SEQUEST

2020.06

김현우

Introduction

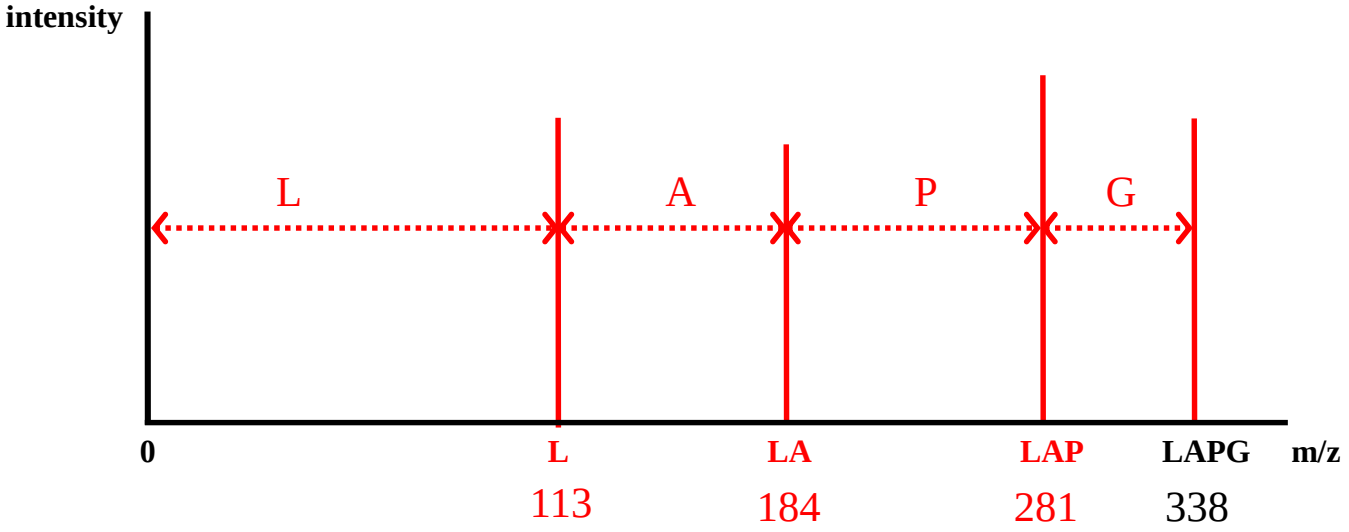
- **Protein**
  - An amino acid sequence
    - MEMEKEFEQIDKSGSWAAIYQDIRHEASDFPCRVAKLPKNK NRNRYRDVSPFDHSRKLHQEDNDYINASLIKMEEAQRSYIL TQ….

- **Peptide**
  - Substring of the protein
    - LPKNKNRNRYRDVSPFDHSR

Introduction

- **Tandem mass spectrometry**
  - Peptide LAPG
    - L, AGP
    - LA, PG
    - LAP, G

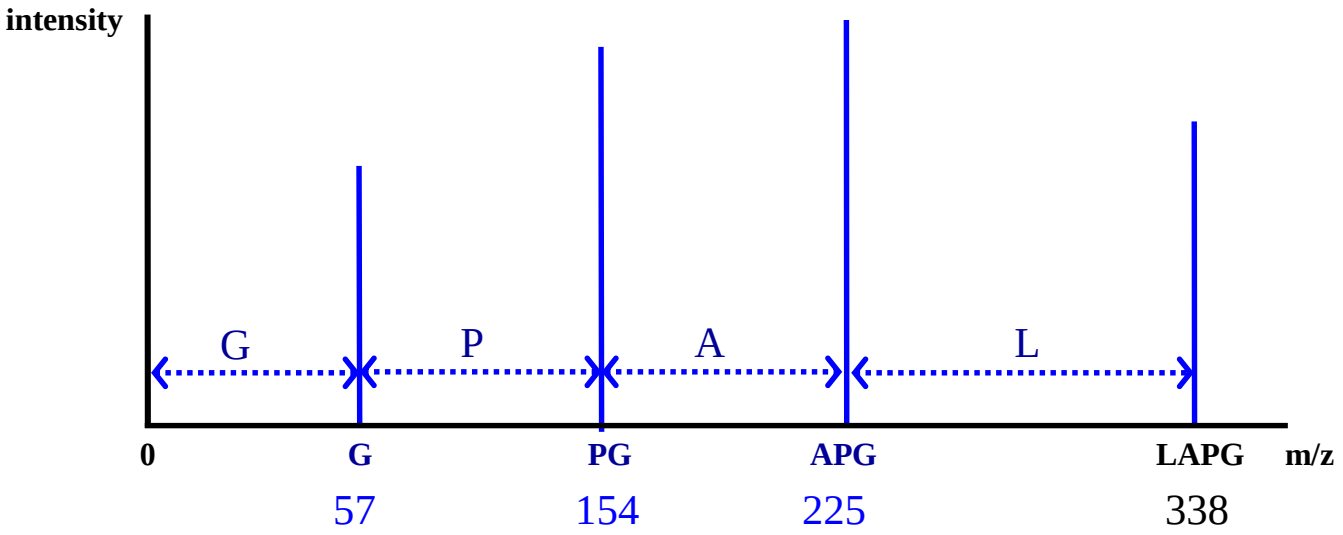
L	113
A	71
P	97
G	57



Introduction

- **Tandem mass spectrometry**
  - Peptide LAPG
    - L, AGP
    - LA, PG
    - LAP, G

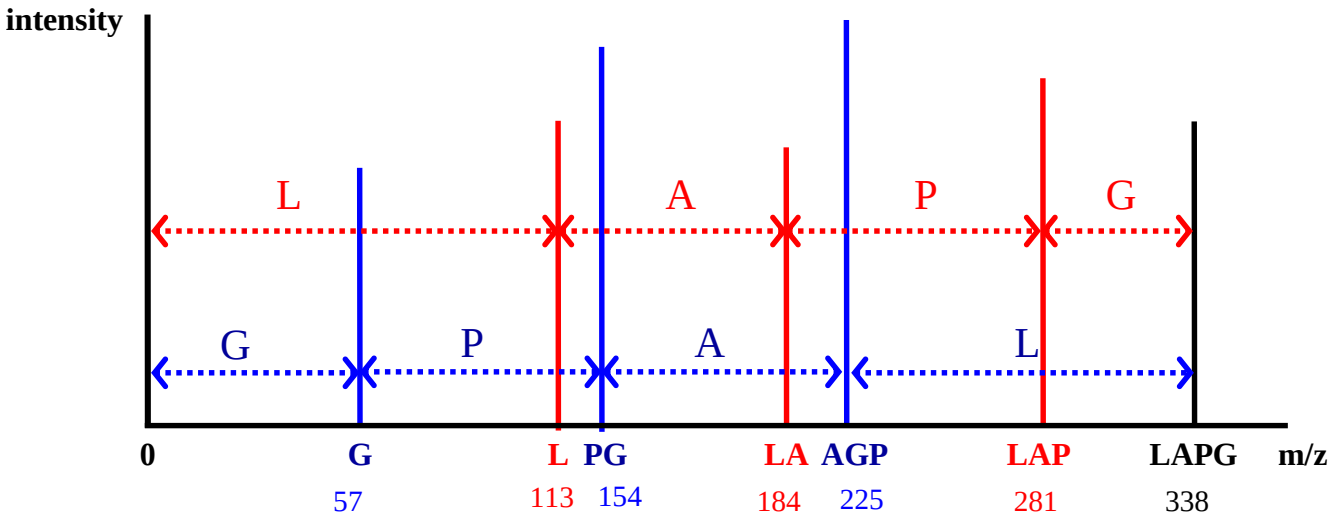
L	113
A	71
P	97
G	57



Introduction

- **Tandem mass spectrometry**
  - Peptide LAPG
    - L, AGP
    - LA, PG
    - LAP, G

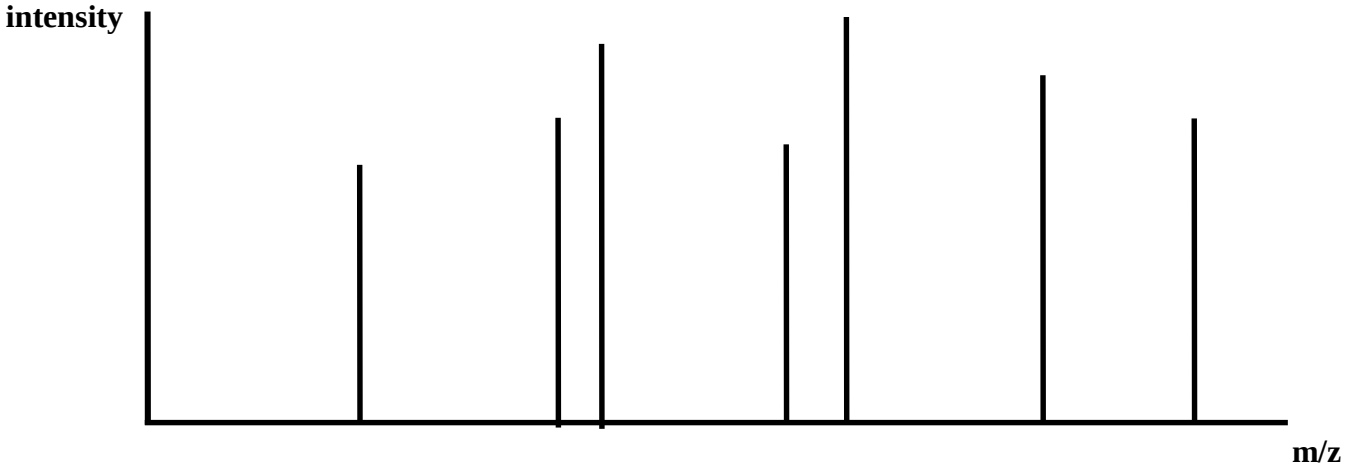
L	113
A	71
P	97
G	57



Introduction

- **Tandem mass spectrometry**
  - Peptide LAPG
    - L, AGP
    - LA, PG
    - LAP, G

L	113
A	71
P	97
G	57



Introduction

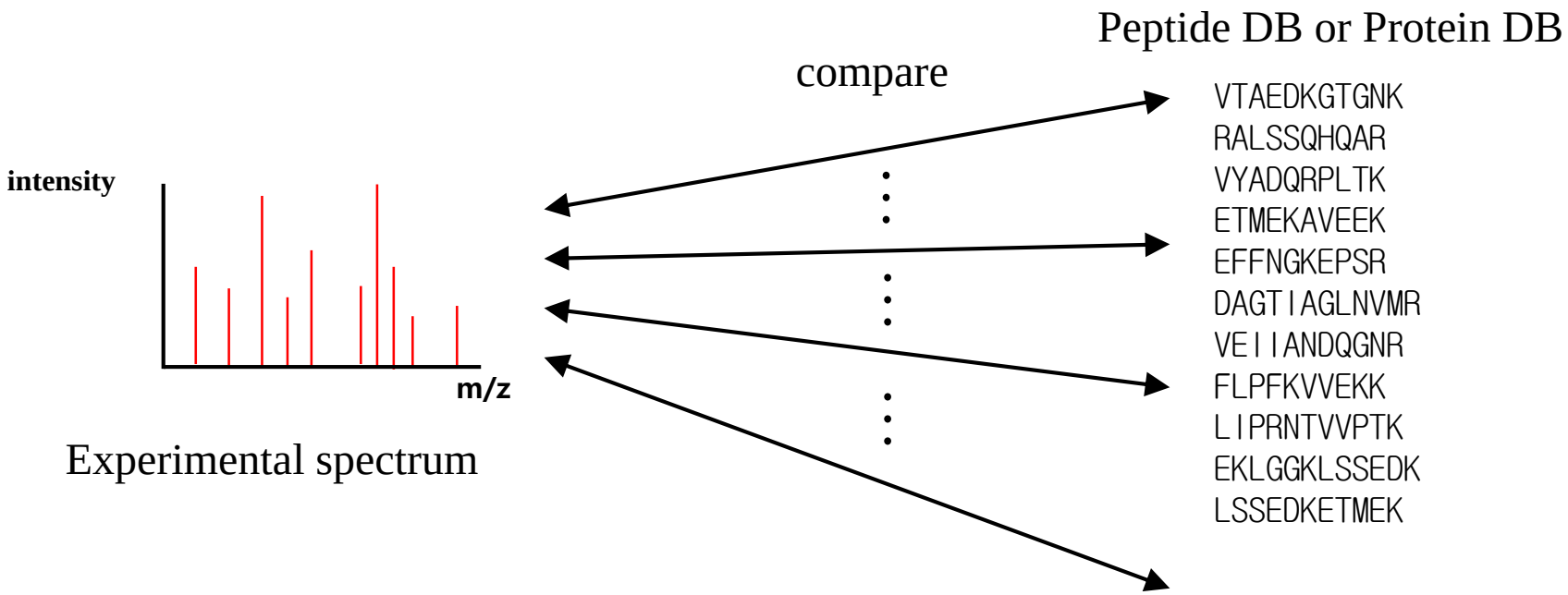
- Peptide identification is an important problem in proteomics
- One of the most popular scoring schemes for peptide identification is XCorr (cross-correlation)
- Since calculating XCorr is computationally intensive, a lot of efforts have been made to develop fast XCorr engines

Introduction

- **Cross-correlation algorithm**
  - Time complexity is $\Theta(n\log n)$ ➔ SEQUEST
    - Eng, J.K. *et al*., An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database., *J. am. Soc. Mass Spectrom.*, 1994
  - Time complexity is $\Theta(\tau n)$ ➔ Crux
    - Eng, J.K. *et al*., A Fast SEQUEST Cross Correlation Algorithm., *J. of Proteome Res.*, 2008
    - Mcllwain, S. *et al*., Crux: rapid open source protein tandem mass spectrometry analysis., *J. of Proteome Res.*, 2014
  - Time complexity is $\Theta(\tau + n)$ ➔ Tide, Comet
    - Jo, H. *et al*., Computing Cross-Correlation of SEQUEST in Linear Time., *RECOMB Satellite Conference on Computational Proteomics*, 2010
    - Diament, B.J. *et al*., Faster SEQUEST searching for peptide identification from tandem mass spectra., *J. of Proteome Res.*, 2011
    - Eng, J.K. et al., Comet: an open-source MS/MS sequence database search tool., *Proteomics*, 2013

