sequences, and *in silicon* fragmenting peptide sequences, it calculates chain scores using Algorithm 1. Once is has obtained each spectrum's chain scores, the method pairs peptide chains and finds the highest-scored pair using Algorithm 2. It then calculates an e-value using the linear tail-fit method [24]. Finally, it estimates each spectrum's q-value according to the estimated e-values. There are three kinds of peptide-spectrum matches (PSMs): the first contains two peptide chains from the target database, the second contains two peptide chains from the decoy database, and the third contains one peptide chain from the target database and another peptide chain from the decoy database. Thus, we can estimate the false discovery rate (FDR) using [43, 45] $FDR(s) = \frac{f(s)-d(s)}{t(s)}$, where s is an e-value, $t(s)$ is the number of the first kind of PSMs whose e-values are smaller than or equal to s, $d(s)$ is the number of the second kind of PSMs whose e-values are smaller than or equal to s, and $f(s)$ is the number of the third kind of PSMs whose e-values are smaller than or equal to s. Finally, we convert the FDR to a q-value [39] using $q(t) = \min_{s \geq t} FDR(s)$, where t is a threshold.

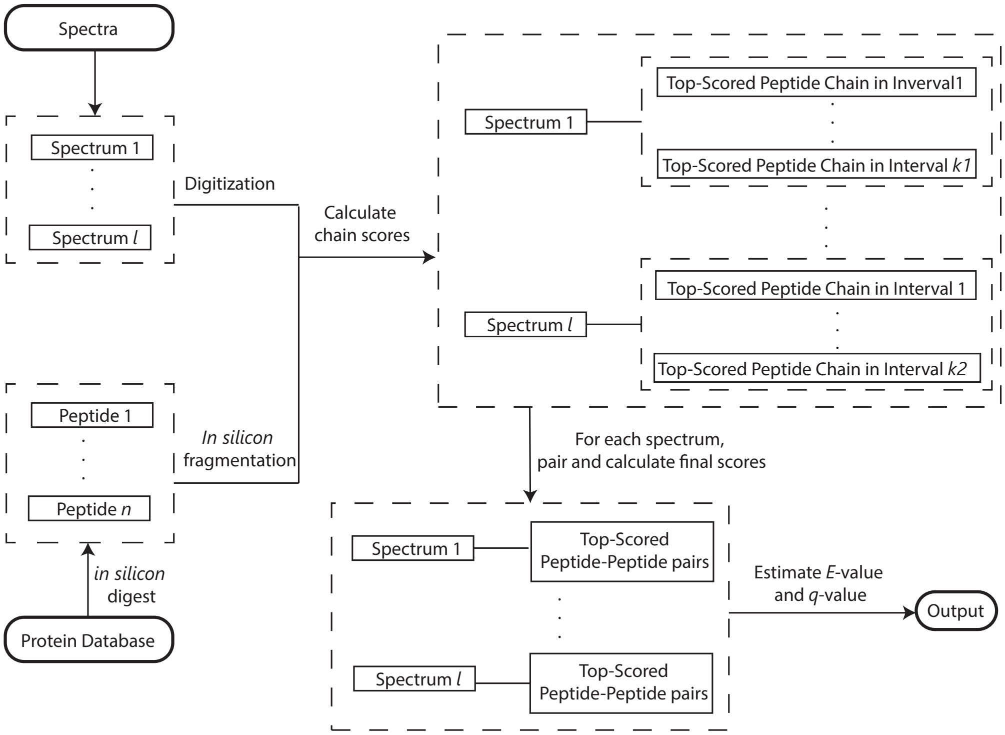

Spectra
Spectrum 1
Spectrum l
Digitization
Calculate chain scores
Spectrum 1
Top-Scored Peptide Chain in Inverval1
Top-Scored Peptide Chain in Interval k1
Spectrum l
Top-Scored Peptide Chain in Interval 1
Top-Scored Peptide Chain in Interval k2
Peptide 1
Peptide n
In silicon fragmentation
in silicon digest
Protein Database
For each spectrum, pair and calculate final scores
Spectrum 1
Top-Scored Peptide-Peptide pairs
Spectrum l
Top-Scored Peptide-Peptide pairs
Estimate E-value and q-value
Output

Fig. 2. The workflow of ECL 2.0.

3 Experiments

We use the public data in Makowski *et al.* [26] to demonstrate the performance of ECL 2.0. There are 10 data files from the cross-linking of human tissues, containing about 3×10^5 MS2 spectra in total. Please refer to Makowski *et al.* [26] for the details of the sample preparation and data acquisition. The paper reported 19 cross-linked proteins with high confidence. In order to demonstrate the effect of database size, we generate 6 databases of various sizes:

1. The first database contains the 19 from Makowski *et al.* [26] proteins and 50 randomly selected proteins.
2. The second database contains all proteins in the first database and an additional 50 randomly selected proteins.
3. The third database contains all proteins in the second database and adds an additional 100 randomly selected proteins.
4. The fourth database contains all proteins in the third database and adds an additional 300 randomly selected proteins.
5. The fifth database contains all proteins in the fourth database and adds an additional 500 randomly selected proteins.
6. The last database contains all proteins in the fifth database and adds an additional 4000 randomly selected proteins.

The randomly selected proteins are from *Arabidopsis thaliana*, while the samples are from human tissue. Thus, without considering decoy sequences, we have 6 databases whose protein numbers are 69, 119, 219, 519, 1019, and 5019, respectively.

