

118-[PF] - 실습 - Python으로 인슐린 분석 준비

Python으로 인슐린 분석 준비

실습 개요

정보 기술 분야에서 Python은 문자열, 시퀀스, 숫자를 조작할 때 프로그래밍 언어로서 잘 작동합니다. Python은 특히 물리학, 화학, 생물학 같은 과학 컴퓨팅 응용 분야에서 선호되고 있습니다.

일부 Python 모듈 실습에서는 인체에서 혈당을 조절하는 것으로 잘 알려진 호르몬인 인간 인슐린에 대해 단순한 시퀀스 조작 및 계산을 수행합니다.

본 실습에서는 다음을 수행합니다.

- 인간 프리프로인슐린에서 인간 인슐린의 단백질 서열 검색

예상 완료 시간

30 분

AWS Cloud9 IDE 액세스

1. 이 지침의 상단으로 이동한 다음 **Start Lab**을 선택하여 실습 환경을 시작합니다.

Start Lab 패널이 열리고 실습 상태가 표시됩니다.

2. *Lab status: ready*라는 메시지가 표시되면 **X**를 선택하여 **Start Lab** 패널을 닫습니다.
3. 지침의 맨 위에서 **AWS**를 선택합니다.

새 브라우저 탭에서 AWS 관리 콘솔이 열립니다. 시스템에 자동으로 로그인됩니다.

참고: 새 브라우저 탭이 열리지 않는 경우 일반적으로 브라우저에서 팝업 창을 열 수 없음을 나타내는 배너 또는 아이콘이 브라우저 상단에 표시됩니다. 배너 또는 아이콘을 선택하고 **Allow pop ups**를 선택합니다.

4. AWS 관리 콘솔에서 **Services > Cloud9**을 선택합니다. **Your environments** 패널에서 **reStart-python-cloud9** 카드를 찾아 **Open IDE**를 선택합니다.

AWS Cloud9 환경이 열립니다.

참고: *.c9/project.settings have been changed on disk*라는 메시지가 담긴 팝업 창이 표시되면 **Discard**를 선택하여 무시합니다. 마찬가지로, *Show third-party content*라는 대화 창이 나타나면 **No**를 선택하여 거절합니다.

Python 연습 파일 생성

5. 메뉴 모음에서 **File > New From Template > Python File**을 선택합니다.

이 작업은 제목이 없는 파일을 생성합니다.

6. 템플릿 파일에서 샘플 코드를 삭제합니다.
7. **File > Save As...**를 선택하고, 연습 파일에 적절한 이름(예: *analyze-insulin.py*)을 입력한 다음 **/home/ec2-user/environment** 디렉터리에 저장합니다.

터미널 세션에 액세스

8. AWS Cloud9 IDE 에서 + 아이콘을 선택하고 **New Terminal** 을 선택합니다.

터미널 세션이 열립니다.

9. 현재 작동 중인 디렉토리를 표시하려면 `pwd` 를 입력합니다. 이 명령은 `/home/ec2-user/environment` 를 가리킵니다.

10. 이 디렉터리에서 이전 섹션에서 생성한 파일을 찾을 수 있어야 합니다.

연습 1: 인간 프리프로인슐린에서 단백질 서열 검색

National Center for Biotechnology Information(NCBI)에는 여러 생물학적 서열에 대한 정보가 있습니다.

11. NCBI(<https://ncbi.nlm.nih.gov>)에 액세스합니다.

12. 검색 창 옆에 있는 드롭다운 메뉴를 선택하고 **Protein** 을 선택합니다. 그런 다음, 검색 창에 `human insulin` 을 입력하고 **Search** 를 선택합니다.



13. 검색 결과에서 **insulin [Homo sapiens]**를 선택합니다.

☐ [insulin \[Homo sapiens\]](#)

3. 110 aa protein

Accession: AAA59172.1 GI: 386828

[Nucleotide](#) [PubMed](#) [Taxonomy](#)

[GenPept](#) [Identical Proteins](#) [FASTA](#) [Graphics](#)

14. 검색 레코드의 하단에서 첫 단어가 **ORIGIN**, 끝 단어가 **//**인 인슐린 서열을 복사합니다.

```
ORIGIN
    1 malwmrllpl lallalwgpdpaaafvnqhl cgshlvealy lvcgergffy tpktrreaed
    61 lqvgqvelgg gpgagslqpl alegslqkrg iveqcctsic slyqlenycn
//
```

15. AWS Cloud9 IDE의 탐색 창에서 **File > New File**을 선택하고 *preproinsulin-seq.txt*로 파일을 저장합니다.
16. 인슐린 서열을 **preproinsulin-seq.txt**에 붙여 넣습니다.

```
ORIGIN
    1 malwmrllpl lallalwgpdpaaafvnqhl cgshlvealy lvcgergffy
tpktrreaed
    61 lqvgqvelgg gpgagslqpl alegslqkrg iveqcctsic slyqlenycn
//
```

보너스: 프로그래밍 방식으로 *preproinsulin-seq.txt* 정리

소스 데이터 파일을 삭제하는 것은 컴퓨터 프로그래밍에서 일반적인 태스크입니다. 프로그래밍 방식으로 다양한 방법을 사용하여 *preproinsulin-seq.txt*를 정리할 수 있습니다. 예를 들어 Bash, Python 또는 원하는 프로그래밍 언어를 사용할 수 있습니다. 정규식을 사용하여 프로그래밍 방식으로 파일에서 *ORIGIN*, 해당 숫자, 2개의 슬래시(/), 공백, 줄 바꿈을 제거하거나 되돌리기 하십시오. 프로그래밍 방식으로 파일이 110 자인지 확인할 수도 있습니다.