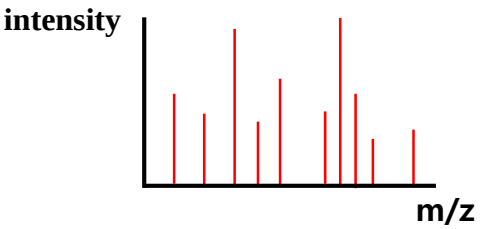
Introduction

- **Peptide identification**



Peptide identification tools

- **Basic algorithm for peptide identification**



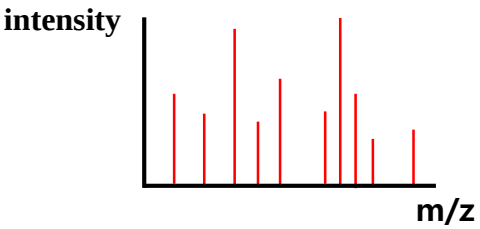
Experimental spectrum

Peptide DB

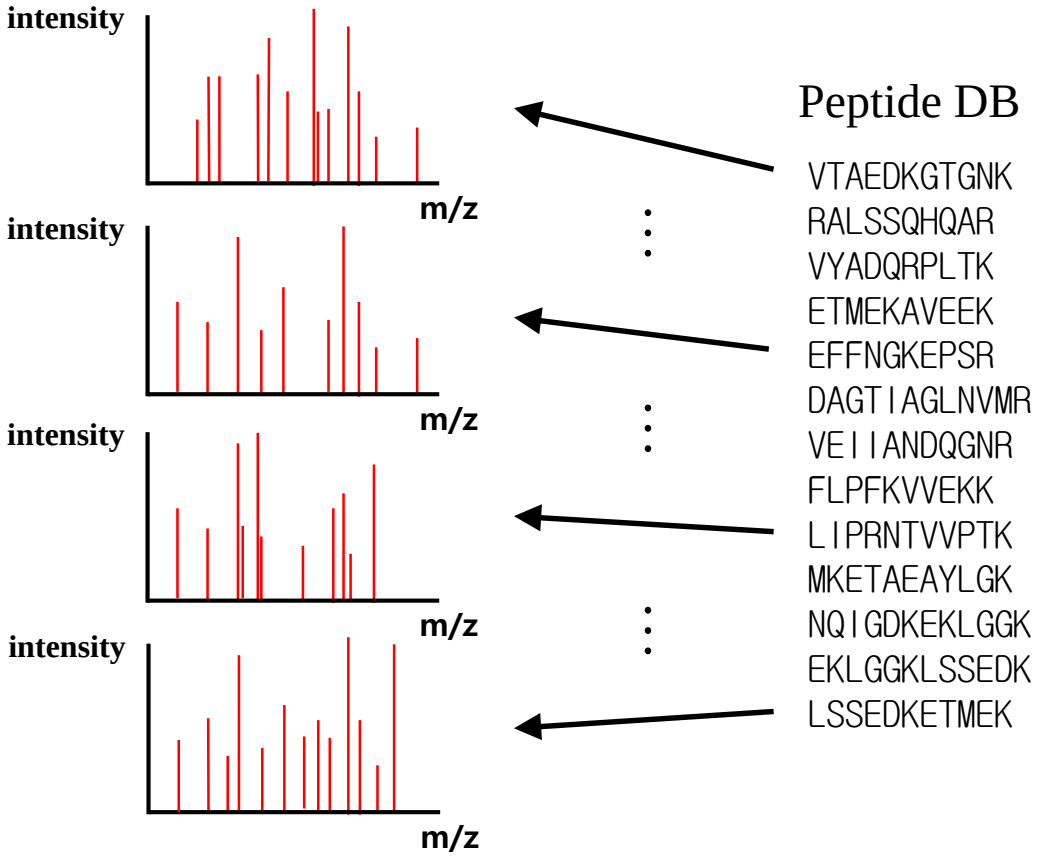
VTAEDKGTGNK
RALSSQHQAR
VYADQRPLTK
ETMEKAVEEK
EFFNGKEPSR
DAGTIAGLNVMR
VEIIANDQGNR
FLPFKVVEKK
LIPRNTVVPTK
MKETAEAYLGK
NQIGDKEKLGGK
EKLGGKLSSEDK
LSSEDKETMEK

Peptide identification tools

- **Basic algorithm for peptide identification**



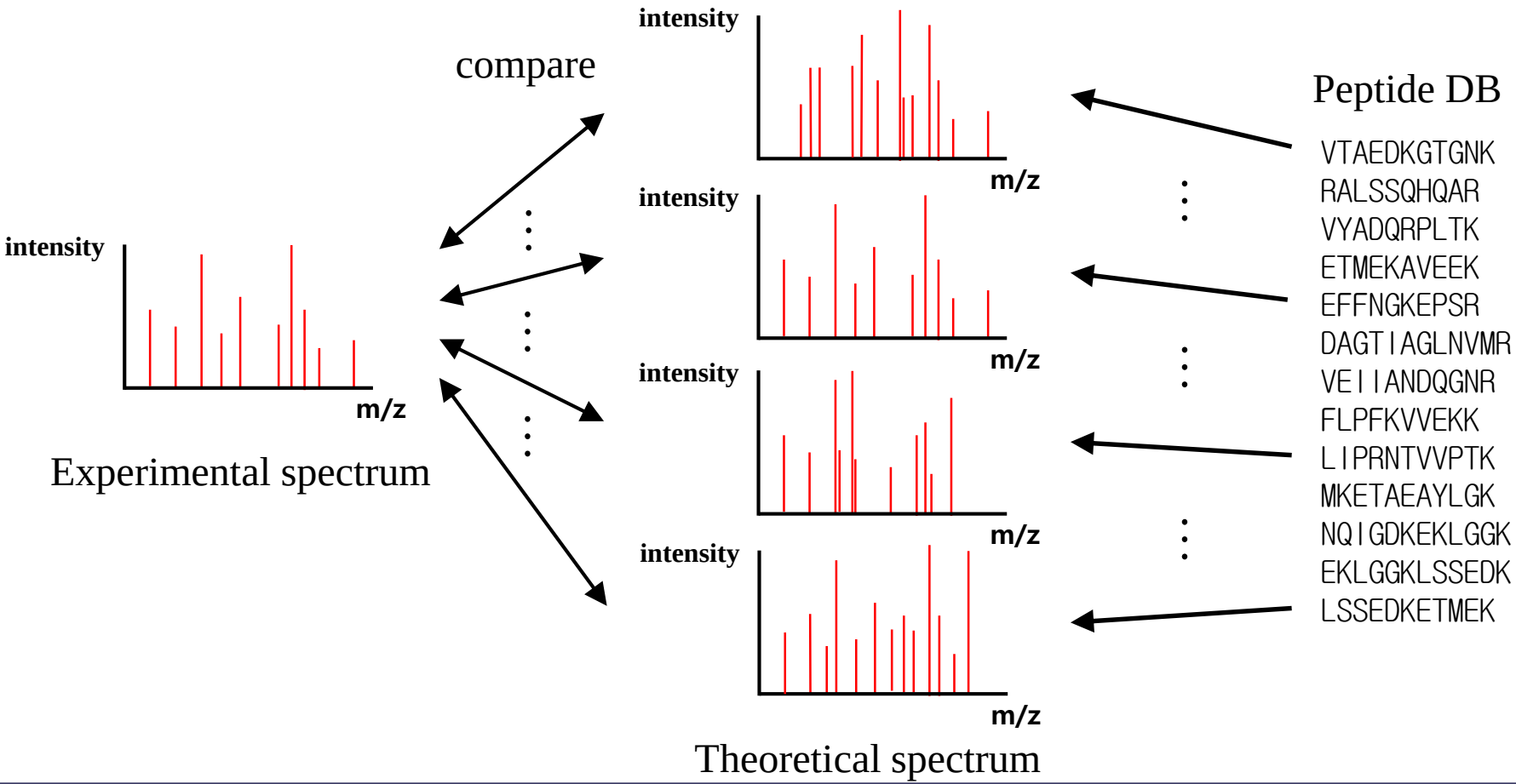
Experimental spectrum



Theoretical spectrum

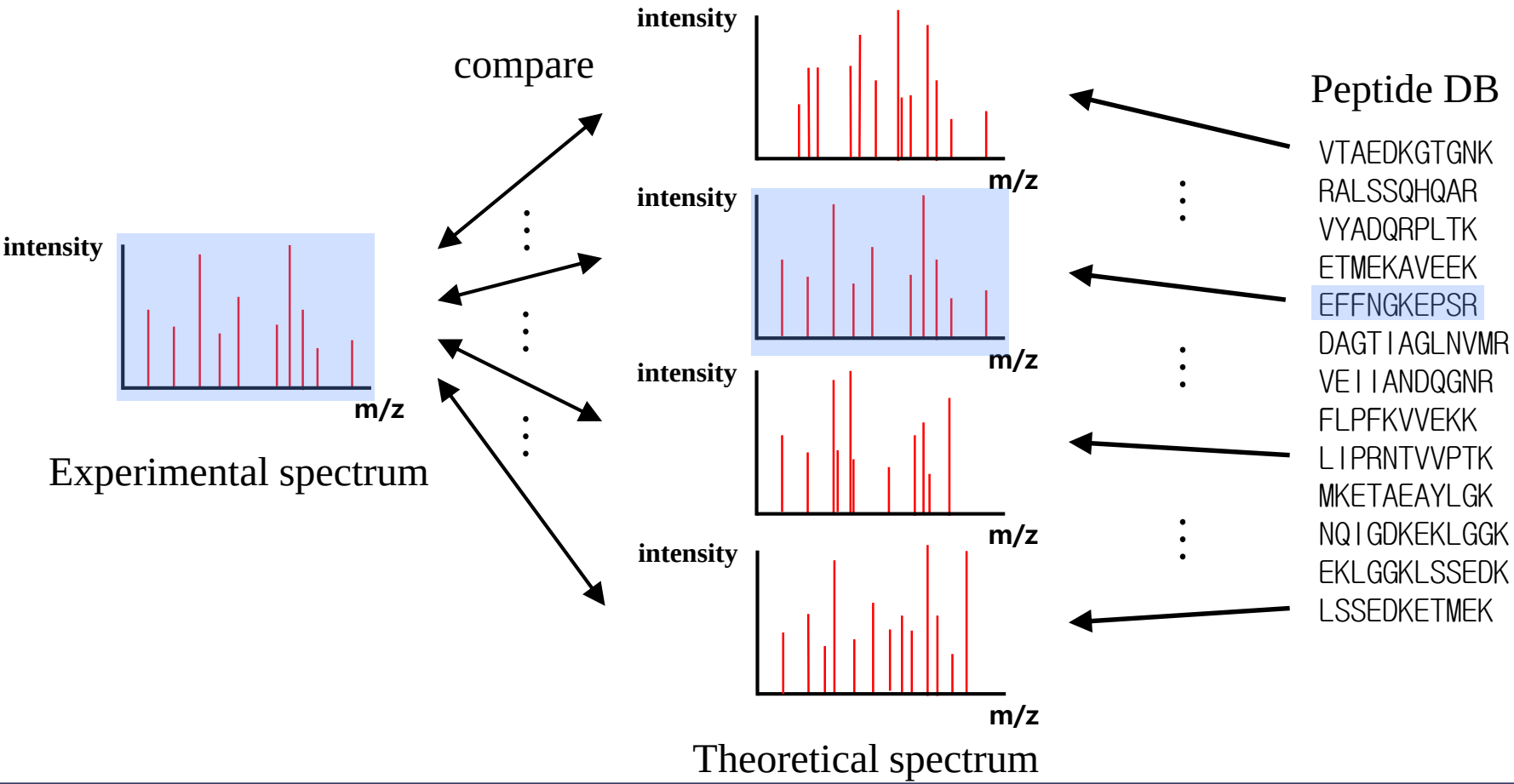
Peptide identification tools

- **Basic algorithm for peptide identification**



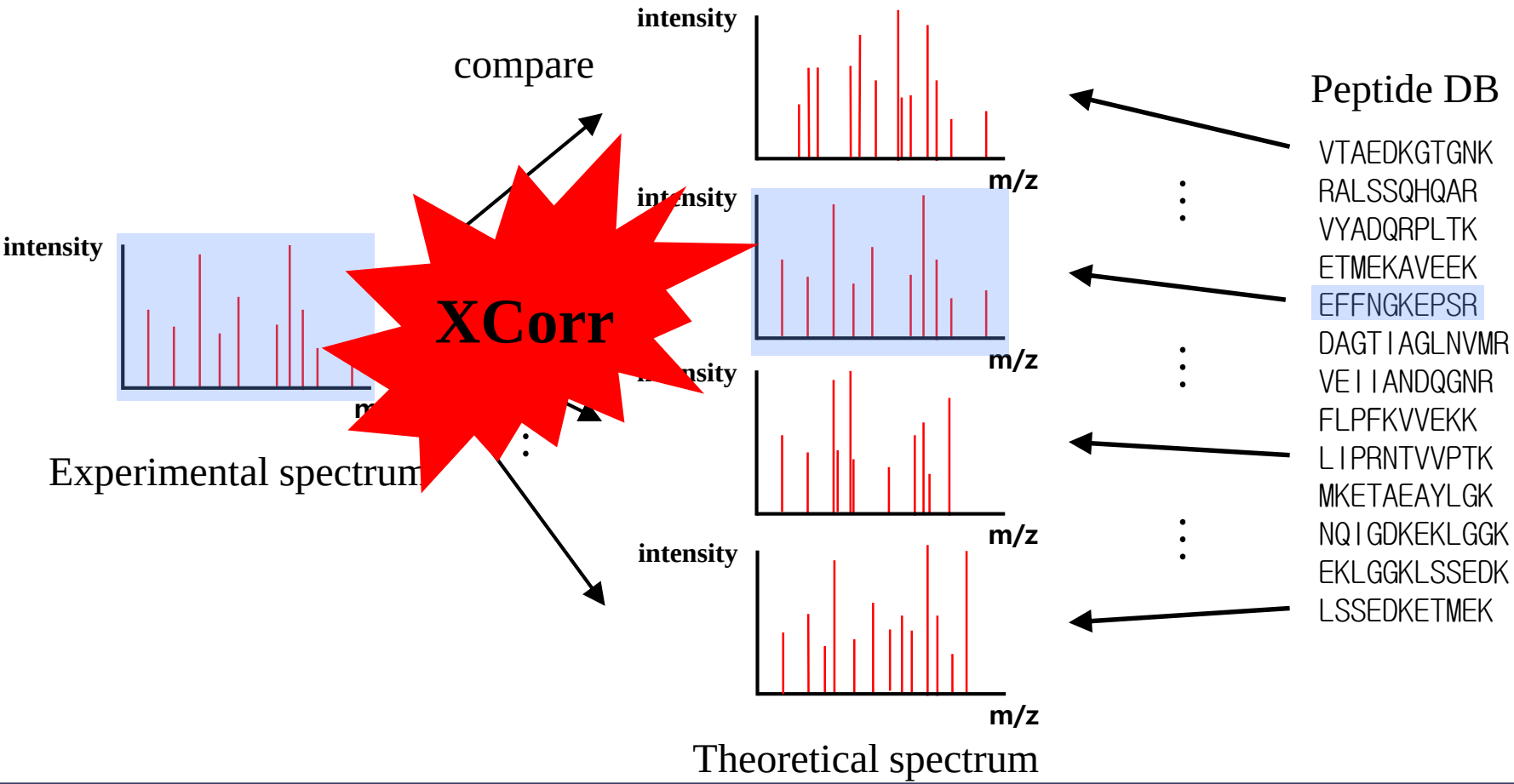
Peptide identification tools

- **Basic algorithm for peptide identification**



Peptide identification tools

- **Basic algorithm for peptide identification**



SEQUEST

- **SEQUEST**

1. Tandem mass spectrometry data reduction
2. Database search
3. Filtering by a simple scoring method
4. Cross-correlation

Tamdem mass spectrometry data reduction

- Fragment ion mass-to-charge ratios are converted to the rounded nominal values(nearest integer).
  - We realizes an increase in computational speed.

- A 10-u window around the precursor ion is removed.
  - To eliminate possible matches of a predicted fragment ion to the mass-to charge ratio of the precursor ion.