We use StavroX (Version 3.6.0), pLink (Version 1.23), ProteinProspector (Version 5.17.1), Kojak (Version 1.4.3), and ECL 2.0 (Version 2.3.1) to search these data files against 6 databases, respectively. The precursor mass tolerance is 10 ppm, and the MS2 m/z tolerance is 0.02 Da. The allowed maximum missed cleavage is 2. The allowed precursor masses are from 1000 Da to 12000 Da and the allowed peptide chain lengths are from 5 amino acids to 50 amino acids. All 5 tools use the target-decoy strategy [4, 43, 45] to estimate the q-value. We use a q-value ≤ 0.05 as the cut-off. StavroX and pLink provide q-values for their own results. We estimate q-values for the results of ProteinProspector, Kojak, and ECL 2.0, respectively. With the cut-off results, the PSMs from the cross-linking of the 19 proteins are treated as true positive PSMs, and the PSMs containing at least one peptide chain from the randomly selected proteins are treated as false positive PSMs.

3.1 Sensitivity and False Discovery Proportion

We summarize the true positive PSMs and false positive PSMs identified by those 5 tools. StavroX cannot handle a database with more than 119 proteins and pLink cannot handle a database with more than 1019 proteins. Figure 3 shows the bar plots of the true and false positive PSMs. The blue bars denote the true positive PSMs and the orange bars denote the false positive PSMs. The numbers at the top of the bar are the corresponding FDPs: $FDP = \frac{S}{\max(V+S,1)}$, where V is the number of true positives and S is the number of false positives. Because the correctness of a PSM is decided from its peptide chains' protein type, the calculated FDP is the lower bound of the underlying true FDP.

Figure 3 shows that ECL 2.0 has the lowest FDP. Its sensitivity is similar to that of the other state-of-the-art tools. Figure 3 also shows that the FDP increases and the number of true positive PSMs decreases as the size of database increases. This phenomenon is more significant for the tools using non-exhaustive search (i.e. pLink, ProteinProspector, and Kojak).

3.2 Running Speed

ECL 2.0 has a fast speed. We calculate each tool's average running time with respect to different database sizes. StavroX, pLink, Kojak, and ECL 2.0 are run on a standard PC with an Intel Core i7-2600 CPU (3.40 GHz, 8 cores) and 32 GB memory. StavroX and pLink don't support multi-thread computing. Kojak and ECL 2.0 support multi-thread computing. Thus, Kojak and ECL 2.0 are run with 8 cores. ProteinProspector is run on the authors' web server. Table 1 shows the average

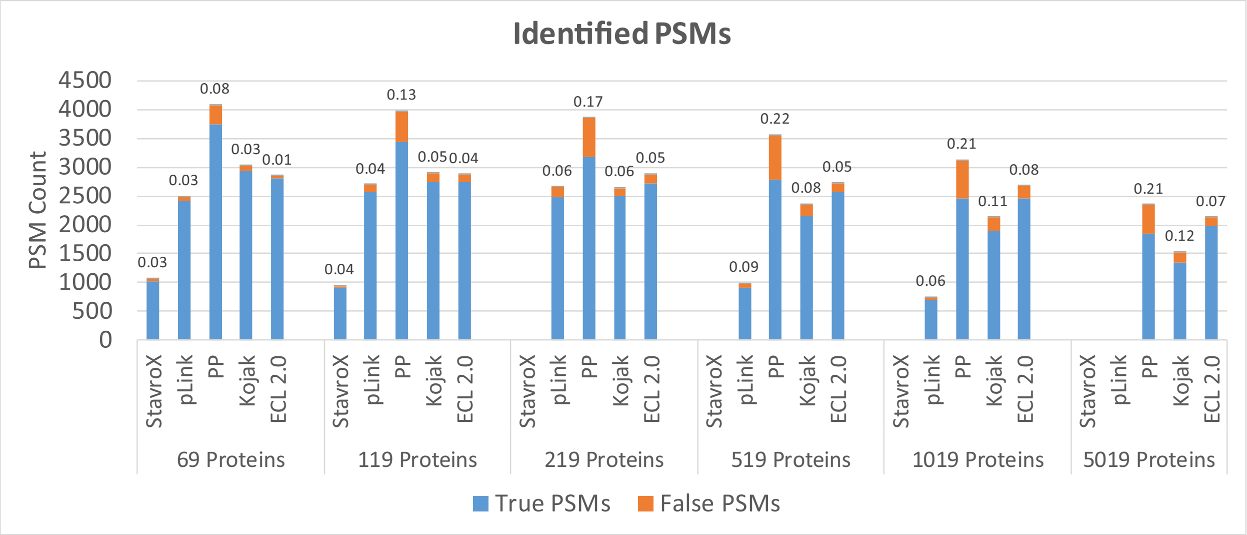


Fig. 3. Bar plots showing identified true and false positive PSMs. 6 bar plots correspond to the results of searching 6 databases. Without considering decoy proteins, there are 69, 119, 219, 519, 1019, and 5019 proteins in 6 databases, respectively. The blue bars denote true positive PSMs and the orange bars denote false positive PSMs. The numbers on top of each bar are the corresponding FDPs. “PP” stands for ProteinProspector. StavroX cannot handle a database with more than 119 proteins and pLink cannot handle a database with more than 1019 proteins, so the corresponding parts are empty.

running time in hours. ECL 2.0 is much faster than pLink, StavroX, and ProteinProspector. It is slower than Kojak. In order to show that ECL 2.0 does have a linear time complexity, we plot the average running time with respect to different numbers of peptide chains (including decoy sequences) in a database in Figure 4. For comparison, we also plot the average running time of ECL 1.0 (version 1.1.0) which has a quadratic time complexity. The Figure shows the advantage of linear complexity clearly.

