

XCorr 구현 실습

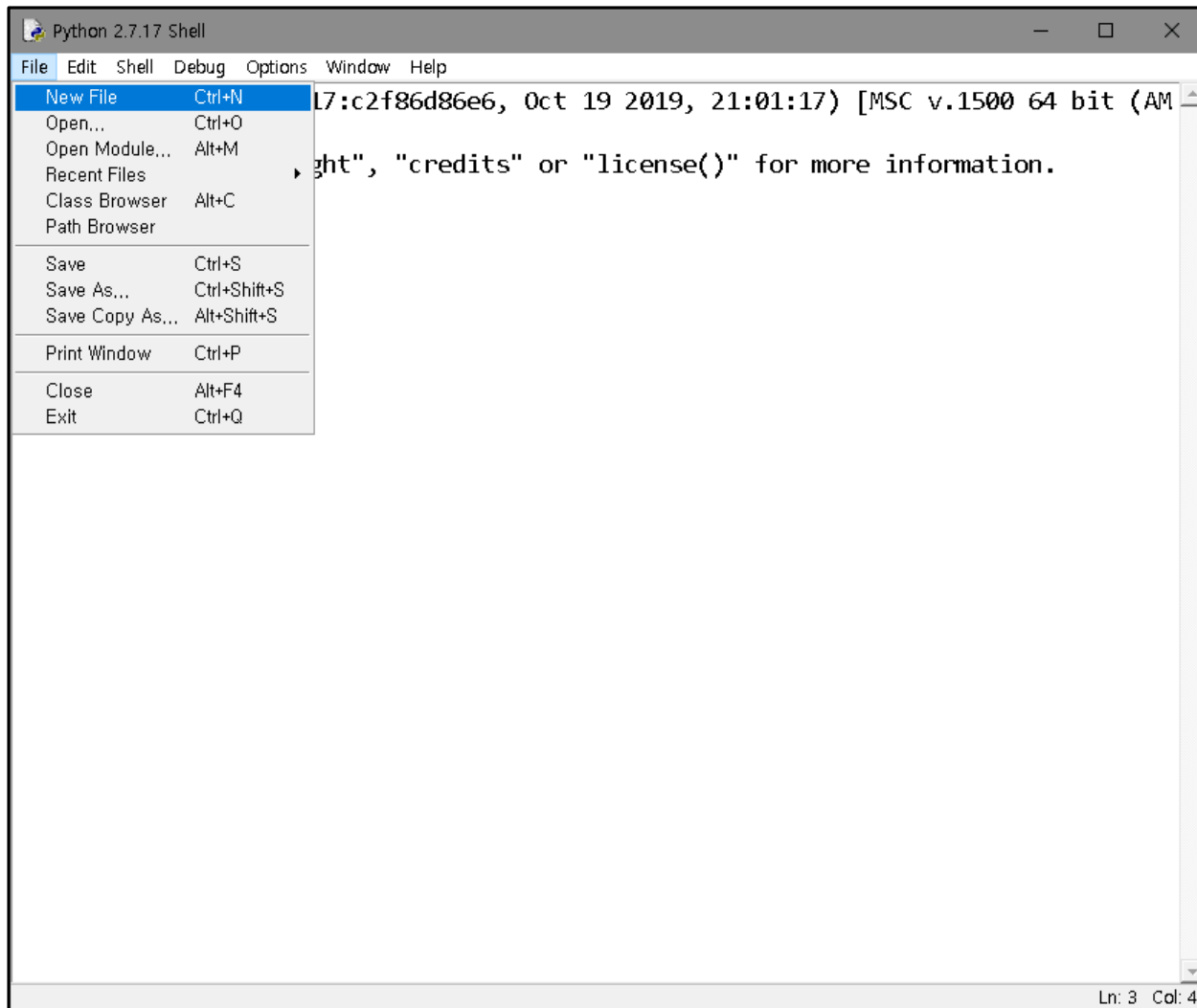
2020.06

김현우

XCorr 구현 실습

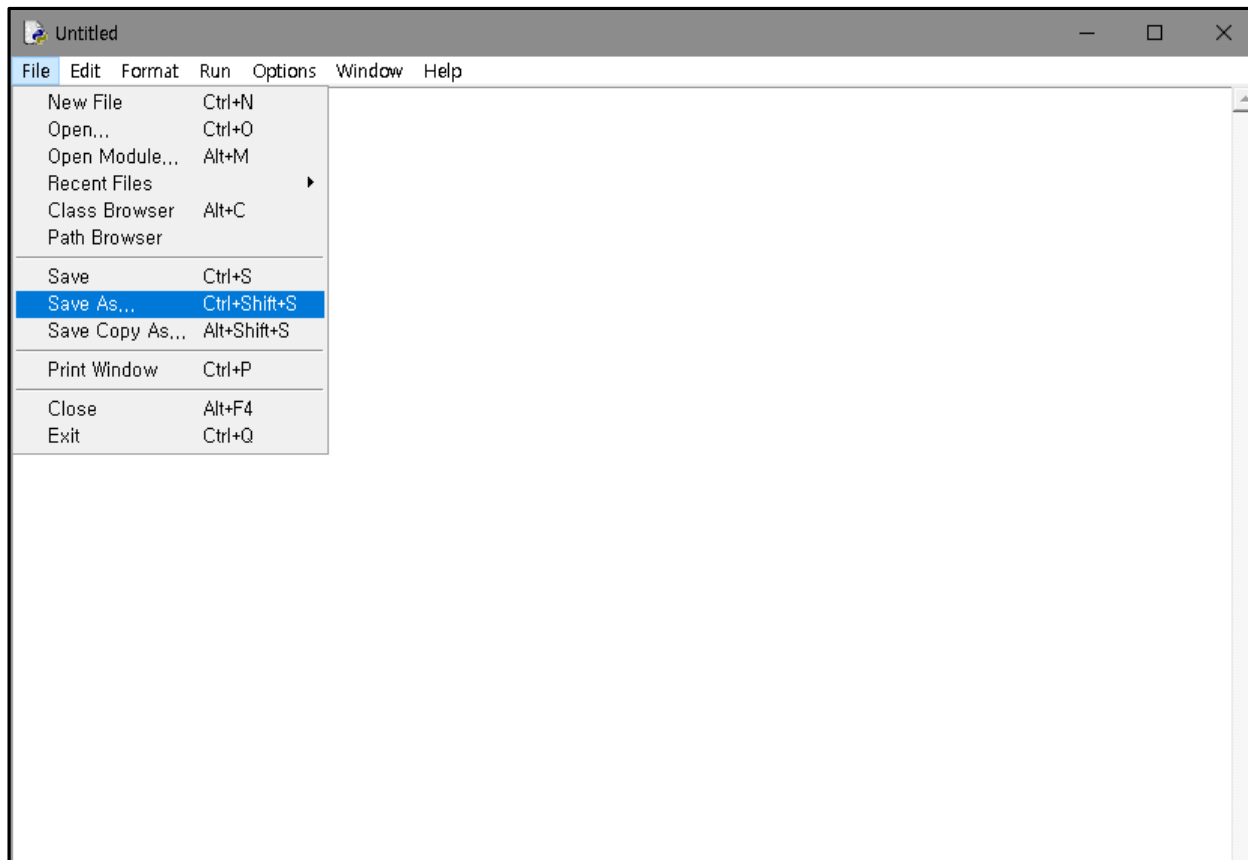
- **XCorr 구현 실습 (Python 2.7)**
 - 데이터 입력, 데이터베이스 입력
 - 임의의 가상 데이터: mgf 파일 & fasta 파일
- XCorr 구현
 - XCorr Crux(fast SEQUEST) version
 - $\Theta(\tau n)$
 - XCorr Comet version
 - $\Theta(\tau + n)$
 - XCorr HiXCorr version
 - $\Theta(p)$
- 시간 비교

XCorr 구현 실습



XCorr 구현 실습

- 실습 폴더에 저장



XCorr 구현 실습



XCorr 구현 실습

아미노산 정보 및 해상도 입력



실험 스펙트럼 파일 입력



데이터베이스 파일 입력



이론 스펙트럼 생성



XCorr 구현 !