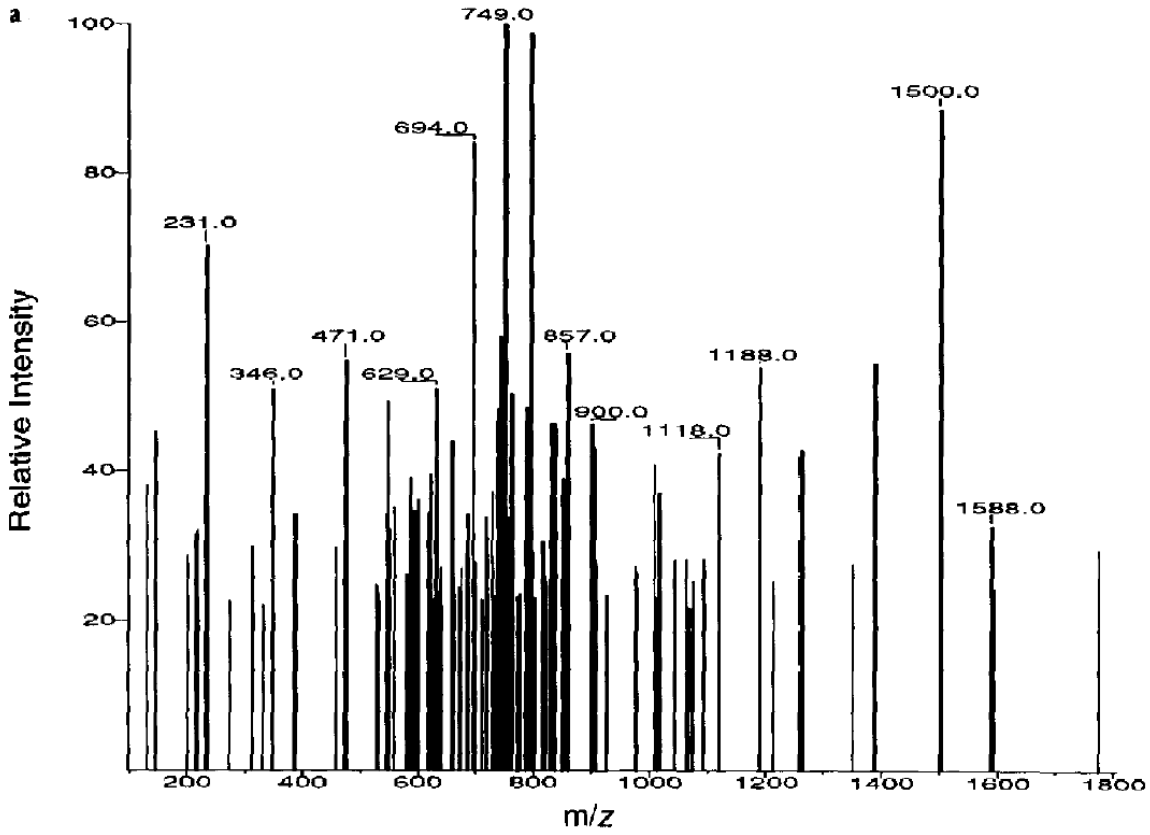
Tamdem mass spectrometry data reduction

- To eliminate noise from the spectrum and to reduce the number of ions to be considered.
  - All but the 200 most abundant ions are removed
  - The remaining ions are renormalized to 100.
  - The abundances of fragment ions within ±1 u of each other are equalized to the higher value.

- Immonium ions are frequently observed when the amino acids His, Met, Trp, Tyr, or Phe are present in the spectrum.
  - The presence of these ions in the mass spectrum is noted during this preprocessing step.

Tamdem mass spectrometry data reduction

- Data for the peptide DLRSWTAADTAAQISQ obtained from the ion $[M + 2H]^{+2}$ at m/z 868.



Database search

- Select each peptide from DB whose mass is within a specified mass tolerance.
- Mass tolerance: ± 0.05% or a minimum of 1 u
- Chemical modifications to amino acids can be considered by changing the amino acid masses.

Filtering by a simple scoring method

- Select 500 high scoring peptides.

- $S_P = (\sum i_m) n_i (1+\beta)(1+\rho) \,/\, n_t$
  - $\sum i_m$: the sum of matched intensities
  - n_i : the number of matched peaks
  - β : continuity of an ion series
  - ρ : immonium ion peaks
  - n_t : the total number of amino acids

Cross-correlation

1. Generation of theoretical spectrum for every peptide.

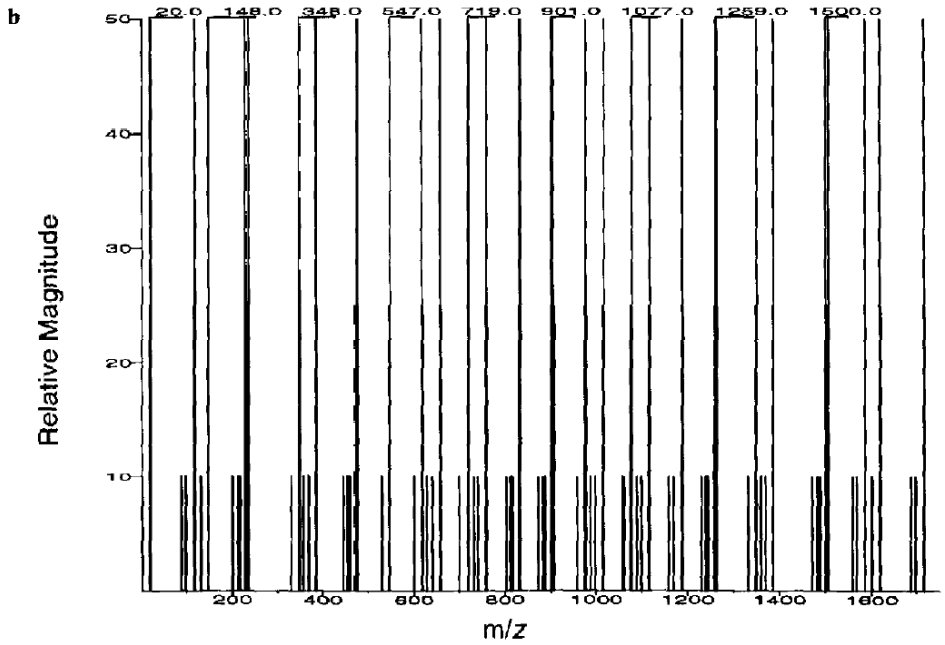
2. Preprocessing the experimental spectrum.

3. Compute the cross correlation score between every pair of theoretical spectrum and the experimental spectrum.

4. Select the most promising peptide using the score.

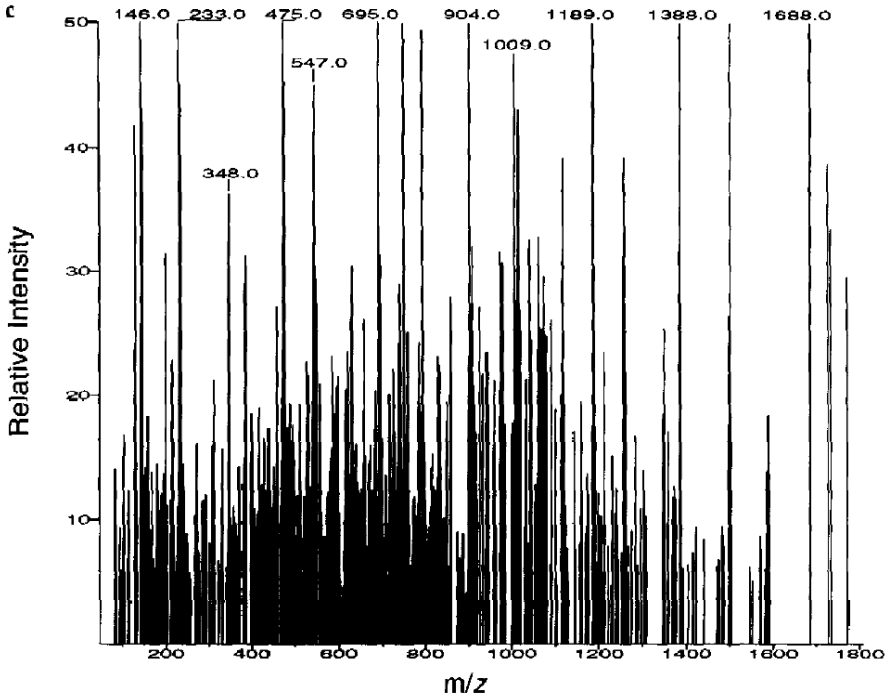
Cross-correlation

1. Generation of theoretical spectrum for every peptide.
   - 50 : b-ions, y-ions
   - 25 : within ±1 of b-ions and y-ions
   - 10 : neutral loss of ammonia, water, and carbon monoxide and their ±1



Cross-correlation

2. Preprocessing the experimental spectrum.
   - Remove the mass-to-charge ratio for the precursor ion.
   - Dividing the original spectrum into 10 equal regions and normalizing the ions in each region to a value of 50.



Cross-correlation

3. Compute the cross-correlation score
   - Compute a scoring function.

$$R_\tau = \sum_{i=0}^{n-1} x[i]\,y[i+\tau]$$

   - A discrete correlation of two real functions *x* and *y* is one member of the discrete Fourier transform pair

$$R_\tau \leftrightarrow X_\tau Y_\tau^*$$

     where *X* and *Y* are the discrete Fourier transforms of *x* and *y* the *Y** denotes complex conjugation.

Cross-correlation

3. Compute the cross-correlation score
   - The score
     = (The value at $\tau = 0$) –
     (the mean of the cross-correlation over $-75 < \tau < 75$)

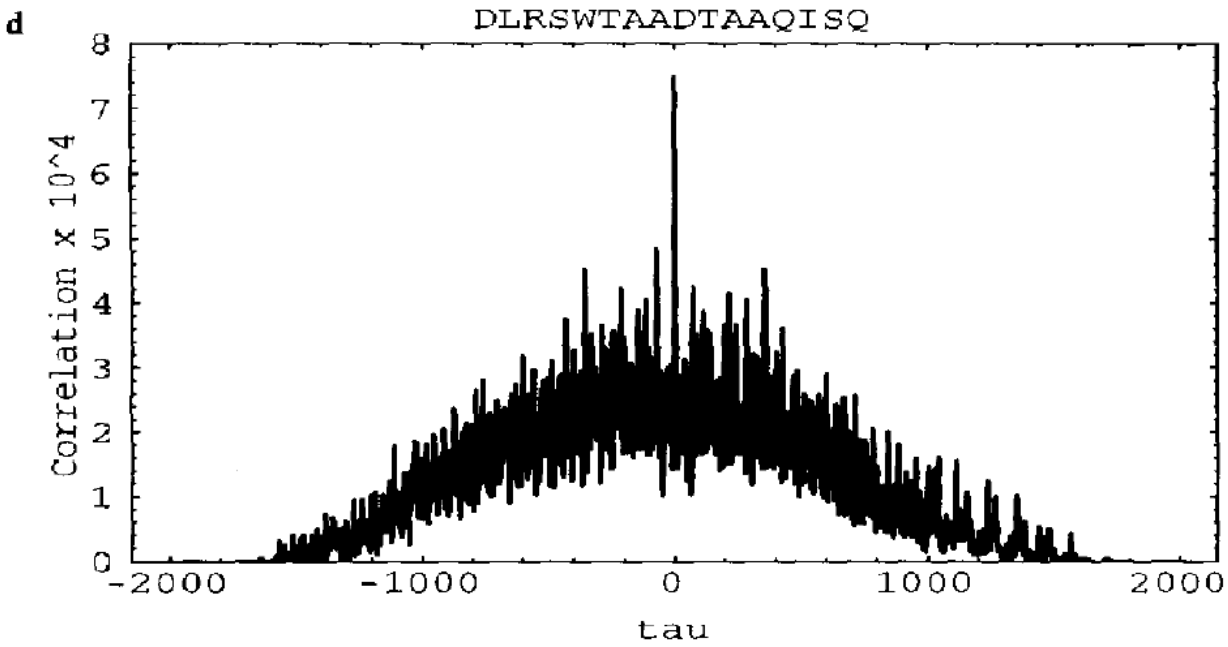


Figure 2. (*Continued*)

Cross-correlation

4. Select the most promising peptide using the score.
   - The scores are normalized to 1.0 and termed *Cn*.
   - If [(C_n for the highest) – (C_n for the second highest)] is larger than 0.1($\Delta C_n > 0.1$), it is usually correct.