Table 1. The average running time of 5 tools with respect to different database sizes. The unit is hours. StavroX cannot handle a database larger than 119 proteins and pLink cannot handle a database larger than 1019 proteins. “NA” stands for not applicable. “PP” stands for ProteinProspector.

	Target Protein Number in the Database					
	69	119	219	519	1019	5000
StavroX	5.11	10.13	NA	NA	NA	NA
pLink	30.25	6.13	10.40	19.46	33.62	NA
PP	0.74	0.78	1.52	1.07	1.32	7.77
Kojak	0.05	0.06	0.10	0.10	0.15	0.51
ECL 2.0	0.67	0.73	0.88	1.25	1.85	6.42

4 Discussions

In this paper, we demonstrated that it is feasible to exhaustively search all possible peptide-peptide pairs for cross-linked peptides identification with a linear time and space complexity. Given a data file with tens of thousands of MS2 spectra, it can finish the analysis using a big database in a few

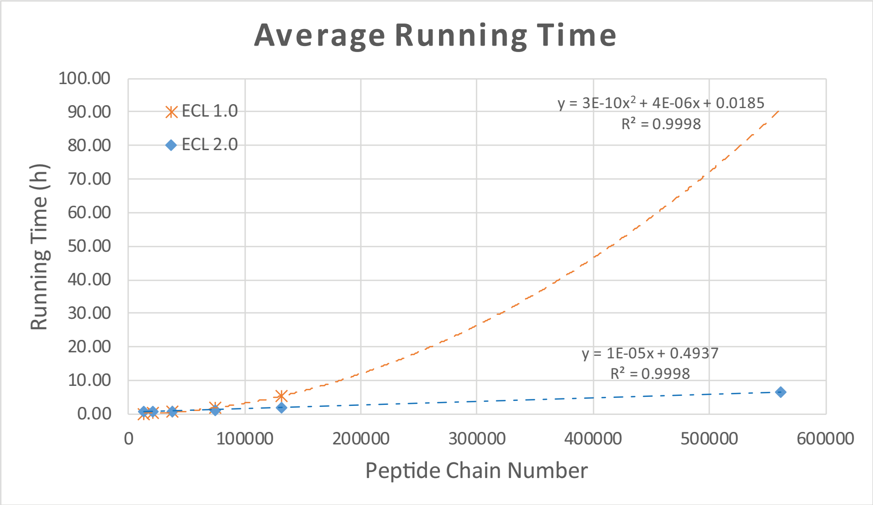


Fig. 4. A plot showing average running time with respect to different numbers of peptide chains. The orange crosses are the observed running time of ECL 1.0 and the orange dashed line is the quadratic regression line. The blue dots are the observed running time of ECL 2.0 and the blue dashed line is the linear regression line. It also shows the equations and R^2 values. Under the 561227 peptide chains setting, ECL 1.0 will need about 90 hours to finish the analysis according to the estimation, while ECL 2.0 only needs about 6.4 hours. Decoy sequences are included.

hours. To our knowledge, this is the first tool that can exhaustively search about 5000 proteins in such a short time. Such a speedup is possible due to the following factors:

1) Taking advantage of an additive score function: A final score can be split into two chain scores, which facilitates further optimizing the algorithm.
2) Digitizing the whole mass range and assigning peptide chains to digitized intervals: With such a digitization, only one score is kept for each interval. This reduces the time complexity greatly. Given a peptide chain, the time complexity of finding another peptide chain having the highest chain score is $\mathcal{O}(n \log(n))$ with the help of quicksort [2]. Thus, the bottleneck lies in sorting.
3) Achieving a constant time complexity in summing up chain scores by fixing the number of digitized intervals.

   By fixing the number of ranges, we eliminate the bottleneck and the total time complexity reduces to $\mathcal{O}(n)$. Actually, using counting sort can achieve $\mathcal{O}(n)$ time complexity [2] without fixing the number of ranges. However, its space complexity is quite high, which will be an issue in cross-linked peptides identification.

Acknowledgments. This work is partially supported by a theme-based project T12-402/13N from the Research Grants Council (RGC) of the Hong Kong S.A.R. government, internal grant VPRGO15EG01 from HKUST, grants 16101114 and 661613 from the General Research Fund (GRF) of the Hong Kong S.A.R. government, and grant 31370315 from the National Natural Science Foundation of China (NSFC). We thank Meng Wang and Jiaan Dai for valuable discussions.

Competing Financial Interests The authors declare that they have no competing financial interests.

Bibliography

[1] Chu, F., Shan, S.O., Moustakas, D.T., Alber, F., Egea, P.F., Stroud, R.M., Walter, P., Burlingame, A.L.: Unraveling the interface of signal recognition particle and its receptor by using chemical cross-linking and tandem mass spectrometry. Proceedings of the National Academy of Sciences of the United States of America 101(47), 16454–16459 (2004)

[2] Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, Third Edition. The MIT Press (2009)

[3] Du, X., Chowdhury, S.M., Manes, N.P., Wu, S., Mayer, M.U., Adkins, J.N., Anderson, G.A., Smith, R.D.: Xlink-Identifier: an automated data analysis platform for confident identifications of chemically cross-linked peptides using tandem mass spectrometry. Journal of Proteome Research 10(3), 923–931 (2011)

[4] Elias, J., Gygi, S.: Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. Nature Methods 4, 207–214 (2007)