XCorr 구현 실습

- 각 아미노산 질량 정보를 가지게 하는 코드 작성

Name	Symbol	Monoisotopic Mass
Leucine	L	45
Serine	S	35
Lysine	K	55
Glutamic Acid	E	50
Glycine	G	25
Alanine	A	30
Valine	V	40

XCorr 구현 실습

- 1의 해상도를 갖도록 하는 코드 작성

XCorr 구현 실습

아미노산 정보 및 해상도 입력



실험 스펙트럼 파일 입력



데이터베이스 파일 입력



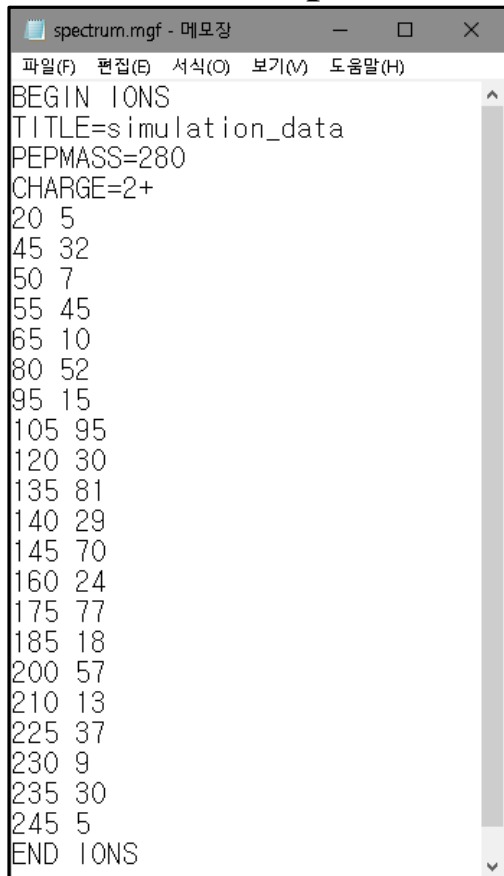
이론 스펙트럼 생성



XCorr 구현 !

XCorr 구현 실습

- 실험 스펙트럼 파일 입력
 - File name: "spectrum.mgf"

A screenshot of a text editor window titled "spectrum.mgf - 메모장". The window contains the following text:

```
BEGIN IONS
TITLE=simulation_data
PEPMASS=280
CHARGE=2+
20 5
45 32
50 7
55 45
65 10
80 52
95 15
105 95
120 30
135 81
140 29
145 70
160 24
175 77
185 18
200 57
210 13
225 37
230 9
235 30
245 5
END IONS
```

XCorr 구현 실습

- 실험 스펙트럼 파일 입력

- File name: "spectrum.mgf"

```

BEGIN IONS
TITLE=simulation_data
PEPMASS=280
CHARGE=2+
20 5
45 32
50 7
55 45
65 10
80 52
95 15
105 95
120 30
135 81
140 29
145 70
160 24
175 77
185 18
200 57
210 13
225 37
230 9
235 30
245 5
END IONS

```

스펙트럼의 총 크기를 나타냄

스펙트럼의 m/z 리스트

스펙트럼의 intensity 리스트

XCorr 구현 실습

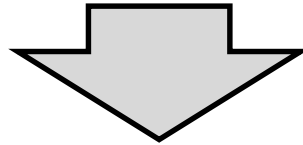
- 실험 스펙트럼 파일 입력
 - 스펙트럼 파일 read
- 실험 스펙트럼 배열 생성하는 코드를 작성
 - 배열 크기 = 스펙트럼의 총 크기 ÷ 해상도
 - 배열의 index: m/z
 - 배열의 value: intensity
- m/z 리스트와 intensity 리스트를 읽으며, 해당하는 index에 intensity 값 저장