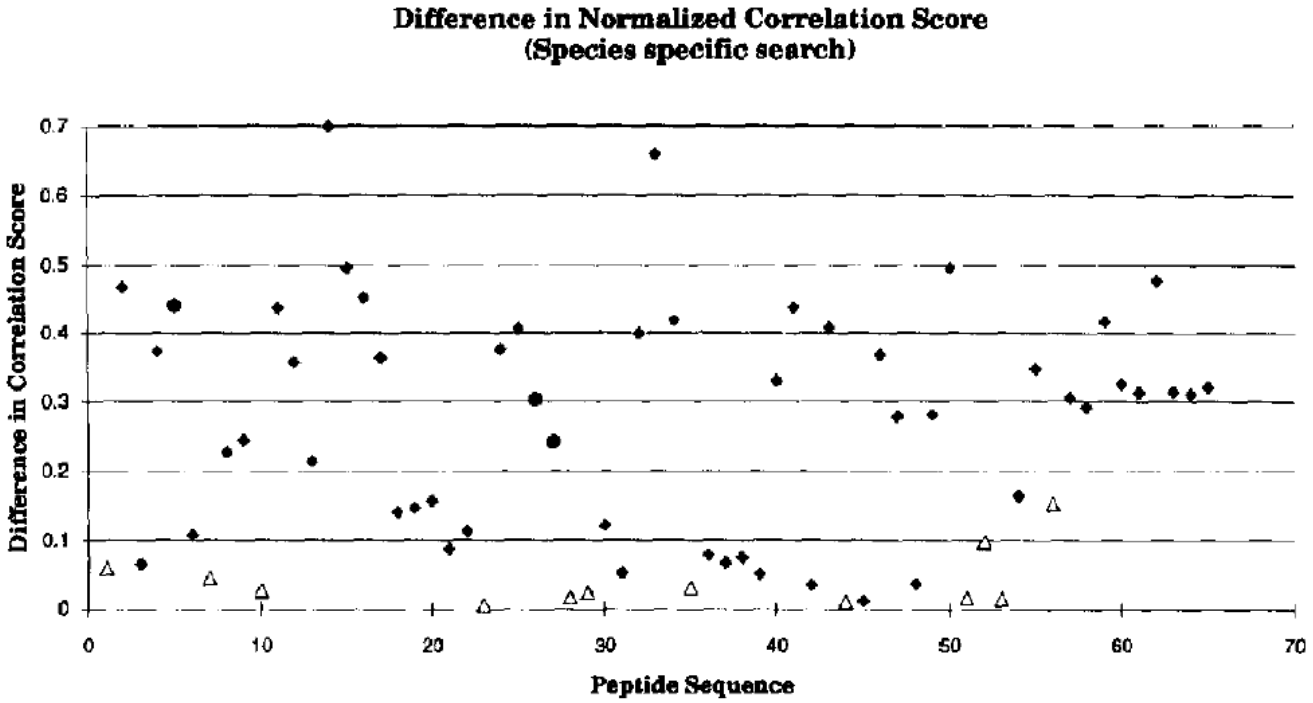
Cross-correlation

4. Select the most promising peptide using the score.

No.	Db	Protein identified	Sequence identified	S ΔC_n	S C_n	G C_n
42	E	trypsin, bovine	SSGTSYPDVLK	0.035	1	1
43	E	trypsin, bovine	TLNNDIMLIK	0.407	1	1
44	E	trypsin, bovine	SIVHPSYNSNTLNNDIMLIK	0.012	4	17
45	E	trypsin, bovine	VASISLPTSCASAGTQCLISGWGNTK	0.012	1	7
46	E	trypsin, bovine	VASISLPTSCASAGTQCLISGWGNTK	0.367	1	1
47	E	tufA gene product, *E. coli*	VGEEVEIVGIK	0.278	1	1
48	E	tufA gene product, *E. coli*	VTLIHPIAMDDGLR	0.037	1	2
49	E	ORF-D, *E. coli*	GGDTVTLNETDLTQIPK	0.281	1	1
50	E	GAD α protein, *E. coli*	GWQVPAFTLGGEATDIVVMR	0.494	1	1
51	H	mRNA for HLA-DR antigens	MATPLLMQALP	0.017	5	—
52	H	mRNA for HLA-DR antigens	MATPLLMQALPM	0.098	2	9
53	H	mRNA for HLA-DR antigens	KPPKPVSKMR*	0.015	7	26
54	H	mRNA for HLA-DR antigens	PKPPKPVSKMR*	0.163	1	2
55	H	lymphocyte antigen	DLRSWTAADTAAQISQ	0.347	1	1
56	H	MHC Class I HLA	DLRSWTAADTAAQITQ	0.152	2	2
57	Y	enolase gene, *Sc*	EEALDLIVDAIK	0.305	1	1
58	Y	enolase gene, *Sc*	NPTVEVELTTEK	0.292	1	1
59	Y	enolase gene, *Sc*	DPFAEDDWEAWSH	0.416	1	1
60	Y	pyruvate kinase, *Sc*	LPGTDVDLPALSEK	0.325	1	1
61	Y	hexokinase PI gene, *Sc*	IEDDPFVFLEDTDDIFQK	0.312	1	1
62	Y	hypusine protein HP2, *Sc*	APEGELGDSLQTAFDEGK	0.476	1	1
63	Y	chromosome III compl. DNA, *Sc*	IPAGWQGLDNGPESR	0.314	1	1
64	Y	chromosome III compl. DNA, *Sc*	TGGGASLELLEGK	0.310	1	1
65	Y	BMH1 gene product, *Sc*	QAFDDAIAELDTLSEESYK	0.321	1	1

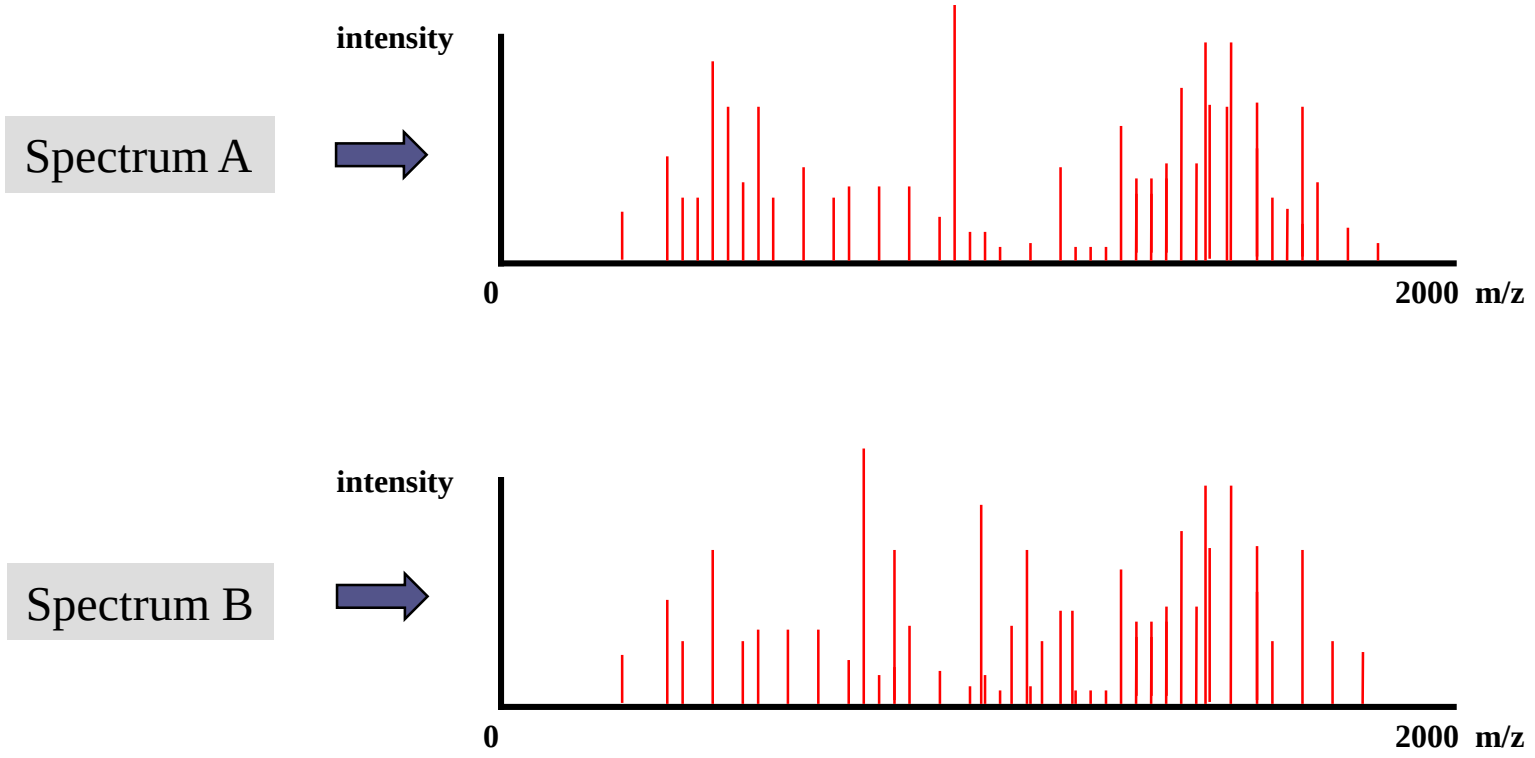
Cross-correlation

4. Select the most promising peptide using the score.



Cross-correlation

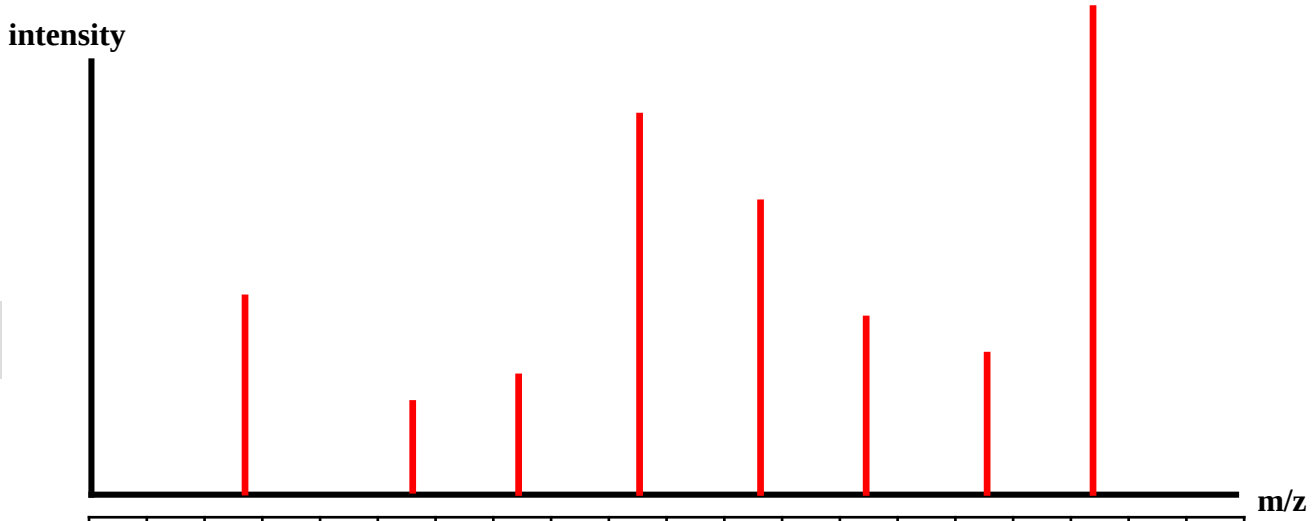
- **Cross-correlation**
  - Measures the similarity of two spectra



Cross-correlation

- **Cross-correlation**

Spectrum *A*

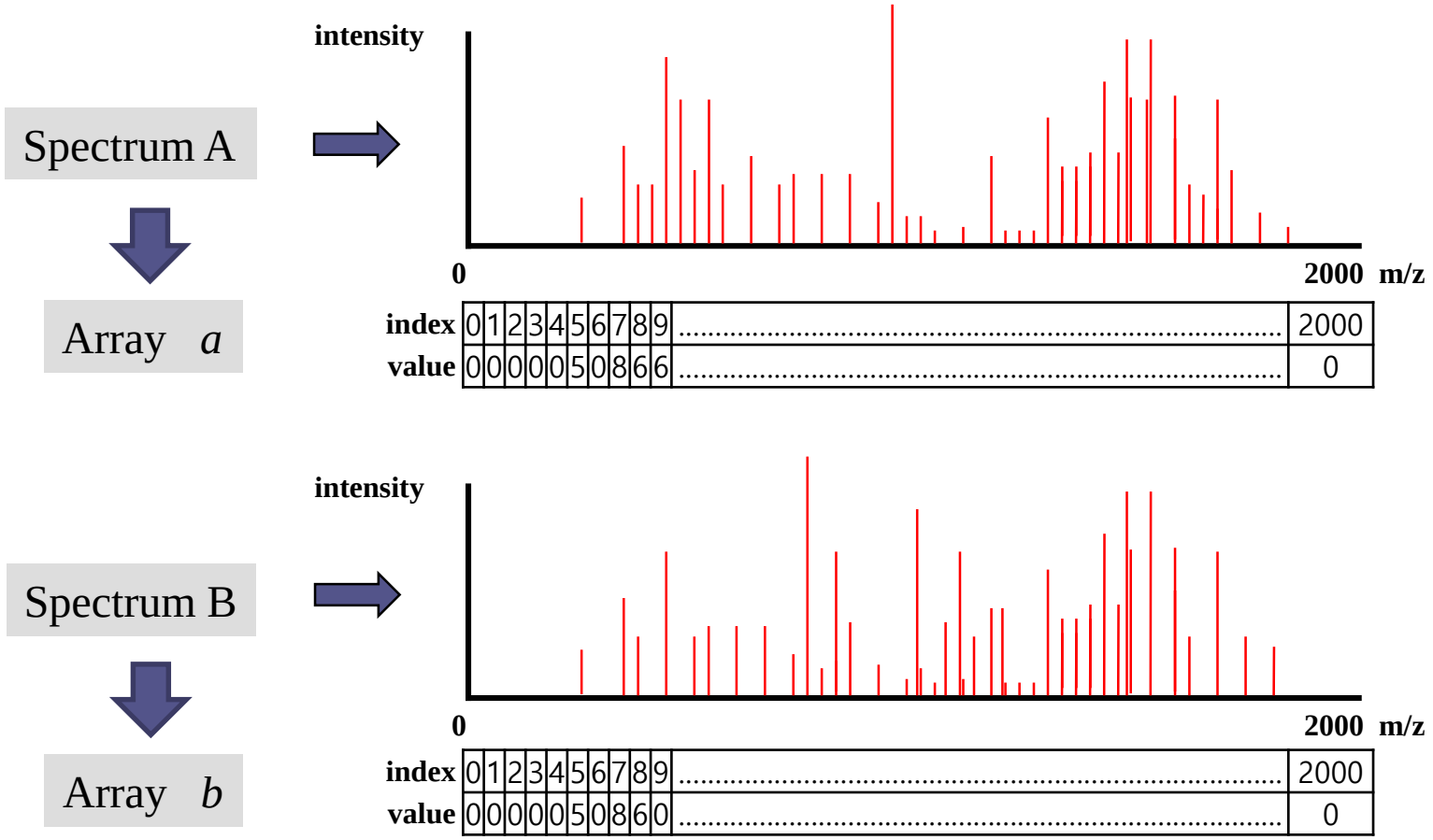


Array *a*

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
value	0	0	5	0	0	3	0	4	0	10	0	8	0	5	0	4	0	12	0	0

Cross-correlation

- **Cross-correlation**

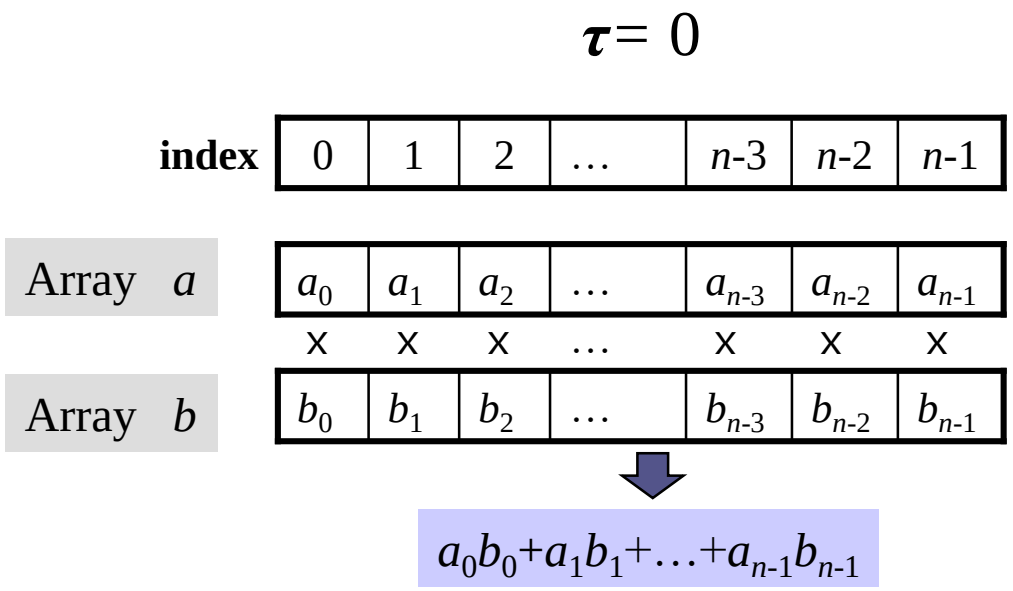


Cross-correlation

- **Cross-correlation for peptide identification**
  - XCorr Score = XCorr value at $\tau = 0$
    - Mean of XCorr values over $-75 < \tau < 75$