[5] Eng, J.K., Jahan, T.A., Hoopmann, M.R.: Comet: An open-source ms/ms sequence database search tool. Proteomics 13(1), 22–24 (2013)

[6] Götze, M., Pettelkau, J., Fritzsche, R., Ihling, C.H., Schäfer, M., Sinz, A.: Automated assignment of MS/MS cleavable cross-links in protein 3d-structure analysis. Journal of The American Society for Mass Spectrometry 26(1), 83–97 (2015)

[7] Götze, M., Pettelkau, J., Schaks, S., Bosse, K., Ihling, C.H., Krauth, F., Fritzsche, R., Kühn, U., Sinz, A.: StavroX-a software for analyzing crosslinked products in protein interaction studies. Journal of the American Society for Mass Spectrometry 23(1), 76–87 (2012)

[8] Greber, B.J., Boehringer, D., Leitner, A., Bieri, P., Voigts-Hoffmann, F., Erzberger, J.P., Leibundgut, M., Aebersold, R., Ban, N.: Architecture of the large subunit of the mammalian mitochondrial ribosome. Nature 505(7484), 515–519 (2014)

[9] Herzog, F., Kahraman, A., Boehringer, D., Mak, R., Bracher, A., Walzthoeni, T., Leitner, A., Beck, M., Hartl, F.U., Ban, N., et al.: Structural probing of a protein phosphatase 2A network by chemical cross-linking and mass spectrometry. Science 337(6100), 1348–1352 (2012)

[10] Holding, A.N., Lamers, M.H., Stephens, E., Skehel, J.M.: Hekate: software suite for the mass spectrometric analysis and three-dimensional visualization of cross-linked protein samples. Journal of Proteome Research 12(12), 5923–5933 (2013)

[11] Hoopmann, M.R., Zelter, A., Johnson, R.S., Riffle, M., MacCoss, M.J., Davis, T.N., Moritz, R.L.: Kojak: efficient analysis of chemically cross-linked protein complexes. Journal of Proteome Research 14(5), 2190–2198 (2015)

[12] Howbert, J.J., Noble, W.S.: Computing exact p-values for a cross-correlation shotgun proteomics score function. Molecular & Cellular Proteomics 13(9), 2467–2479 (2014)

[13] Ihling, C., Schmidt, A., Kalkhof, S., Schulz, D.M., Stingl, C., Mechtler, K., Haack, M., Beck-Sickinger, A.G., Cooper, D.M., Sinz, A.: Isotope-labeled cross-linkers and Fourier transform ion cyclotron resonance mass spectrometry for structural analysis of a protein/peptide complex. Journal of the American Society for Mass Spectrometry 17(8), 1100–1113 (2006)

[14] Jimmy, K., Ashley, L., John, R.: An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. Journal of the American Society for Mass Spectrometry 5(11), 976 – 989 (1994)

[15] Kaake, R.M., Wang, X., Burke, A., Yu, C., Kandur, W., Yang, Y., Novtisky, E.J., Second, T., Duan, J., Kao, A., et al.: A new in vivo cross-linking mass spectrometry platform to define protein–protein interactions in living cells. Molecular & Cellular Proteomics 13(12), 3533–3543 (2014)

[16] Kall, L., Canterbury, J.D., Weston, J., Noble, W.S., MacCoss, M.J.: Semi-supervised learning for peptide identification from shotgun proteomics datasets. Nature Methods 4(11), 923–925 (2007)

[17] Kao, A., Chiu, C.l., Vellucci, D., Yang, Y., Patel, V.R., Guan, S., Randall, A., Baldi, P., Rychnovsky, S.D., Huang, L.: Development of a novel cross-linking strategy for fast and accurate identification of cross-linked peptides of protein complexes. Molecular & Cellular Proteomics 10, M110.002212 (2010)

[18] Kim, S., Pevzner, P.A.: MS-GF+ makes progress towards a universal database search tool for proteomics. Nature Communications 5 (2014)

[19] Koning, L.J., Kasper, P.T., Back, J.W., Nessen, M.A., Vanrobaeys, F., Beeumen, J., Gherardi, E., Koster, C.G., Jong, L.: Computer-assisted mass spectrometric analysis of naturally occurring and artificially introduced cross-links in proteins and protein complexes. FEBS Journal 273(2), 281–291 (2006)

[20] Lam, H., Deutsch, E., Eddes, J., Eng, J., King, N., Stein, S., Aebersold, R.: Development and validation of a spectral library searching method for peptide identification from MS/MS. Proteomics 7, 655–667 (2007)

[21] Lam, H., Deutsch, E.W., Eddes, J.S., Eng, J.K., Stein, S.E., Aebersold, R.: Building consensus spectral libraries for peptide identification in proteomics. Nature Methods 5(10), 873–875 (2008)

[22] Lee, Y., Lackner, L., Nunnari, J., Phinney, B.: Shotgun cross-linking analysis for studying quaternary and tertiary protein structures. Journal of Proteome Research 6(10), 3908–3917 (2007)

[23] Liu, F., Rijkers, D.T., Post, H., Heck, A.J.: Proteome-wide profiling of protein assemblies by cross-linking mass spectrometry. Nature Methods 12, 1179–1184 (2015)

[24] Lynn, A.J., Chalkley, R.J., Baker, P.R., Segal, M.R., Burlingame, A.L.: Protein Prospector and ways of calculating expectation values. In: Proceedings of the 54th ASMS Conference on Mass Spectrometry, Seattle. p. 351 (2006)