XCorr 구현 실습

- 실험 스펙트럼 파일 입력

- 해상도가 1일 때, 실험 스펙트럼 배열 출력 결과는 다음과 같음

```
print experimental
```



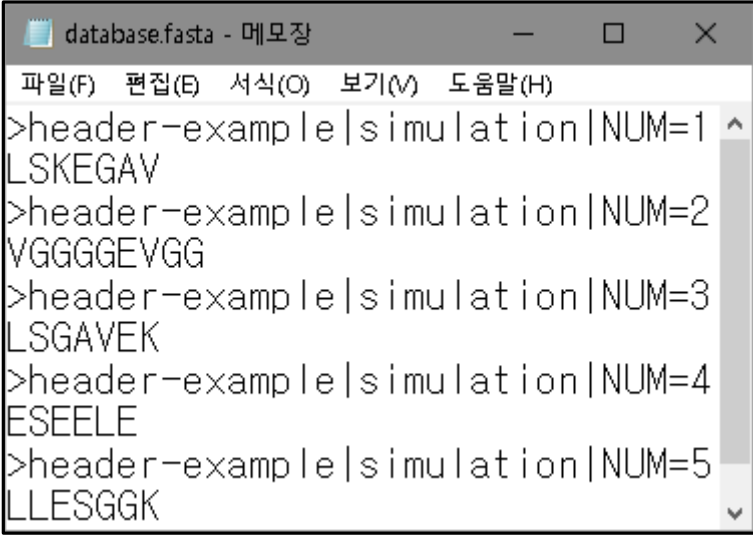
```
[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5.0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 32.0, 0, 0, 0, 0, 0, 7.0,
 0, 0, 0, 0, 45.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 10.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 , 0, 0, 0, 0, 52.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 15.0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 95.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 30.0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 81.0, 0, 0, 0, 0, 29.0, 0, 0, 0, 0, 70.0, 0, 0, 0,
 , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 24.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 77.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 18.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 57.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 13.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 , 0, 37.0, 0, 0, 0, 0, 9.0, 0, 0, 0, 30.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 5.0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 , 0, 0, 0, 0, 0, 0, 0]
```

XCorr 구현 실습



XCorr 구현 실습

- 데이터베이스 파일 입력
 - File name: "database.fasta"



```
database.fasta - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
>header-example|simulation|NUM=1
LSKEGAV
>header-example|simulation|NUM=2
VGGGGEVGG
>header-example|simulation|NUM=3
LSGAVEK
>header-example|simulation|NUM=4
ESEELE
>header-example|simulation|NUM=5
LLESGGK
```

XCorr 구현 실습

- 데이터베이스 파일 입력
 - 변수 생성하는 코드를 작성:
 - peptides 배열
 - 1등 peptide의 sequence
 - 1등 peptide의 점수

XCorr 구현 실습

- 데이터베이스 파일 입력

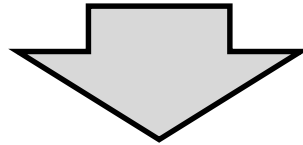
- database파일 내 5개의 peptides개를 peptides 배열에 저장하는 코드 작성

XCorr 구현 실습

- 데이터베이스 파일 입력

- peptides 배열 출력 결과는 다음과 같음

```
print peptides
```



```
['LSKEGAV', 'VGGGGEVGG', 'LSGAVEK', 'ESEELE', 'LLESGGK']
```

XCorr 구현 실습



XCorr 구현 실습

- 이론 스펙트럼

- 각 peptide로 부터 이론 스펙트럼 배열 생성하는 코드를 작성
 - 이론 스펙트럼 배열의 크기 = 실험 스펙트럼 배열의 크기
 - 이론 스펙트럼 배열에 b-ion 저장
 - 이론 스펙트럼 배열에 y-ion 저장
 - 해상도를 고려할 것