Cross-correlation

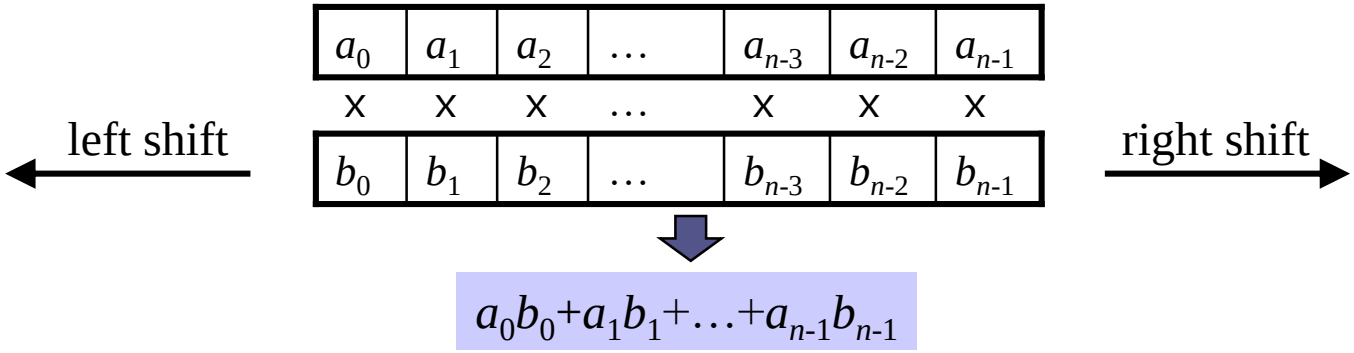
- **Cross-correlation for peptide identification**



Cross-correlation

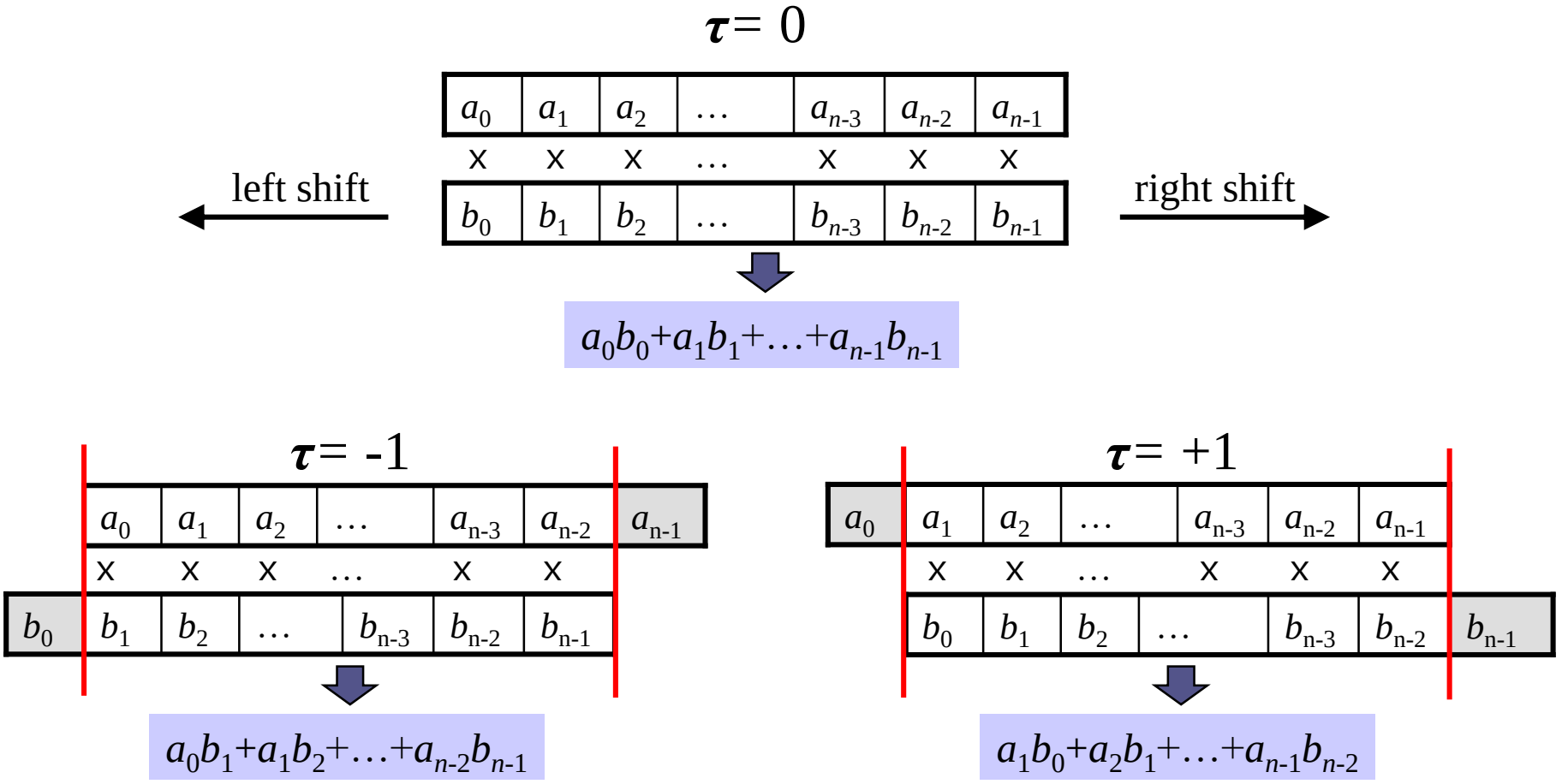
- **Cross-correlation for peptide identification**

$\tau = 0$



Cross-correlation

- **Cross-correlation for peptide identification**

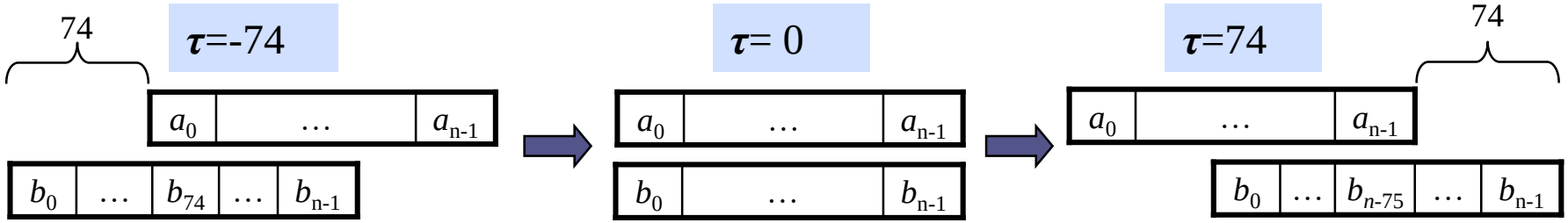


Cross-correlation

- **Cross-correlation for peptide identification**
  - XCorr Score = XCorr value at $\tau = 0$
    - Mean of XCorr values over $-75 < \tau < 75$

Cross-correlation

- **How to compute mean of the XCorr values over $-75 < \tau < 75$**



$$\frac{1}{149}\sum_{\tau=-74}^{74}\sum_{i=0}^{n-1} a[i]b[i+\tau] = \frac{1}{149}(a_0b_{74} + a_1b_{75} + \cdots + a_{n-75}b_{n-1}) \quad \tau = -74$$

$$+ a_0b_{73} + a_1b_{74} + \cdots + a_{n-74}b_{n-1} \quad \tau = -73$$

$$\vdots$$

$$+ a_0b_0 + a_1b_1 + \cdots + a_{n-1}b_{n-1} \quad \tau = 0$$

$$\vdots$$

$$+ a_{73}b_0 + a_{74}b_1 + \cdots + a_{n-1}b_{n-74} \quad \tau = 73$$

$$+ a_{74}b_0 + a_{75}b_1 + \cdots + a_{n-1}b_{n-75}) \quad \tau = 74$$

Linear time algorithm

- **Time complexity is $\Theta(\tau n)$**

$$\sum_{\tau=-74}^{74}\sum_{i=0}^{n-1} a[i]b[i+\tau] = a_0b_{74} + a_1b_{75} + \cdots + a_{n-75}b_{n-1} \quad \tau = -74$$

$$+ a_0b_{73} + a_1b_{74} + \cdots + a_{n-74}b_{n-1} \quad \tau = -73$$

$$\vdots$$

$$+ a_0b_0 + a_1b_1 + \cdots + a_{n-1}b_{n-1} \quad \tau = 0$$

$$\vdots$$

$$+ a_{73}b_0 + a_{74}b_1 + \cdots + a_{n-1}b_{n-74} \quad \tau = 73$$

$$+ a_{74}b_0 + a_{75}b_1 + \cdots + a_{n-1}b_{n-75} \quad \tau = 74$$

Linear time algorithm

- **Time complexity is $\Theta(\tau + n)$**

$$
\begin{aligned}
& a_0\ (b_0 + \quad \cdots \quad + b_{74}) \\
+\ & a_1\ (b_0 + \quad \cdots \quad + b_{74} + b_{75}) \\
& \qquad \vdots \\
+\ & a_{74}\ (b_0 + \qquad \cdots \qquad b_{147} + b_{148}) \\
\hline
+\ & a_{75}\ (\quad b_1 + \qquad \cdots \qquad + b_{148} + b_{149}) \\
& \qquad \vdots \\
+\ & a_{n-75}(\quad b_{n-149} + \qquad \cdots \qquad + b_{n-2} + b_{n-1}) \\
\hline
+\ & a_{n-74}(b_{n-148} + b_{n-147} + \quad \cdots \quad + b_{n-1}) \\
& \qquad \vdots \\
+\ & a_{n-2}\,(b_{n-76} + b_{n-75} + \cdots + b_{n-1}) \\
+\ & a_{n-1}\,(b_{n-75} + \cdots + b_{n-1})
\end{aligned}
$$

Linear time algorithm

- **Time complexity is $\Theta(\tau + n)$**

$$
\begin{aligned}
& a_0\ (b_0 + \cdots + b_{74}) \\
+\ & a_1\ (b_0 + \cdots + b_{74} + b_{75}) \\
& \vdots \\
+\ & a_{74}\ (b_0 + \cdots + b_{147} + b_{148}) \\
\hline
+\ & a_{75}\ (b_1 + \cdots + b_{148} + b_{149}) \\
& \vdots \\
+\ & a_{n-75}\ (b_{n-149} + \cdots + b_{n-2} + b_{n-1}) \\
\hline
+\ & a_{n-74}\ (b_{n-148} + b_{n-147} + \cdots + b_{n-1}) \\
& \vdots \\
+\ & a_{n-2}\ (b_{n-76} + b_{n-75} + \cdots + b_{n-1}) \\
+\ & a_{n-1}\ (b_{n-75} + \cdots + b_{n-1})
\end{aligned}
$$