[25] Maiolica, A., Cittaro, D., Borsotti, D., Sennels, L., Ciferri, C., Tarricone, C., Musacchio, A., Rappsilber, J.: Structural analysis of multi-protein complexes by cross-linking, mass spectrometry and database searching. Molecular & Cellular Proteomics 6, 2200–2211 (2007)

[26] Makowski, M.M., Willems, E., Jansen, P.W., Vermeulen, M.: Cross-linking immunoprecipitation-MS (xIP-MS): Topological analysis of chromatin-associated protein complexes using single affinity purification. Molecular & Cellular Proteomics 15(3), 854–865 (2016)

[27] McIlwain, S., Draghicescu, P., Singh, P., Goodlett, D., Noble, W.: Detecting cross-linked peptides by searching against a database of cross-linked peptide pairs. Journal of Proteome Research 9(5), 2488–2495 (2010)

[28] Mueller-Planitz, F.: Crossfinder-assisted mapping of protein crosslinks formed by site-specifically incorporated crosslinkers. Bioinformatics p. btv083 (2015)

[29] Nadeau, O.W., Wyckoff, G.J., Paschall, J.E., Artigues, A., Sage, J., Villar, M.T., Carlson, G.M.: CrossSearch, a user-friendly search engine for detecting chemically cross-linked peptides in conjugated proteins. Molecular & Cellular Proteomics 7(4), 739–749 (2008)

[30] Nguyen, V.Q., Ranjan, A., Stengel, F., Wei, D., Aebersold, R., Wu, C., Leschziner, A.E.: Molecular architecture of the ATP-dependent chromatin-remodeling complex SWR1. Cell 154(6), 1220–1231 (2013)

[31] Panchaud, A., Singh, P., Shaffer, S., Goodlett, D.: xComb: A cross-linked peptide database approach to protein-protein interaction analysis. Journal of Proteome Research 9(5), 2508–2515 (2010)

[32] Petrotchenko, E.V., Borchers, C.H.: ICC-CLASS: isotopically-coded cleavable crosslinking analysis software suite. BMC Bioinformatics 11(1), 64 (2010)

[33] Petrotchenko, E.V., Serpa, J.J., Borchers, C.H.: An isotopically coded cid-cleavable biotinylated cross-linker for structural proteomics. Molecular & Cellular Proteomics 10(2), M110–001420 (2011)

[34] Politis, A., Stengel, F., Hall, Z., Hernández, H., Leitner, A., Walzthoeni, T., Robinson, C.V., Aebersold, R.: A mass spectrometry-based hybrid method for structural modeling of protein complexes. Nature Methods 11(4), 403–406 (2014)

[35] Rinner, O., Seebacher, J., Walzthoeni, T., Mueller, L., Beck, M., Schmidt, A., Mueller, M., Aebersold, R.: Identification of cross-linked peptides from large sequence databases. Nature Methods 5(4), 315–318 (2008)

[36] Schilling, B., Row, R., Gibsonb, B., Guo, X., Young, M.: MS2Assign: Automated assignment and nomenclature of tandem mass spectra of chemically crosslinked peptides. Journal of The American Society for Mass Spectrometry 14, 834–850 (2003)

[37] Singh, P., Shaffer, S., Scherl, A., Holman, C., Pfuetzner, R., Freeman, T.L., Miller, S., Hernandez, P., Appel, R., Goodlett, D.: Characterization of protein cross-links via mass spectrometry and an open-modification search strategy. Analytical Chemistry 80(22), 8799–8806 (2008)

[38] Spivak, M., Bereman, M., MacCoss, M., Noble, W.: Learning score function parameters for improved spectrum identification in tandem mass spectrometry experiments. Journal of Proteome Research 11(9), 4499–4508 (2012)

[39] Storey, J.D., Tibshirani, R.: Statistical significance for genomewide studies. Proceedings of the National Academy of Sciences 100(16), 9440–9445 (2003)

[40] Tang, Y., Chen, Y., Lichti, C., Hall, R., Raney, K., Jennings, S.: CLPM: A cross-linked peptide mapping algorithm for mass spectrometric analysis. BMC Bioinformatics 6, S9 (2005)

[41] Tanner, S., Shu, H., Frank, A., Wang, L.C., Zandi, E., Mumby, M., Pevzner, P.A., Bafna, V.: InsPecT: identification of posttranslationally modified peptides from tandem mass spectra. Analytical Chemistry 77(14), 4626–4639 (2005)

[42] Trnka, M.J., Baker, P.R., Robinson, P.J., Burlingame, A., Chalkley, R.J.: Matching cross-linked peptide spectra: only as good as the worse identification. Molecular & Cellular Proteomics 13(2), 420–434 (2014)

[43] Walzthoeni, T., Claassen, M., Leitner, A., Herzog, F., Bohn, S., Förster, F., Beck, M., Aebersold, R.: False discovery rate estimation for cross-linked peptides identified by mass spectrometry. Nature Methods 9(9), 901–903 (2012)

[44] Wang, L.H., Li, D.Q., Fu, Y., Wang, H.P., Zhang, J.F., Yuan, Z.F., Sun, R.X., Zeng, R., He, S.M., Gao, W.: pFind 2.0: a software package for peptide and protein identification via tandem mass spectrometry. Rapid Communications in Mass Spectrometry 21(18), 2985–2991 (2007)

[45] Yang, B., Wu, Y.J., Zhu, M., Fan, S.B., Lin, J., Zhang, K., Li, S., Chi, H., Li, Y.X., Chen, H.F., et al.: Identification of cross-linked peptides from complex samples. Nature Methods 9(9), 904–906 (2012)