XCorr 구현 실습

- 이론 스펙트럼
 - 해상도가 1일 때, 이론 스펙트럼 배열 출력 결과는 다음과 같음

```
for p in range(len(peptides)):

    #make theoretical spectrum

    print theoretical
    print ""

    #XCorr
```

[illegible]

XCorr 구현 실습



XCorr 구현 실습

아미노산 정보 및 해상도 입력



실험 스펙트럼 파일 입력



데이터베이스 파일 입력



이론 스펙트럼 생성



Crux(fast SEQUEST) version

Comet version

HiXCorr version

XCorr 구현 실습

- **XCorr**
 - Crux(fast SEQUEST) version
 - experimental 배열과 theoretical 배열을 사용
- XCorr 결과 중 가장 높은 점수 및 peptide sequence 저장

XCorr 구현 실습

- **XCorr**
 - XCorr 점수 출력 결과는 다음과 같음

```
for p in range(len(peptides)):
    #make theoretical spectrum
    #XCorr
    if xcorr > bestXCorr:
        bestPeptide = peptides[p]
        bestXCorr = xcorr
    print xcorr
    print
print bestPeptide
print bestXCorr
```



```
338.094594595
22.7972972973
551.175675676
177.222972973
425.533783784
LSGAVEK
551.175675676
```

XCorr 구현 실습

- 시간 측정

```
import time
startTime = time.time()

#####XCorr#####

print "running time : ", time.time() - startTime
```