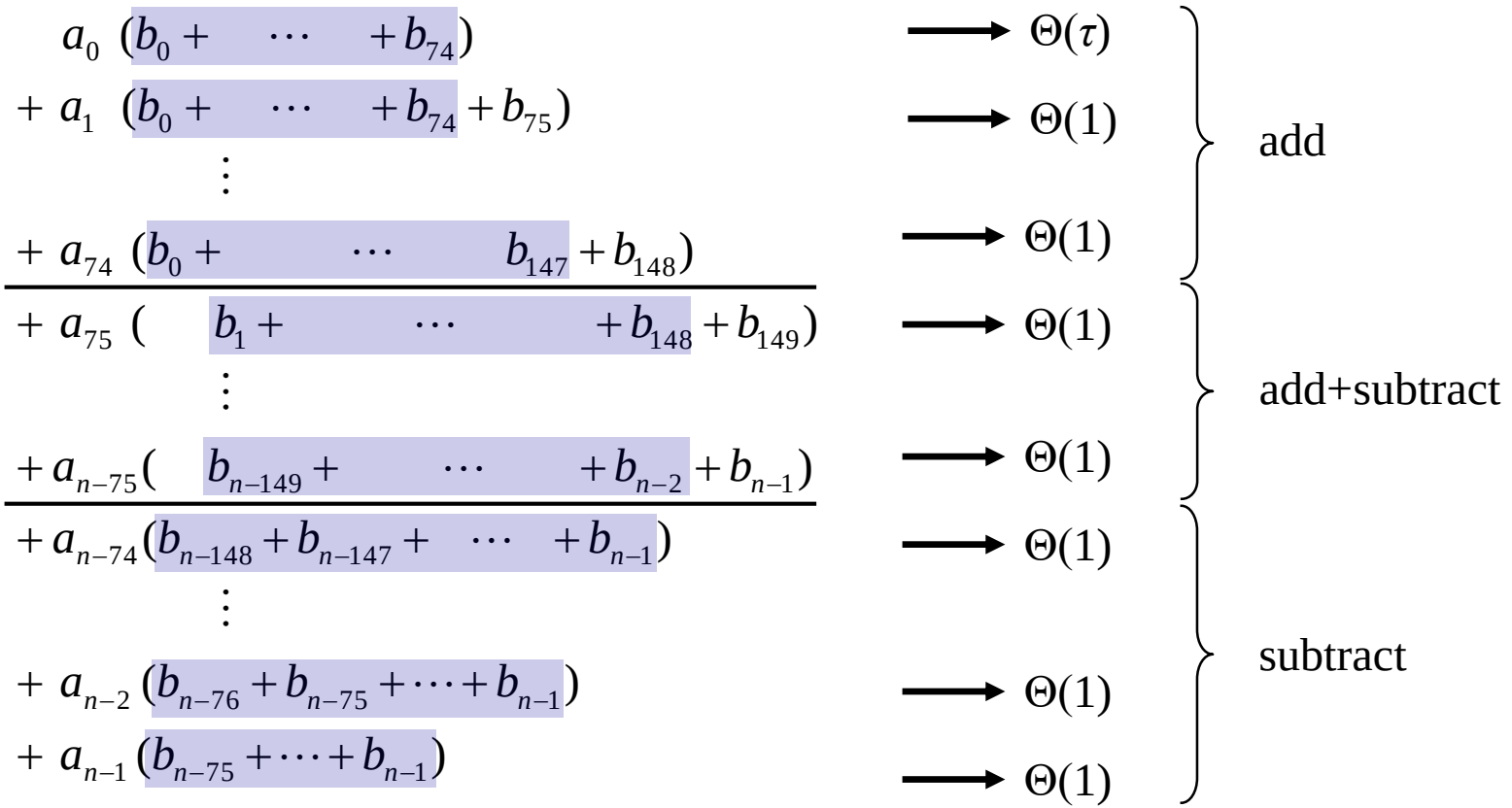
add

add+subtract

subtract

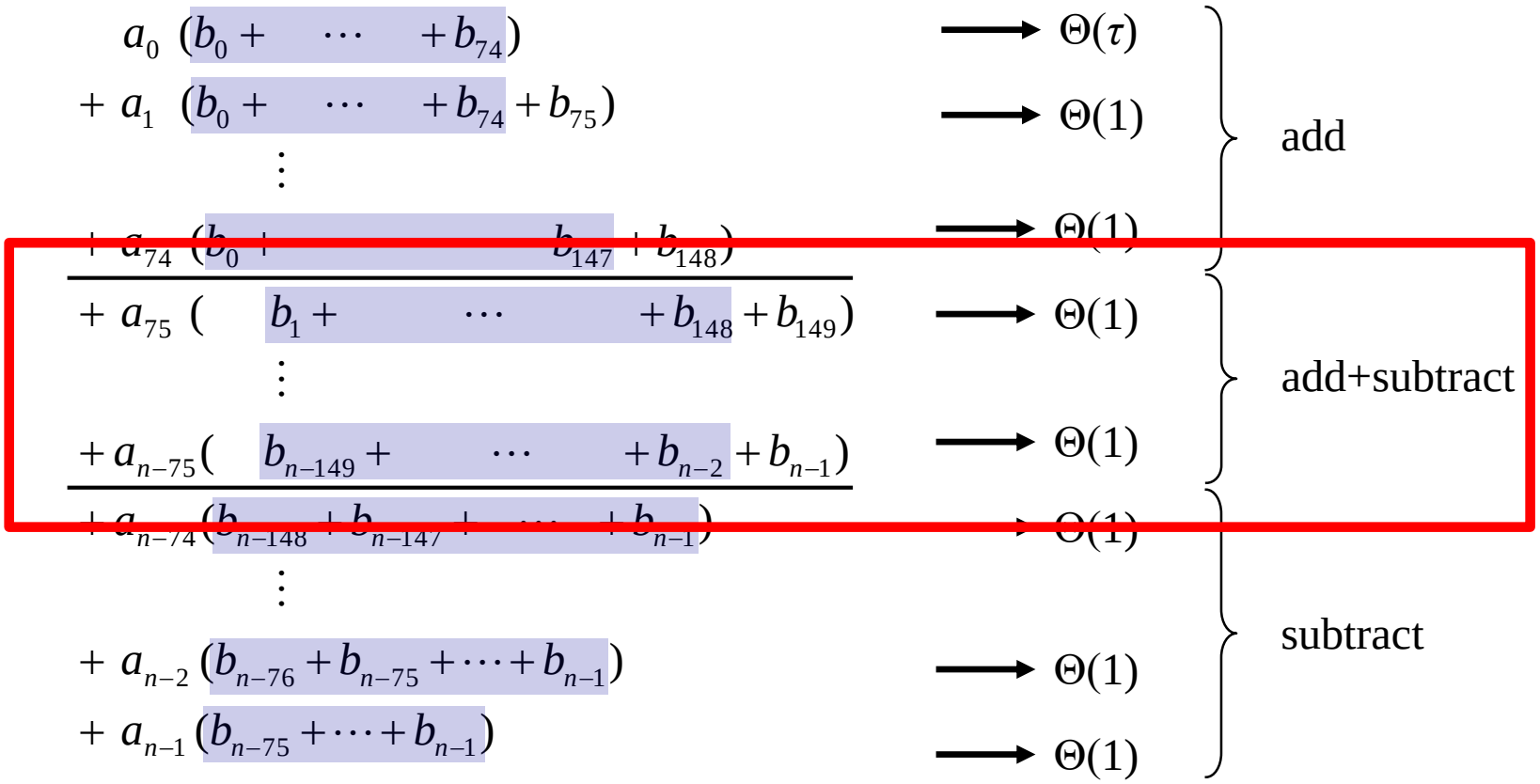
Linear time algorithm

- **Time complexity is $\Theta(\tau + n)$**



Linear time algorithm

- **Time complexity is $\Theta(\tau + n)$**



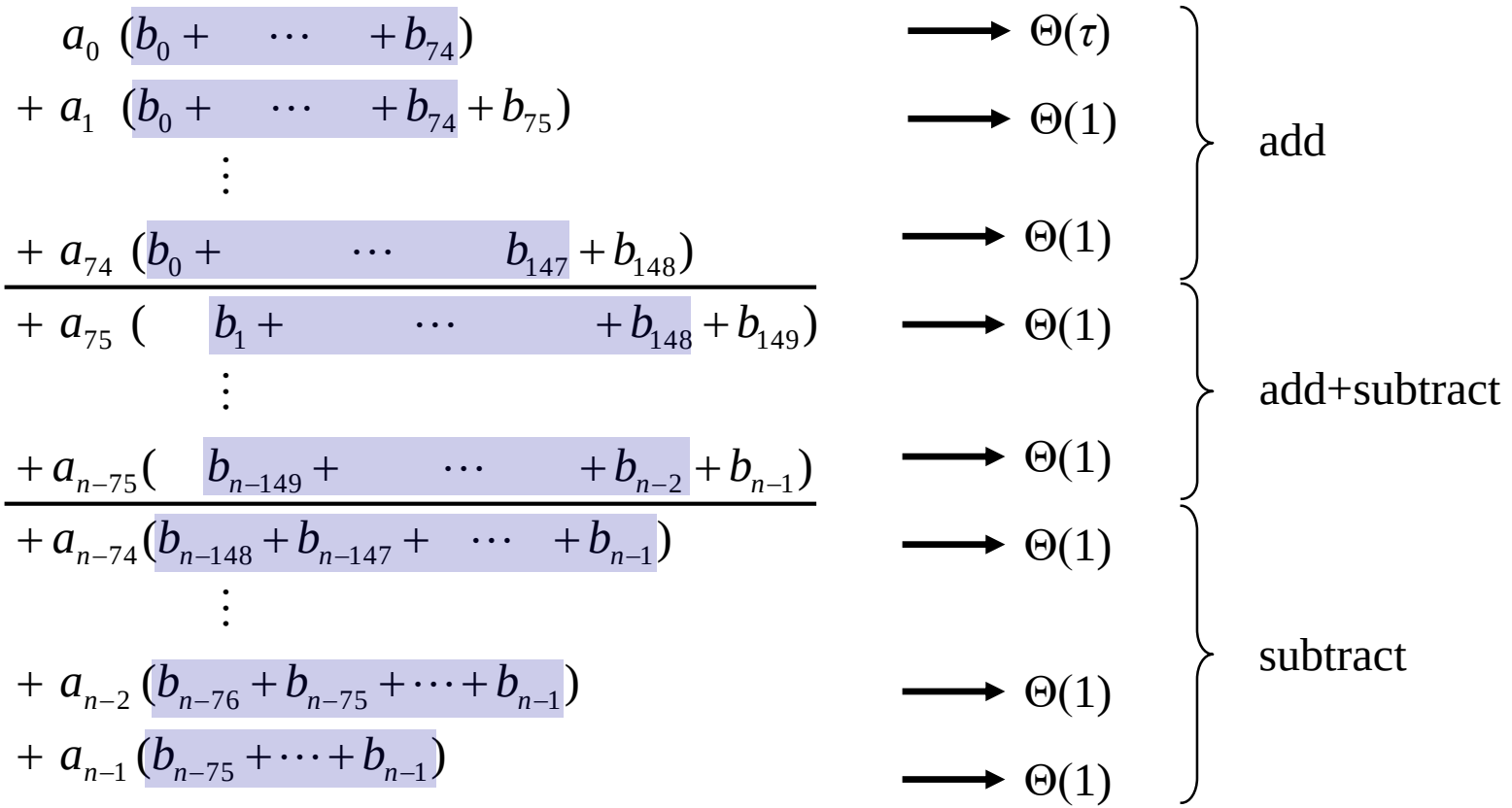
Linear time algorithm

- **Time complexity is $\Theta(\tau + n)$**

$$
\begin{array}{lllll}
 & \vdots & & & \\
+ a_{74} & (b_0 + b_1 + b_2 + b_3 + & \cdots & + b_{148}) & \\
+ a_{75} & (\quad b_1 + b_2 + b_3 + & \cdots & + b_{148} + b_{149}) & \\
+ a_{76} & (\quad \quad b_2 + b_3 + & \cdots & + b_{148} + b_{149} + b_{150}) & \\
+ a_{77} & (\quad \quad \quad b_3 + & \cdots & + b_{148} + b_{149} + b_{150} + b_{151}) & \\
 & \vdots & & &
\end{array}
$$

Linear time algorithm

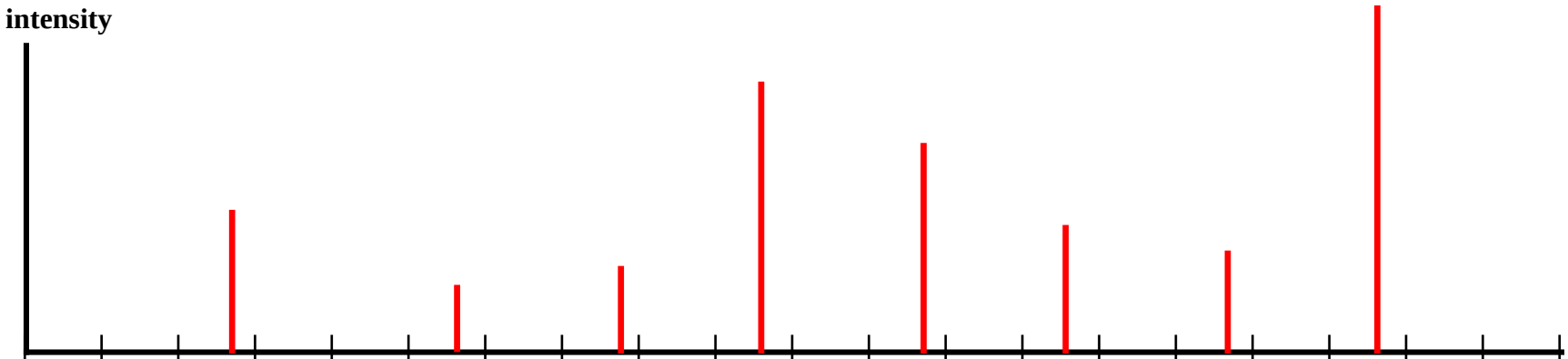
- **Time complexity is $\Theta(\tau + n)$**



Previous Research

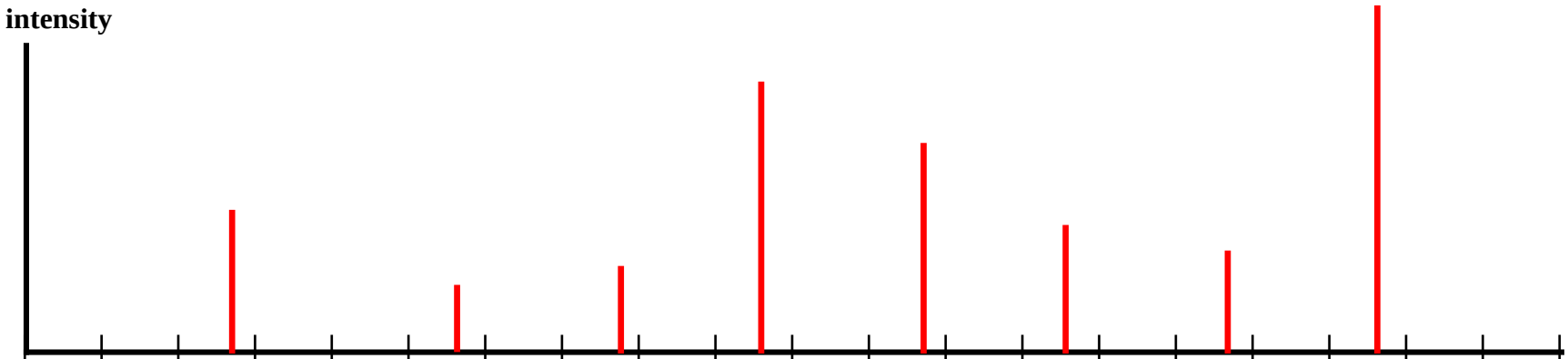
- **Previous algorithm**
  - Time complexity is $\Theta(\tau + n)$
    - n is the number of bins in a spectrum
  - 최근 장비가 발달함에 따라서 고해상도 MS/MS가 생산됨
    - 예전 장비는 저해상도 MS/MS를 생산

Previous Research

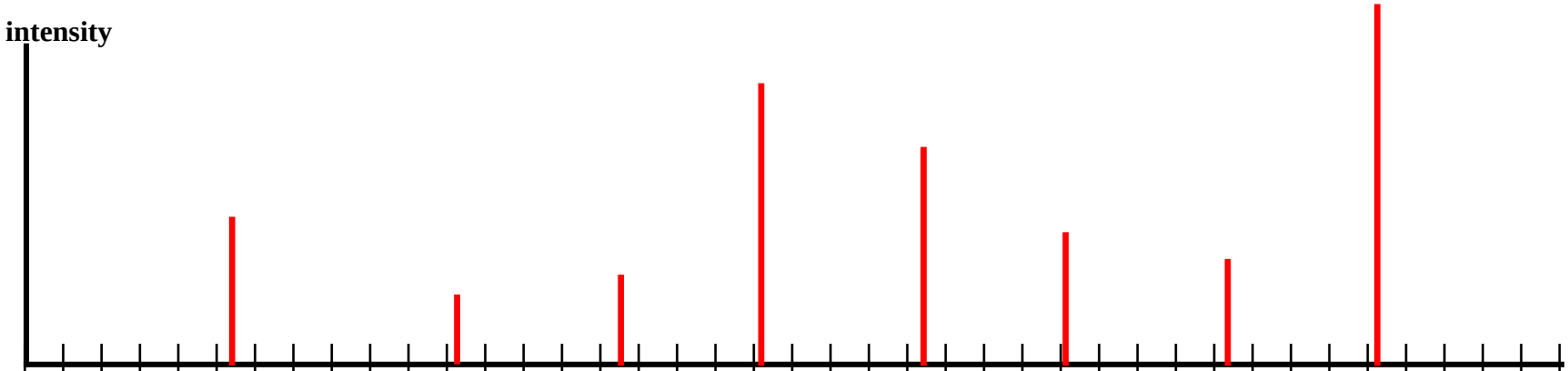


index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
value	0	0	5	0	0	3	0	4	0	10	0	8	0	5	0	4	0	12	0	0

Previous Research



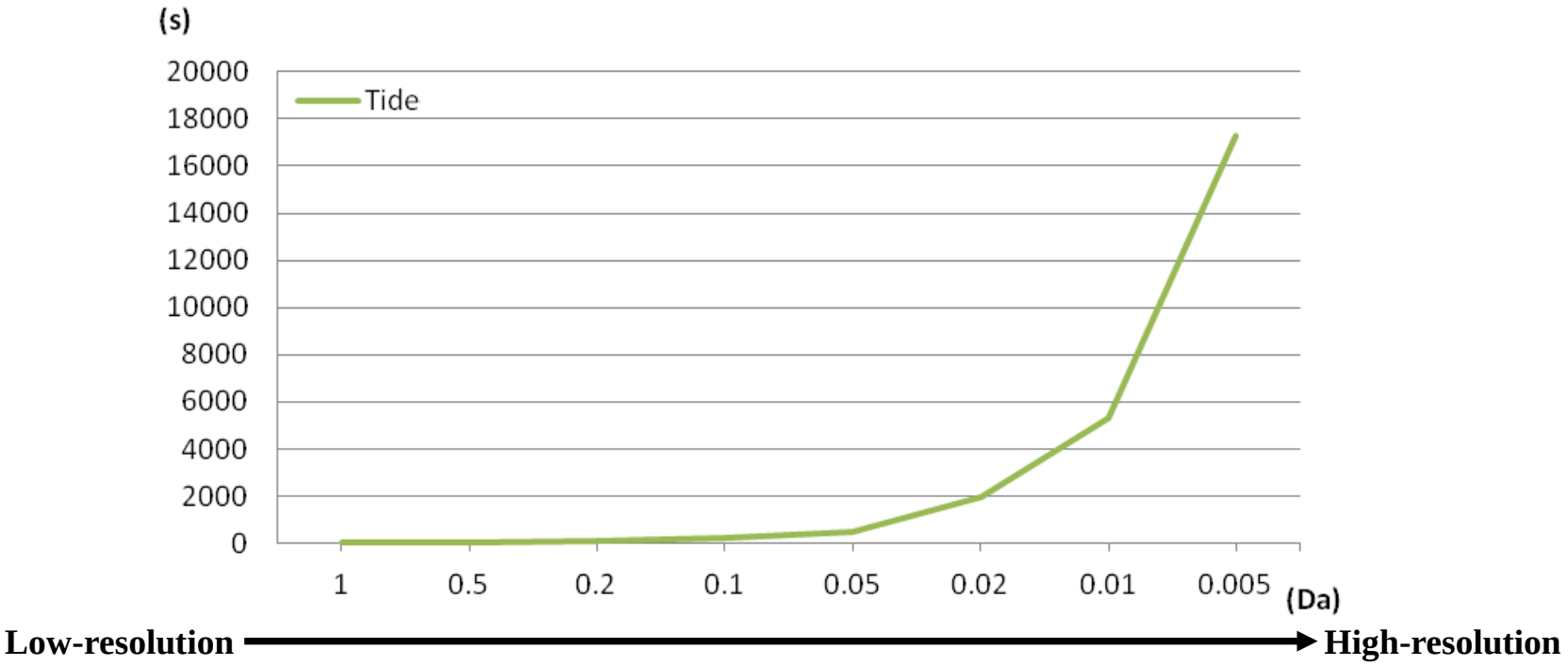
index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
value	0	0	5	0	0	3	0	4	0	10	0	8	0	5	0	4	0	12	0	0



index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
value	0	0	0	0	0	5	0	0	0	0	0	3	0	0	0	4	0	0	0	10	0	0	0	8	0	0	0	5	0	0	0	4	0	0	0	12	0	0	0	0