[46] Young, M., Tang, N., Hempel, J., Oshiro, C., Taylor, E., Kuntz, I., Gibson, B., Dollinger, G.: High throughput protein fold identification by using experimental constraints derived from intramolecular cross-links and mass spectrometry. Proceedings of the National Academy of Sciences of the United States of America 97, 5802–5806 (2000)

[47] Yu, E.T., Hawkins, A., Kuntz, I.D., Rahn, L.A., Rothfuss, A., Sale, K., Young, M.M., Yang, C.L., Pancerella, C.M., Fabris, D.: The collaboratory for MS3D: a new cyberinfrastructure for the structural elucidation of biological macromolecules and their assemblies using mass spectrometry-based approaches. Journal of Proteome Research 7(11), 4848–4857 (2008)

[48] Yu, F., Li, N., Yu, W.: ECL: an exhaustive search tool for the identification of cross-linked peptides using whole database. BMC Bioinformatics 17(1), 1–8 (2016)

[49] Zhu, X., Yu, F., Yang, Z., Liu, S., Dai, C., Lu, X., Liu, C., Yu, W., Li, N.: In planta chemical cross-linking and mass spectrometry analysis of protein structure and interaction in Arabidopsis. Proteomics 16(13), 1915–1927 (2016)

Appendix

How To Use ECL 2.0

ECL 2.0 can be downloaded at `http://bioinformatics.ust.hk/ecl2.0.html`. It is a command line program. Users can run it under Windows and Linux. The command is

```
java −Xmx25g −jar ECL2.0.jar parameter.def data
```

where "parameter.def" is a parameter file containing all parameters, "data" is a mass spectra file in mzXML format. There are five outputs: "data.intra.target.csv", "data.intra.decoy.csv", "data.inter.target.csv", "data.inter.decoy.csv", and "ECL2.0.log". The first file contains PSMs from target intra protein cross-linked peptides; the second file contains PSMs from decoy intra protein cross-linked peptides; the third file contains PSMs from target inter protein cross-linked peptides; the fourth file contains PSMs from decoy inter protein cross-linked peptides; and the fifth file is a log. The first four files can be opened with Excel. Columns in the first four files are arranged as follows:

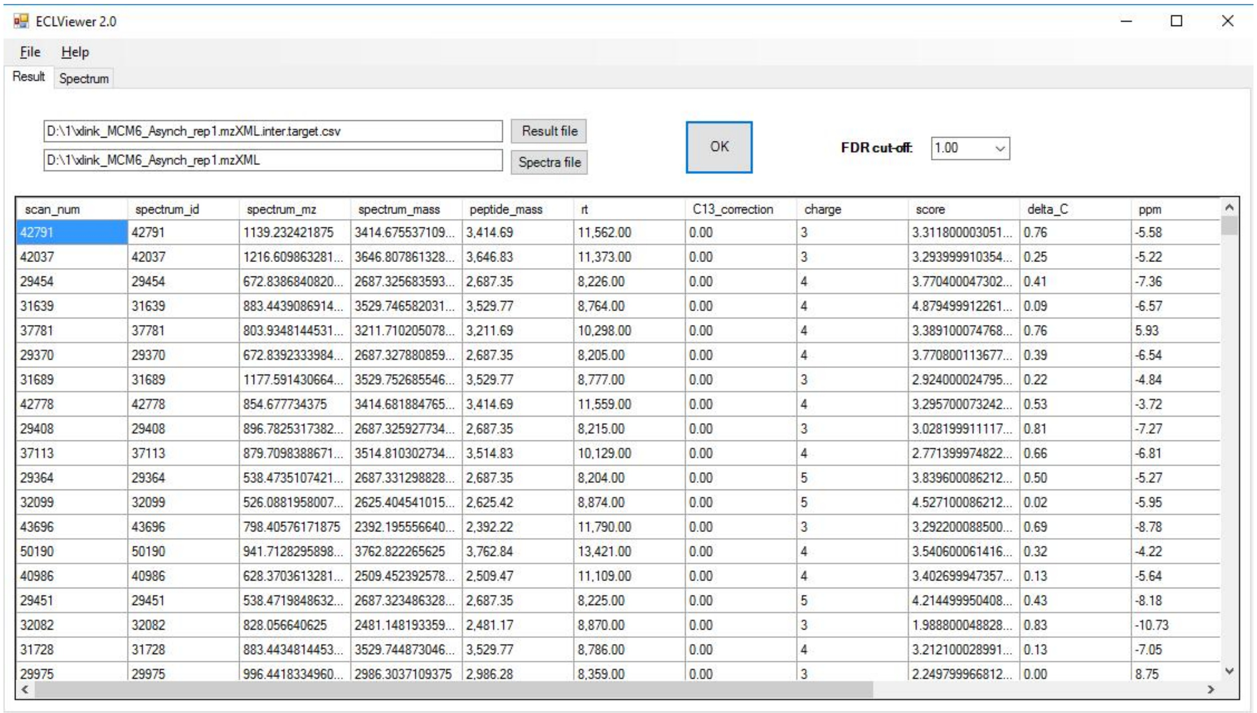
1. Scan number.
2. Spectrum id.
3. Spectrum m/z value.
4. Spectrum mass.
5. Theoretical mass.
6. Retention time.
7. C13 correction number.
8. Charge.
9. Score.
10. One minus the ratio between the second ranked score and the top ranked score.
11. The derivation between the spectrum's precursor mass and theoretical precursor mass.
12. Peptide sequence.
13. Protein identifiers.
14. The annotation of the first linked protein.
15. The annotation of the second linked protein.
16. e-value.
17. q-value (decoy files don't have this column).