XCorr 구현 실습

- 해상도 변경
 - 코드 상단 부분의 해상도 값 변경하여 실행 (F5)
 - 1
 - 0.1
 - 0.01
 - 0.001

XCorr 구현 실습

아미노산 정보 및 해상도 입력



실험 스펙트럼 파일 입력



데이터베이스 파일 입력



이론 스펙트럼 생성



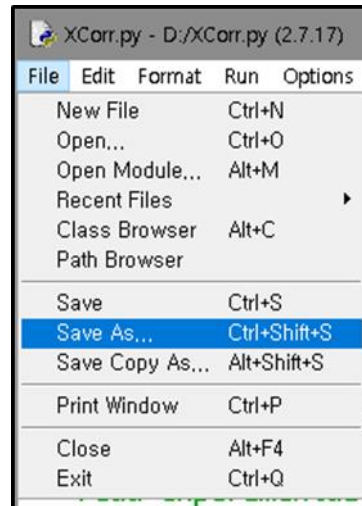
Crux(fast SEQUEST) version

Comet version

HiXCorr version

XCorr 구현 실습

- 복사
 - Comet version을 위해 파일 복사



XCorr 구현 실습

- **XCorr**
 - Comet version으로 코드 수정하기
 - experimental 배열과 theoretical 배열을 사용

XCorr 구현 실습

- 해상도 변경
 - 코드 상단 부분의 해상도 값 변경하여 실행 (F5)
 - 1
 - 0.1
 - 0.01
 - 0.001

XCorr 구현 실습

아미노산 정보 및 해상도 입력



실험 스펙트럼 파일 입력



데이터베이스 파일 입력



이론 스펙트럼 생성



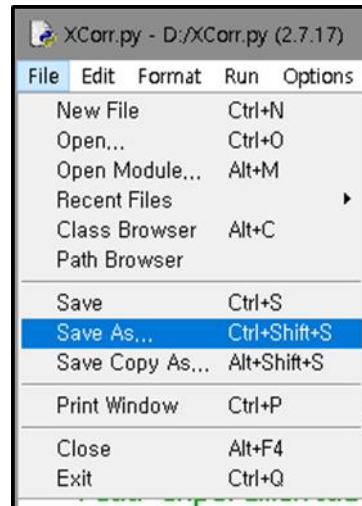
Crux(fast SEQUEST) version

Comet version

HiXCorr version

XCorr 구현 실습

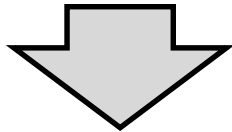
- 복사
 - HiXCorr version을 위해 파일 복사



XCorr 구현 실습

- 스펙트럼의 배열 사이즈 및 형태가 다름

배열 크기 = 스펙트럼의 크기 ÷ 해상도



배열 크기 = peaks의 개수

XCorr 구현 실습

아미노산 정보 및 해상도 입력



실험 스펙트럼 파일 입력



데이터베이스 파일 입력



이론 스펙트럼 생성



Crux(fast SEQUEST) version

Comet version

HiXCorr version

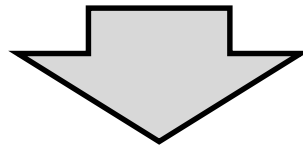
XCorr 구현 실습

- HiXCorr version을 위한 수정
 - 실험 스펙트럼 파일 입력
 - 스펙트럼 파일 read
 - 실험 스펙트럼 peaks 리스트 배열 생성하는 코드를 작성
 - 배열 크기 = 스펙트럼내 peaks의 개수
 - 리스트의 0번 index: m/z
 - 리스트의 1번 index: intensity
 - m/z 와 intensity를 읽으며, 해당하는 index에 값 저장

XCorr 구현 실습

- 실험 스펙트럼 파일 입력 (HiXCorr version)
 - 실험 스펙트럼 peaks 리스트 배열 출력 결과는 다음과 같음

```
print "number of peaks : ", e_peaks_num  
print  
print e_peaks
```



```
number of peaks : 21  
[[20, 5.0], [45, 32.0], [50, 7.0], [55, 45.0], [65, 10.0], [80, 52.0], [95, 15.0], [105, 95.0], [120, 30.0], [135, 81.0], [140, 29.0],  
[145, 70.0], [160, 24.0], [175, 77.0], [185, 18.0], [200, 57.0], [210, 13.0], [225, 37.0], [230, 9.0], [235, 30.0], [245, 5.0]]
```

XCorr 구현 실습

아미노산 정보 및 해상도 입력



실험 스펙트럼 파일 입력



데이터베이스 파일 입력



이론 스펙트럼 생성



Crux(fast SEQUEST) version

Comet version

HiXCorr version

XCorr 구현 실습

- **HiXCorr version을 위한 수정**
 - 각 peptide로 부터 **이론 스펙트럼 peaks 리스트 배열** 생성하는 코드를 작성
 - 이론 스펙트럼 배열의 크기 = 실험 스펙트럼 배열의 크기
 - 이론 스펙트럼 peaks 리스트 배열에 b-ion 저장
 - 이론 스펙트럼 peaks 리스트 배열에 y-ion 저장
 - 중복되는 peaks 리스트 제거
 - 각 peak의 0번째 $\text{index}(m/z)$ 를 기준으로 오름차순 정렬