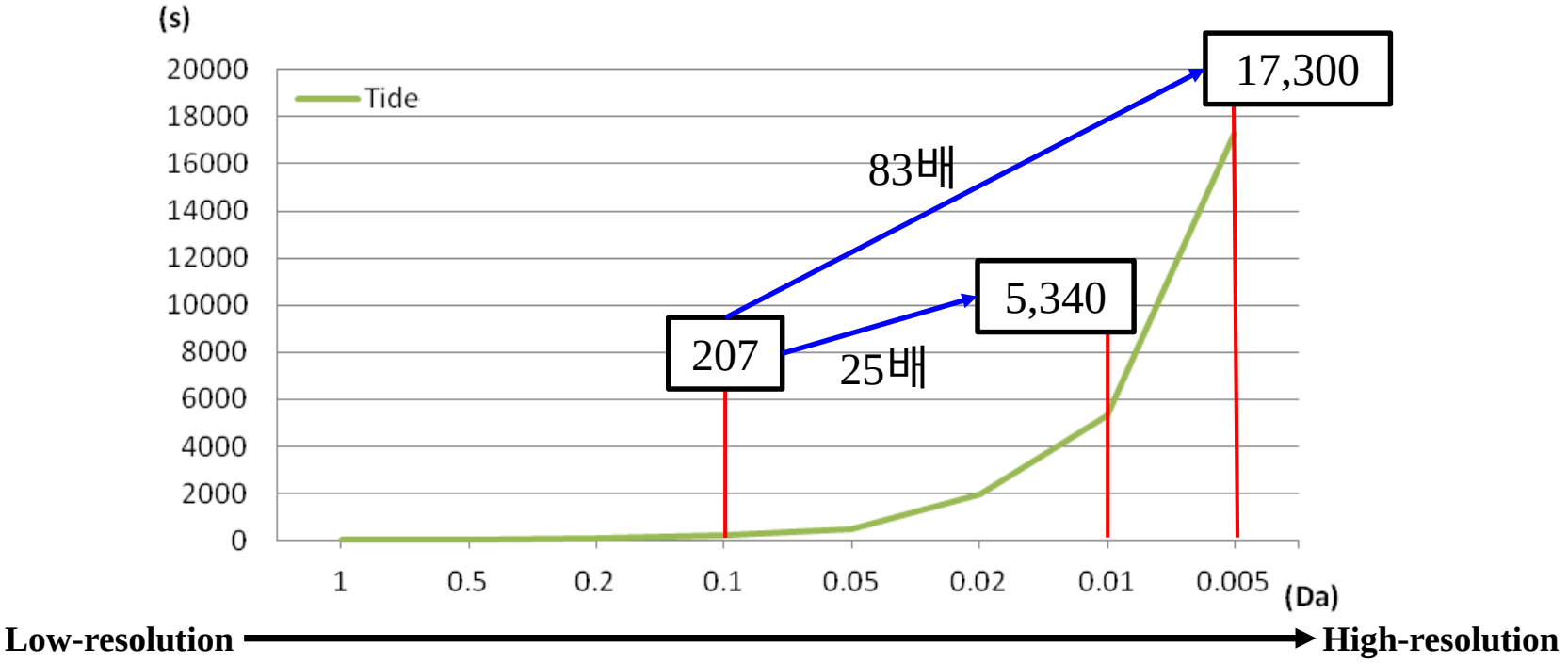
Previous Research

- **Measure the running time of Tide**



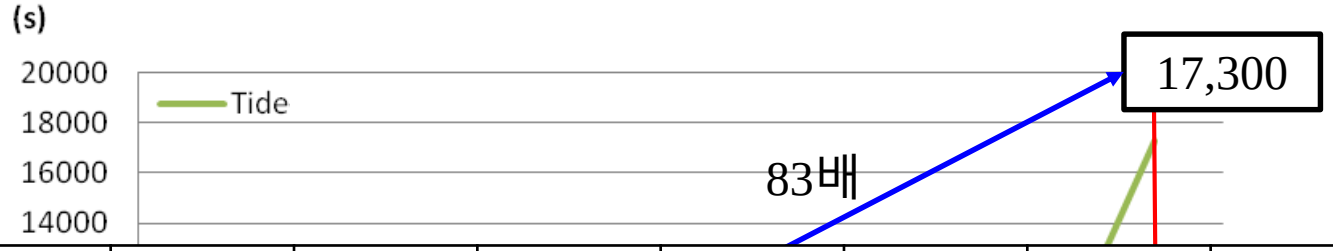
Previous Research

- **Measure the running time of Tide**



Previous Research

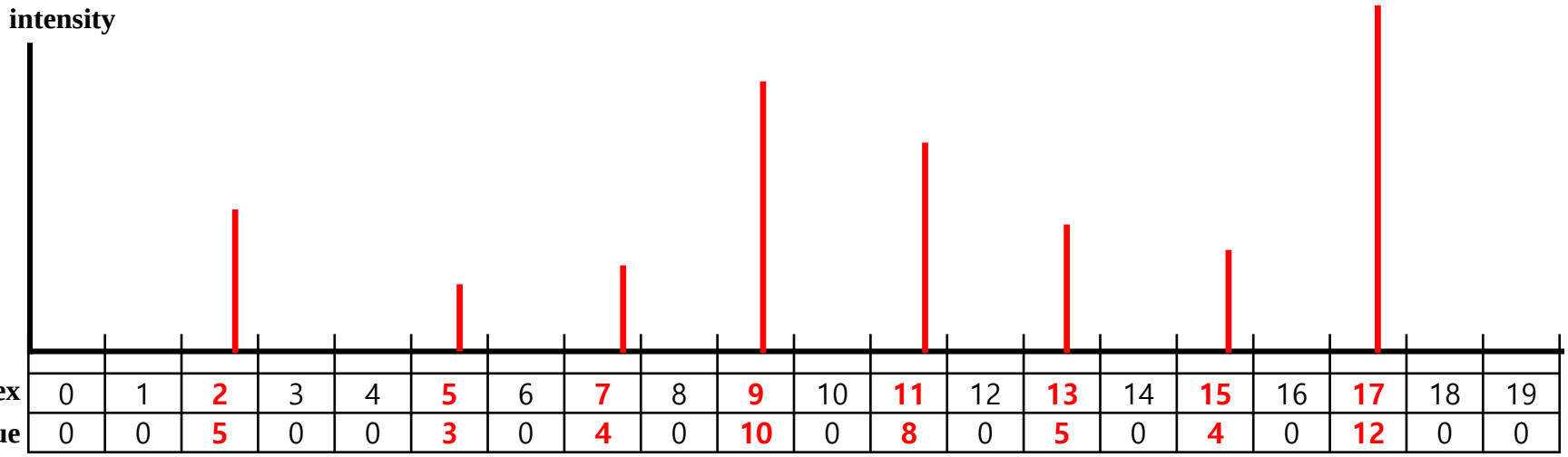
- **Measure the running time of Tide**



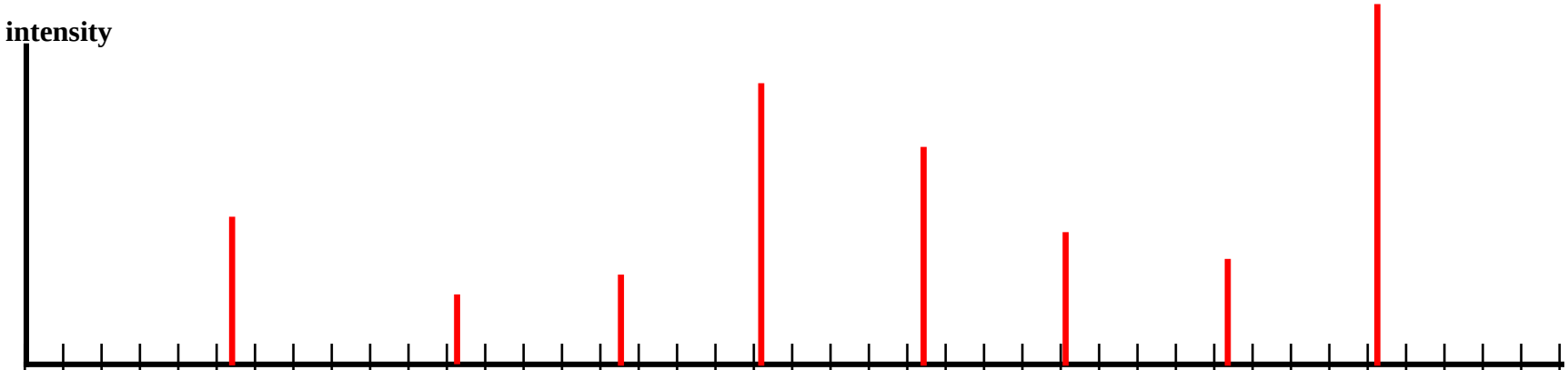
Fragment Tolerance		1	0.5	0.2	0.1	0.05	0.02	0.01	0.005
Tide	Xcorr time	13.51 (50%)	24.43 (58%)	68.77 (69%)	159.1 (76%)	396.8 (82%)	1,727 (87%)	5,313 (99%)	17,208 (99%)
	Total time	26.5	41.9	98.7	207	483	1,970	5,340	17,300

Low-resolution → **High-resolution**

Previous Research

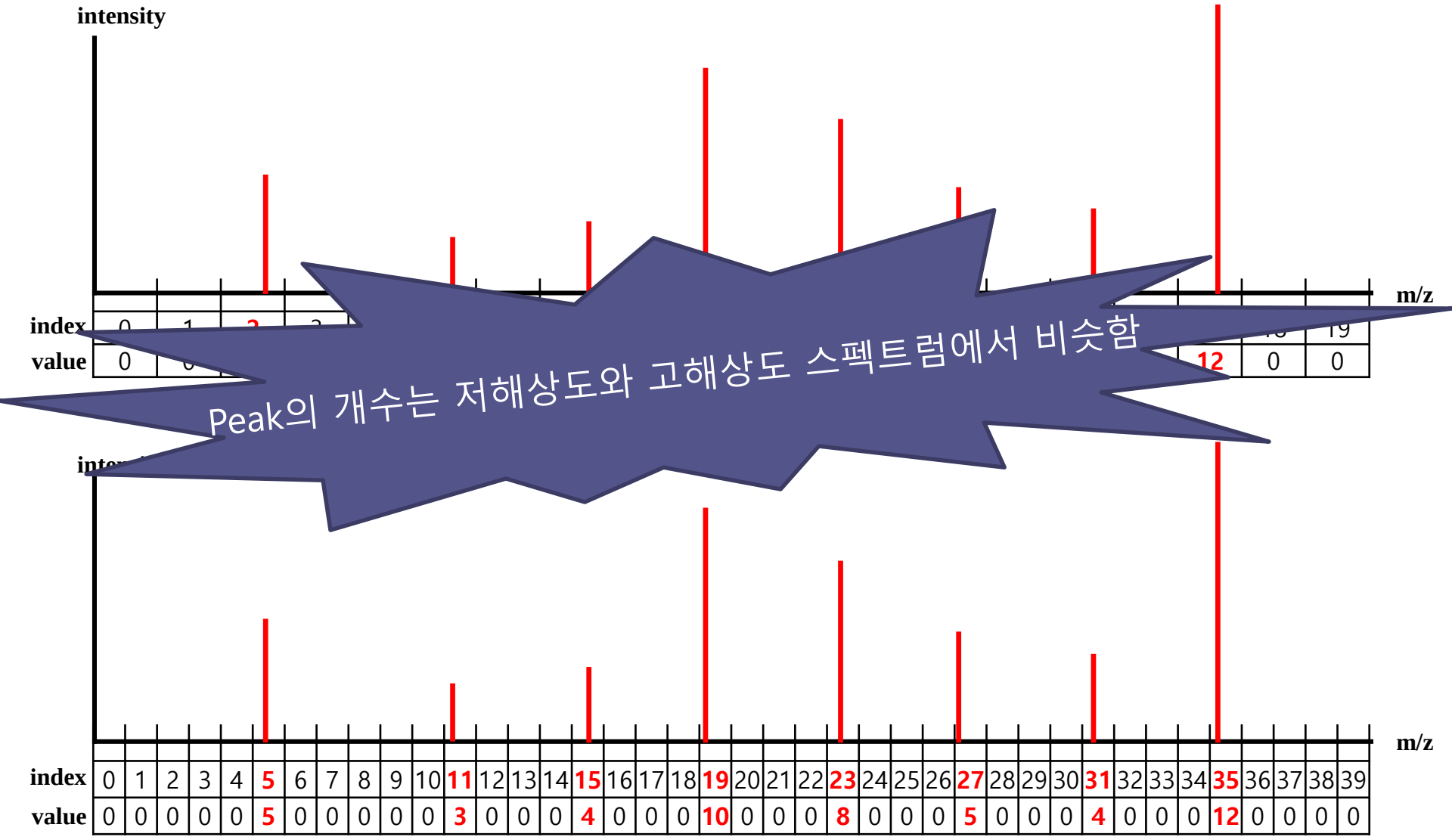


index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
value	0	0	5	0	0	3	0	4	0	10	0	8	0	5	0	4	0	12	0	0



index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
value	0	0	0	0	0	5	0	0	0	0	0	3	0	0	0	4	0	0	0	10	0	0	0	8	0	0	0	5	0	0	0	4	0	0	0	12	0	0	0	0

Previous Research



Our Algorithm

- **Our Algorithm (HiXCorr)**
  - Time complexity is $\Theta(p)$
    - p is the number of peaks in a spectrum

Linear time algorithm

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
value	0	0	5	0	0	3	0	4	0	10	0	8	0	5	0	4	0	12	0	0

