

EN-US



Preparing to Analyze Insulin with Python

Lab overview

In information technology, Python works well as the programming language of choice for manipulating strings, sequences, and numbers. Python is especially preferred in scientific computing applications such as physics, chemistry, and biology.

In some of the labs for the Python modules, you will perform simple sequence manipulations and calculations on human insulin, which is a well-known hormone in the human body that is responsible for regulating sugars.

In this lab, you will:

- Retrieve the protein sequence of human insulin from human preproinsulin

Estimated completion time

30 minutes

Accessing the AWS Cloud9 IDE

1. Start your lab environment by going to the top of these instructions and choosing **Start Lab**.

A **Start Lab** panel opens, displaying the lab status.

2. Wait until you see the message *Lab status: ready*, and then close the **Start Lab** panel by choosing the X.
3. At the top of these instructions, choose **AWS**.

Activate Windows
Go to Settings to activate Windows.

EN-US

2. Wait until you see the message *Lab status: ready*, and then close the **Start Lab** panel by choosing the X.
3. At the top of these instructions, choose **AWS**.

The AWS Management Console opens in a new browser tab. The system automatically logs you in.

Note: If a new browser tab does not open, a banner or icon at the top of your browser typically indicates that your browser is preventing the site from opening pop-up windows. Choose the banner or icon, and choose **Allow pop ups**.

4. In the AWS Management Console, choose **Services > Cloud9**. In the **Your environments** panel, locate the **reStart-python-cloud9** card, and choose **Open IDE**.
The AWS Cloud9 environment opens.

Note: If a pop-up window opens with the message *.c9/project.settings have been changed on disk*, choose **Discard** to ignore it. Likewise, if a dialog window prompts you to *Show third-party content*, choose **No** to decline.

Creating your Python exercise file

5. From the menu bar, choose **File > New From Template > Python File**.

This action creates an untitled file.

6. Delete the sample code from the template file.

7. Choose **File > Save As...**, and provide a suitable name for the exercise file (for example, *analyze-insulin.py* and save it under the */home/ec2-user/environment* directory.

Go to Settings to activate Windows.

EN-US



Accessing the terminal session

8. In your AWS Cloud9 IDE, choose the + icon and select **New Terminal**.

A terminal session opens.

9. To display the present working directory, enter `pwd`. This command points to `/home/ec2-user/environment`.

10. In this directory, you should also be able to locate the file you created in the previous section.

Exercise 1: Retrieving the protein sequence of human preproinsulin

The National Center for Biotechnology Information (NCBI) has information on many biological sequences.

11. Access **NCBI** at <https://ncbi.nlm.nih.gov>.

12. Next to the search bar, choose the dropdown menu and select **Protein**. Next, in the search bar, enter `human insulin` and choose **Search**.

The screenshot shows the NCBI homepage. At the top, there is a blue header bar with the NCBI logo, the word "Resources", and a "How To" link. Below the header, there is a search bar with a dropdown menu set to "Protein". The search term "human insulin" is entered into the search bar. Below the search bar, there are two buttons: "Create alert" and "Advanced".

13. Choose the following search result: **insulin [Homo sapiens]**.

[insulin \[Homo sapiens\]](#)

3. 110 aa protein

Accession: AAA59172.1 GI: 386828

blastp blastn tblastn

Activate Windows
Go to Settings to activate Windows.

EN-US

13. Choose the following search result: **insulin [Homo sapiens]**.

[insulin \[Homo sapiens\]](#)

3. 110 aa protein

Accession: AAA59172.1 GI: 386828

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

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

14. At the bottom of the search record, copy the insulin sequence, which starts with the word **ORIGIN** and ends with **//**.

ORIGIN

```
1 malwmrllpl lallalwgpd paaafvnqhl cgshlvealy lvcgergffy tpktrreaed
61 lqvgqvelgg gpgagslqlpl alegslqkrg iveqcctsic slyqlenyncn
```

//

15. In the AWS Cloud9 IDE, on the navigation pane, choose **File > New File** and save the file as *preproinsulin-seq.txt*.