ECLViewer 2.0

In order to visualize identified MS2 spectra, we develop a software named ECLViewer 2.0. It has a graphical user interface. Figure 5 shows one example of ECLViewer 2.0. The user specifies the result file and the spectra file, and clicks "OK". ECLViewer 2.0 will load two files and display the results. After the user double-clicks a scan number in the result page, ECLViewer 2.0 will jump to the spectrum page (Figure 6) showing annotated spectrum. The name of buttons and labels are self-explaining.

Supplementary Experiments

Most existing tools pre-filter peptide chains before peptide-peptide pairing: pLink keeps top 500 peptide chains, ProteinProspector keeps top 1000 peptide chains, and Kojak keeps top 250 peptide chains. Peptide chains whose ranks are lower than the threshold are filtered out and won't be searched. We summarize peptide chains' ranks from the true positive PSMs identified by ECL 2.0.



D:\1\xlink_MCM6_Asynch_rep1.mzXML.inter.target.csv

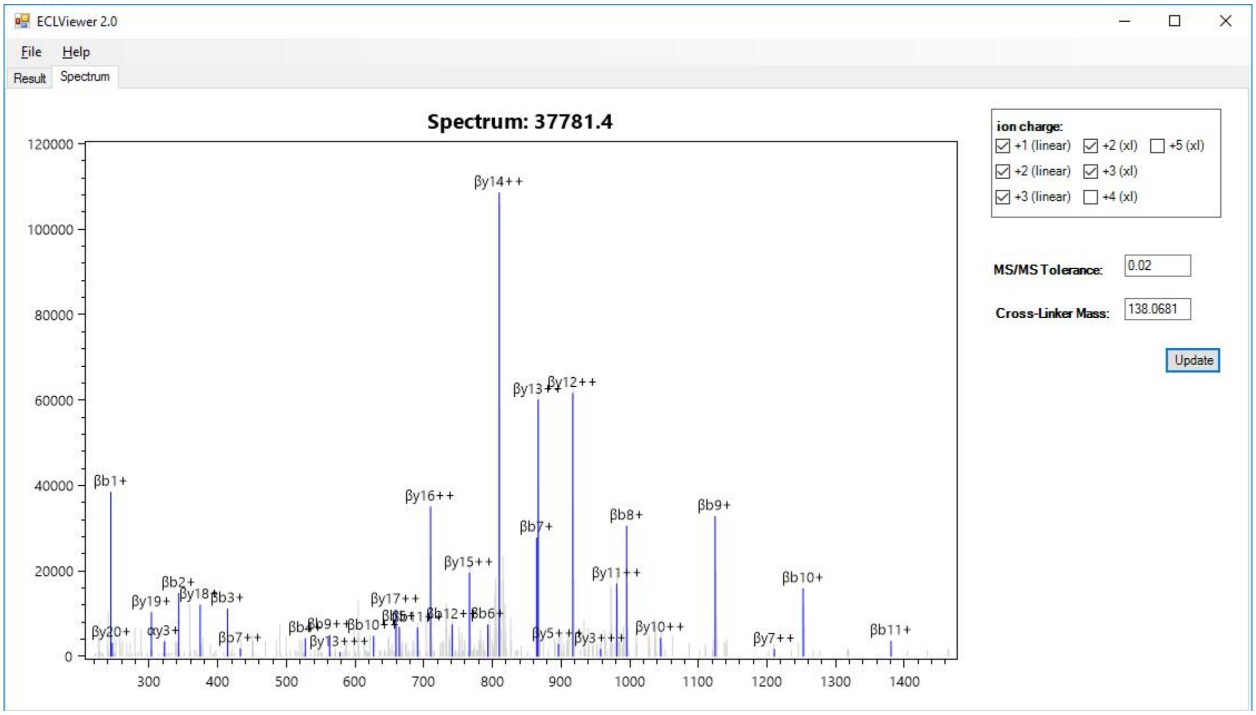
D:\1\xlink_MCM6_Asynch_rep1.mzXML

FDR cut-off: 1.00

scan_num	spectrum_id	spectrum_mz	spectrum_mass	peptide_mass	rt	C13_correction	charge	score	delta_C	ppm
42791	42791	1139.232421875	3414.675537109...	3,414.69	11,562.00	0.00	3	3.311800003051...	0.76	-5.58
42037	42037	1216.609863281...	3646.807861328...	3,646.83	11,373.00	0.00	3	3.293999910354...	0.25	-5.22
29454	29454	672.8386840820...	2687.325683593...	2,687.35	8,226.00	0.00	4	3.770400047302...	0.41	-7.36
31639	31639	883.4439086914...	3529.746582031...	3,529.77	8,764.00	0.00	4	4.879499912261...	0.09	-6.57
37781	37781	803.9348144531...	3211.710205078...	3,211.69	10,298.00	0.00	4	3.389100074768...	0.76	5.93
29370	29370	672.8392333984...	2687.327880859...	2,687.35	8,205.00	0.00	4	3.770800113677...	0.39	-6.54
31689	31689	1177.591430664...	3529.752685546...	3,529.77	8,777.00	0.00	3	2.924000024795...	0.22	-4.84
42778	42778	854.677734375	3414.681884765...	3,414.69	11,559.00	0.00	4	3.295700073242...	0.53	-3.72
29408	29408	896.7825317382...	2687.325927734...	2,687.35	8,215.00	0.00	3	3.028199911117...	0.81	-7.27
37113	37113	879.7098388671...	3514.810302734...	3,514.83	10,129.00	0.00	4	2.771399974822...	0.66	-6.81
29364	29364	538.4735107421...	2687.331298828...	2,687.35	8,204.00	0.00	5	3.839600086212...	0.50	-5.27