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
value	0	0	0	0	0	5	0	0	0	0	0	3	0	0	0	4	0	0	0	10	0	0	0	8	0	0	0	5	0	0	0	4	0	0	0	12	0	0	0	0

$$
\begin{aligned}
&0\ (0+0+5)\\
&+\ 0\ \ (0+0+5+0)\\
&+\ 5\ \ (0+0+5+0+0)\\
&+\ 0\ \ (0+5+0+0+3)\\
&\vdots\\
&+\ 0\ \ (0+4+0+12+0)\\
&+\ 12\ \ (4+0+12+0+0)\\
&+\ 0\ \ (0+12+0+0)\\
&+\ 0\ \ (12+0+0)
\end{aligned}
$$

$$
\begin{aligned}
&0\ (0+0+0)\\
&+\ 0\ \ (0+0+0+0)\\
&+\ 0\ \ (0+0+0+0+0)\\
&+\ 0\ \ (0+0+0+0+5)\\
&+\ 0\ \ (0+0+0+5+0)\\
&+\ 5\ \ (0+0+5+0+0)\\
&\vdots\\
&+\ 0\ \ (0+0+0+12+0)\\
&+\ 12\ \ (0+0+12+0+0)\\
&+\ 0\ \ (0+12+0+0+0)\\
&+\ 0\ \ (12+0+0+0+0)\\
&+\ 0\ \ (0+0+0+0)\\
&+\ 0\ \ (0+0+0)
\end{aligned}
$$

Linear time algorithm

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
value	0	0	5	0	0	3	0	4	0	10	0	8	0	5	0	4	0	12	0	0

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
value	0	0	0	0	0	5	0	0	0	0	0	3	0	0	0	4	0	0	0	10	0	0	0	8	0	0	0	5	0	0	0	4	0	0	0	12	0	0	0	0

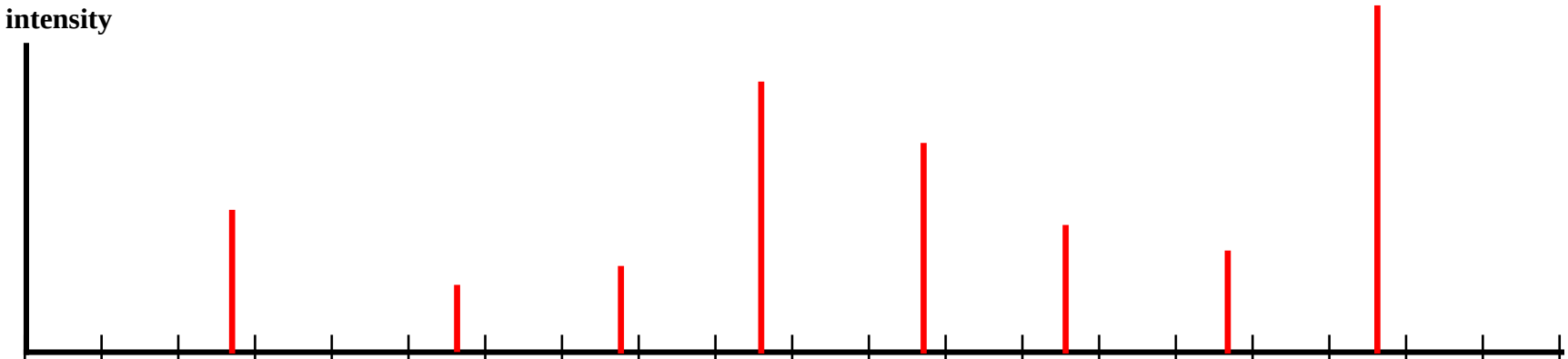
$$
\begin{aligned}
&0\ (0+0+5)\\
+\ &0\ \ (0+0+5+0)\\
+\ &\boxed{5}\ (0+0+5+0+0)\\
+\ &0\ \ (0+5+0+0+3)\\
&\vdots\\
+\ &0\ \ (0+4+0+12+0)\\
+\ &\boxed{12}\ (4+0+12+0+0)\\
+\ &0\ \ (0+12+0+0)\\
+\ &0\ \ (12+0+0)
\end{aligned}
$$

$$
\begin{aligned}
&0\ (0+0+0)\\
+\ &0\ \ (0+0+0+0)\\
+\ &0\ \ (0+0+0+0+0)\\
+\ &0\ \ (0+0+0+0+5)\\
+\ &0\ \ (0+0+0+5+0)\\
+\ &\boxed{5}\ (0+0+5+0+0)\\
&\vdots\\
+\ &0\ \ (0+0+0+12+0)\\
+\ &\boxed{12}\ (0+0+12+0+0)\\
+\ &0\ \ (0+12+0+0+0)\\
+\ &0\ \ (12+0+0+0+0)\\
+\ &0\ \ (0+0+0+0)\\
+\ &0\ \ (0+0+0)
\end{aligned}
$$

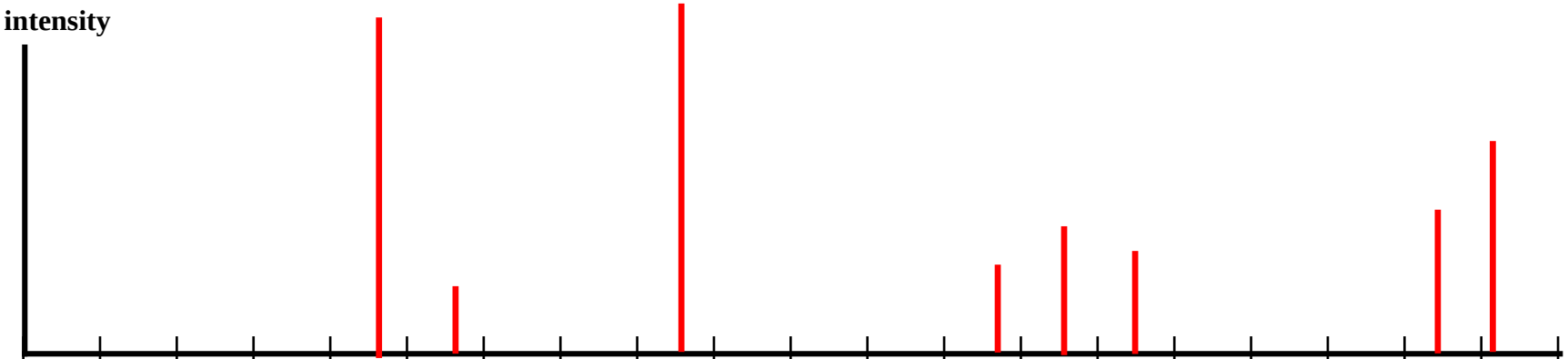
Our Algorithm

- **Our Algorithm (HiXCorr)**
  - Time complexity is $\Theta(p)$
    - p is the number of peaks in a spectrum

Our Algorithm (Example)



index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
value	0	0	5	0	0	3	0	4	0	10	0	8	0	5	0	4	0	12	0	0



index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
value	0	0	0	0	12	3	0	0	12	0	0	0	4	5	4	0	0	0	8	10

Our Algorithm (Example, τ=3)

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	0	0	5	0	0	3	0	4	0	10	0	8	0	5	0	4	0	12	0	0
B	0	0	0	0	12	3	0	0	12	0	0	0	4	5	4	0	0	0	8	10

$$
\begin{aligned}
&0\ (0+0+0)\\
+&0\ (0+0+0+0)\\
+&5\ (0+0+0+0+12)\\
+&0\ (\quad 0+0+0+12+3)\\
+&0\ (\quad\ \ 0+0+12+3+0)\\
+&3\ (\quad\ \ \ 0+12+3+0+0)\\
\hline
+&12\ (\quad\ \ \ \ 0+0+0+8+10)\\
+&0\ \ (\quad\ \ \ \ \ 0+0+8+10)\\
+&0\ \ (\quad\ \ \ \ \ \ 0+8+10)
\end{aligned}
$$

Our Algorithm (Example, τ=3)

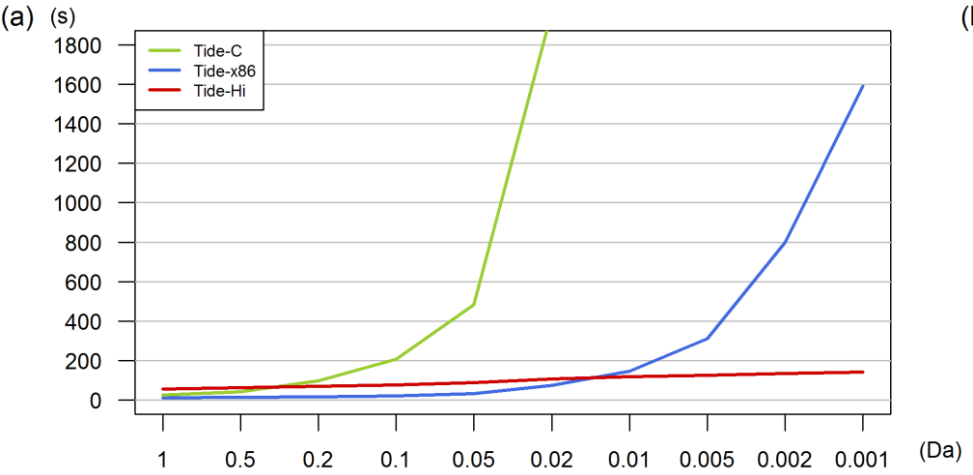
index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	0	0	5	0	0	3	0	4	0	10	0	8	0	5	0	4	0	12	0	0
B	0	0	0	0	12	3	0	0	12	0	0	0	4	5	4	0	0	0	8	10

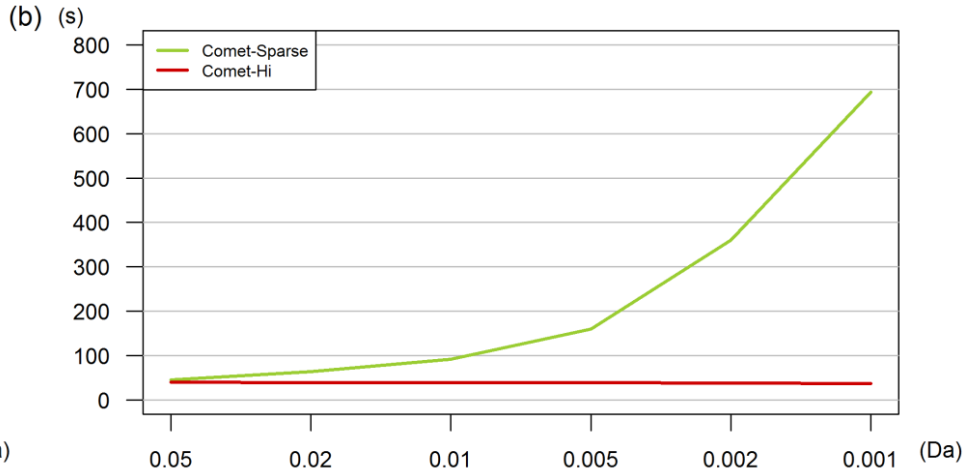
$$
\begin{aligned}
&0\ (0+0+0)\\
&+0\ (0+0+0+0)\\
&+5\ (0+0+0+0+12) && +5\ (\qquad 12)\\
&+0\ (\quad 0+0+0+12+3)\\
&+0\ (\qquad 0+0+12+3+0)\\
&+3\ (\qquad\quad 0+12+3+0+0) && +3\ (\qquad\quad 12+3\qquad)\\
&\hline\\
&+12\ (\qquad\quad 0+0+0+8+10) && +12\ (\qquad\qquad\quad 8+10)\\
&+0\quad (\qquad\quad 0+0+8+10)\\
&+0\quad (\qquad\quad\quad 0+8+10)
\end{aligned}
$$

Our Algorithm (Example, τ=3)

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	0	0	5	0	0	3	0	4	0	10	0	8	0	5	0	4	0	12	0	0
B	0	0	0	0	12	3	0	0	12	0	0	0	4	5	4	0	0	0	8	10

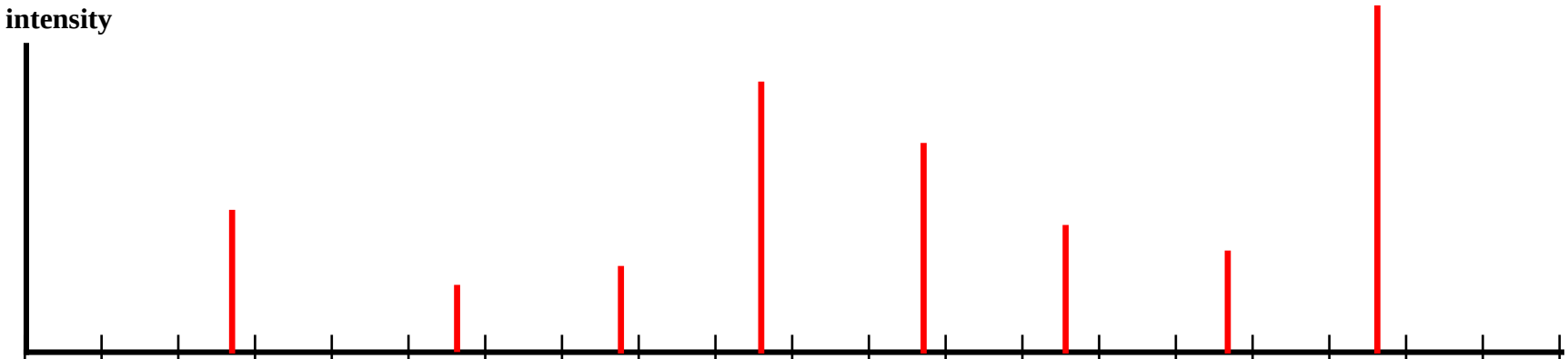
$$0\ (0+0+0)$$
$$+0\ (0+0+0+0)$$



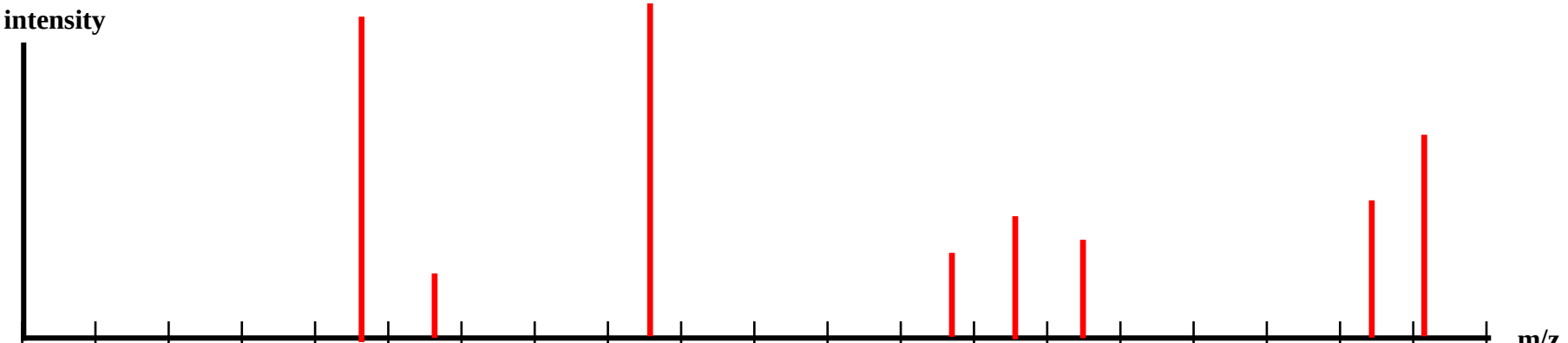


$$+0\ (\qquad 0+0+8+10)$$
$$+0\ (\qquad 0+8+10)$$

실습 구현



index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
value	0	0	5	0	0	3	0	4	0	10	0	8	0	5	0	4	0	12	0	0



index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
value	0	0	0	0	12	3	0	0	12	0	0	0	4	5	4	0	0	0	8	10