연습 2: 인간 인슐린에서 단백질 서열 획득

인슐린은 일련의 잘라내어 붙여 넣기 절차를 통해 프리프로인슐린에서 얻어집니다. 프리프로인슐린에는 24aa 신호 서열과 86aa 프로인슐린 분자가 들어 있습니다. 아미노산 25–54 및 아미노산 90–110은 처리된 인슐린 분자입니다.

Python 또는 Bash 를 사용하거나 수동으로 조작하여 인슐린을 구성하는 서열에서 아미노산만 검색합니다.

17. 수동으로 또는 프로그래밍 방식으로 *ORIGIN, 1, 61, //* 공백 및 되돌리기를 삭제합니다.
18. AWS Cloud9 IDE 의 탐색 창에서 **File > New File** 을 선택하고 *preproinsulin-seq-clean.txt* 로 파일을 저장합니다.
19. **preproinsulin-seq-clean.txt** 파일에서 결과를 복사합니다.
20. 파일이 인간 프리프로인슐린 서열에서 아미노산을 나타내는 소문자 110 자로 되어 있는지 확인합니다.
21. AWS Cloud9 IDE 의 탐색 창에서 **File > New File** 을 선택하고 *lsinsulin-seq-clean.txt* 로 파일을 저장합니다.
22. **lsinsulin-seq-clean.txt** 에서 아미노산 1–24 를 저장합니다. 파일이 24 자인지 확인합니다.
23. AWS Cloud9 IDE 의 탐색 창에서 **File > New File** 을 선택하고 *binsulin-seq-clean.txt* 로 파일을 저장합니다.
24. **binsulin-seq-clean.txt** 에서 아미노산 25–54 를 저장합니다. 파일이 30 자인지 확인합니다.
25. AWS Cloud9 IDE 의 탐색 창에서 **File > New File** 을 선택하고 *cinsulin-seq-clean.txt* 로 파일을 저장합니다.
26. **cinsulin-seq-clean.txt** 에서 아미노산 55–89 를 저장합니다. 파일이 35 자인지 확인합니다.
27. AWS Cloud9 IDE 의 탐색 창에서 **File > New File** 을 선택하고 *ainsulin-seq-clean.txt* 로 파일을 저장합니다.
28. **ainsulin-seq-clean.txt** 에서 아미노산 90–110 를 저장합니다. 파일이 21 자인지 확인합니다.

자동화 시점과 수동으로 작업할 시점 결정: 범위와 시간에 대한 논의

컴퓨터 프로그래머에게 작업을 자동화하느냐 혹은 수동으로 처리하느냐는 문제는 딜레마입니다. 너무 많이 자동화하면 코딩에 시간을 낭비하게 되고, 너무 적게 자동화하면 프로그램의 범위가 제한될 수 있습니다. 코딩에 소요되는 시간을 최소화하면서 프로그램을 최대한의 범위로 생성하기 위해 자동화와 수동 작업의

균형을 맞추도록 노력하십시오. 이 같은 경우, 프로그래밍 방식으로 insulin-seq.txt 를 insulin-seq-clean.txt 로 정리하기 위해 추가 코딩 시간을 소비하는 것은 별 소용이 없을 것입니다. 그러나 수천 또는 수만 개의 파일을 다운로드해야 할 때 동일한 작업을 수행한다면 자동화를 고려하는 것이 좋습니다.

축하합니다! 추가 처리를 위해 데이터를 준비했습니다. 수동으로 이러한 파일을 준비해 봄으로써 Python 에서 제공하는 자동화에 대해 알아볼 수 있었습니다.

실습 종료

축하합니다! 실습을 마치셨습니다.

29. 이 페이지의 상단에서 **End Lab** 을 선택한 다음 Yes 를 선택하여 실습 종료를 확인합니다.

*DELETE has been initiated... You may close this message box now.*라는 내용의 패널이 표시됩니다.

30. *Ended AWS Lab Successfully* 라는 메시지가 잠시 표시되어 실습이 종료되었음을 나타냅니다.

추가 리소스

AWS Training and Certification 에 대한 자세한 내용은 <https://aws.amazon.com/training/>을 참조하십시오.

여러분의 피드백을 환영합니다. 제안이나 수정 사항을 공유하려면 AWS Training and Certification Contact Form 에서 세부 정보를 제공해 주십시오.

© 2022 Amazon Web Services, Inc. 및 계열사. All rights reserved. 본 내용은 Amazon Web Services, Inc.의 사전 서면 허가 없이 전체 또는 일부를 복제하거나 재배포할 수 없습니다. 상업적인 복제, 대여 또는 판매는 금지됩니다.