32099	32099	526.0881958007...	2625.404541015...	2,625.42	8,874.00	0.00	5	4.527100086212...	0.02	-5.95
43696	43696	798.40576171875	2392.195556640...	2,392.22	11,790.00	0.00	3	3.292200088500...	0.69	-8.78
50190	50190	941.7128295898...	3762.822265625	3,762.84	13,421.00	0.00	4	3.540600061416...	0.32	-4.22
40986	40986	628.3703613281...	2509.452392578...	2,509.47	11,109.00	0.00	4	3.402699947357...	0.13	-5.64
29451	29451	538.4719848632...	2687.323486328...	2,687.35	8,225.00	0.00	5	4.214499950408...	0.43	-8.18
32082	32082	828.056640625	2481.148193359...	2,481.17	8,870.00	0.00	3	1.988800048828...	0.83	-10.73
31728	31728	883.4434814453...	3529.744873046...	3,529.77	8,786.00	0.00	4	3.212100028991...	0.13	-7.05
29975	29975	996.4418334960...	2986.3037109375	2,986.28	8,359.00	0.00	3	2.249799966812...	0.00	8.75

Fig. 5. The result page of ECLViewer 2.0.

Because a PSM wouldn't be identified as long as there was one peptide chain being filtered out, summarizing the lower ranked chain for each PSM is enough. Figure 7 shows the trend of missed findings with different chain rank thresholds and database sizes. Those PSMs who have a peptide chain ranked lower than the threshold are treated as missed findings. Under the top 250 setting, there are around 30% $\sim$ 40% missed findings, which is consistent with what we claimed in our former paper [48]. It is easy to see that the pre-filtering strategy misses a significant amount of findings, especially when the database is large.



ion charge:
☑ +1 (linear) ☑ +2 (xl) □ +5 (xl)
☑ +2 (linear) ☑ +3 (xl)
☑ +3 (linear) □ +4 (xl)

MS/MS Tolerance: 0.02

Cross-Linker Mass: 138.0681

Fig. 6. The spectrum page of ECLViewer 2.0. It can be activated by double-clicking a scan number in the result page.

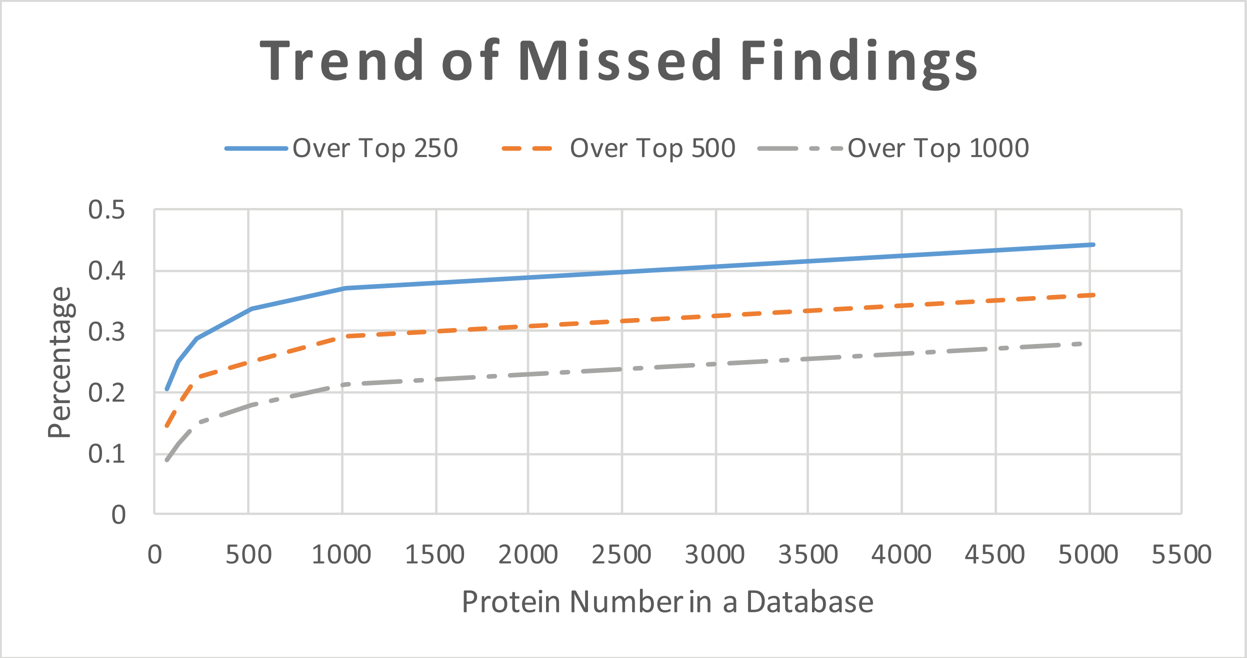


Fig. 7. The trend of missed finding with different chain rank threshold criteria and database sizes. Three chain rank thresholds are used: top 250, top 500, and top 1000. 6 different database sizes are used. Without considering decoy proteins, there are 69, 119, 219, 519, 1019, and 5019 proteins in 6 database, respectively. Those PSMs who have a peptide chain ranked lower than the threshold are treated as missed findings.