XCorr 구현 실습

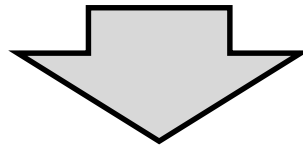
- 이론 스펙트럼 (HiXCorr version)
 - 이론 스펙트럼 peaks 리스트 배열 출력 결과는 다음과 같음

```
for p in range(len(peptides)):

    #make theoretical spectrum'''

    print "number of peaks : ", t_peaks_num
    print t_peaks
    print

    #XCorr
```



```
number of peaks : 12
[(40, 1), (45, 1), (70, 1), (80, 1), (95, 1), (135, 1), (145, 1), (185, 1), (200, 1), (210, 1), (235, 1), (240, 1)]

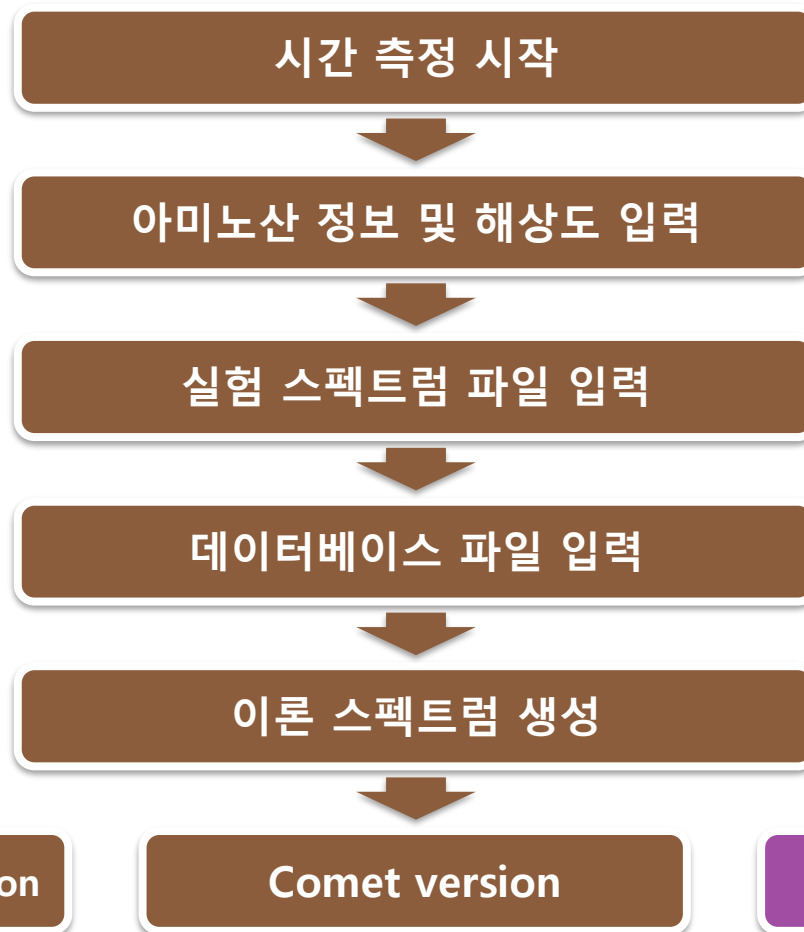
number of peaks : 13
[(25, 1), (40, 1), (50, 1), (65, 1), (90, 1), (115, 1), (140, 1), (165, 1), (190, 1), (215, 1), (230, 1), (240, 1), (255, 1)]

number of peaks : 10
[(45, 1), (55, 1), (80, 1), (105, 1), (135, 1), (145, 1), (175, 1), (200, 1), (225, 1), (235, 1)]

number of peaks : 8
[(50, 1), (85, 1), (95, 1), (135, 1), (145, 1), (185, 1), (195, 1), (230, 1)]

number of peaks : 11
[(45, 1), (55, 1), (80, 1), (90, 1), (105, 1), (140, 1), (175, 1), (190, 1), (200, 1), (225, 1), (235, 1)]
```


XCorr 구현 실습



XCorr 구현 실습

- **XCorr**
 - HiXCorr version으로 코드 수정하기
 - experimental peaks 리스트의 배열과 theoretical peaks 리스트의 배열을 사용

XCorr 구현 실습

- 해상도 변경
 - 코드 상단 부분의 해상도 값 변경하여 실행 (F5)
 - 1
 - 0.1
 - 0.01
 - 0.001

XCorr 구현 실습

• 해상도에 따른 시간 비교

1. XCorr Crux(fast SEQUEST) version $\Theta(\tau n)$
2. XCorr Comet version $\Theta(\tau + n)$
3. XCorr HiXCorr version $\Theta(p)$

해상도		1	2	3
	1	0.128 s	0.03 s	0.03 s
	0.1	0.997 s	0.04 s	0.03 s
	0.01	10.590 s	0.24 s	0.03 s
	0.001	114.08 s	1.52 s	0.03 s
	0.0001	1077.57 s	16.12 s	0.03 s