16. Paste the insulin sequence into **preproinsulin-seq.txt**:

```
ORIGIN
1 malwmrllpl lallalwgpd paaafvnqhl cgshlvealy lvcgergffy tpktrreaed
61 lqvgqvelgg gpgagslqlpl alegslqkrg iveqcctsic slyqlenyncn
//
```

Activate Windows
Go to Settings to activate Windows.

Bonus: Cleaning *preproinsulin-seq.txt* programmatically

Cleaning source data files is a common task in computer programming. You could programmatically clean *preproinsulin-seq.txt* in several ways—for example, by using Bash, Python, or another programming language of choice. Try using regex to programmatically strip the file of *ORIGIN*, its numbers, the two slashes (//), spaces, and line

EN-US



Bonus: Cleaning preproinsulin-seq.txt programmatically

Cleaning source data files is a common task in computer programming. You could programmatically clean preproinsulin-seq.txt in several ways—for example, by using Bash, Python, or another programming language of choice. Try using regex to programmatically strip the file of *ORIGIN*, its numbers, the two slashes (//), spaces, and line breaks or return carriages. You could also confirm programmatically that the file has 110 characters.

Exercise 2: Obtaining the protein sequence of human insulin

Insulin is obtained from preproinsulin through a series of cut-and-paste procedures. Preproinsulin contains a 24aa signal sequence and an 86aa proinsulin molecule. Amino acids 25–54 and amino acids 90–110 are the processed insulin molecule. Use Python, Bash, or manual manipulation to retrieve only those amino acids in the sequence that compose insulin.

17. Manually or programmatically delete *ORIGIN*, 1, 61, //, and the spaces and return carriages.
18. In the AWS Cloud9 IDE, on the navigation pane, choose **File > New File** and save the file as *preproinsulin-seq-clean.txt*.
19. In the file **preproinsulin-seq-clean.txt**, copy your results.
20. Confirm that your file has 110 characters of lowercase letters, which represent the amino acids in the sequence of human preproinsulin.
21. In the AWS Cloud9 IDE, on the navigation pane, choose **File > New File** and save the file as *linsulin-seq-clean.txt*.
22. In **linsulin-seq-clean.txt**, save amino acids 1–24. Verify that your file has 24 characters.
23. In the AWS Cloud9 IDE, on the navigation pane, choose **File > New File** and save the file as *binsulin-seq-clean.txt*.
24. In **binsulin-seq-clean.txt**, save amino acids 25–54. Verify that your file has 30 characters.
25. In the AWS Cloud9 IDE, on the navigation pane, choose **File > New File** and save the file as *cinsulin-seq-clean.txt*.
26. In **cinsulin-seq-clean.txt**, save amino acids 55–89. Verify that your file has 35 characters.

Activate Windows
Go to Settings to activate Windows.

EN-US

27. In the AWS Cloud9 IDE, on the navigation pane, choose **File > New File** and save the file as *ainsulin-seq-clean.txt*.
28. In **ainsulin-seq-clean.txt**, save amino acids 90–110. Verify that your file has 21 characters.

Deciding when to automate and when to work manually: A discussion about scope versus time

Automating your work versus working manually is a dilemma for computer programmers. Too much automation wastes time on coding, whereas too little restricts the scope of your program. Try to balance your automation with working manually in an effort to create a program with the most scope for the least time spent coding. In this case, it is probably not worth the extra coding time to programmatically clean *insulin-seq.txt* to *insulin-seq-clean.txt*. However, if you needed to download thousands or millions of files and do the same task, automation would be good to explore.

Congratulations! You have prepared data for further processing. Manually preparing these files should help you appreciate the automation that Python can provide.

End Lab

☒ Congratulations! You have completed the lab.

29. Choose ■ **End Lab** at the top of this page, and then select Yes to confirm that you want to end the lab.

A panel indicates that *DELETE has been initiated... You may close this message box now.*

30. A message *Ended AWS Lab Successfully* is briefly displayed, indicating that the lab has ended.

Additional Resources

For more information about AWS Training and Certification, see <https://aws.amazon.com/training/>.

Your feedback is welcome and appreciated. If you would like to share any suggestions or corrections, please provide the details in our [AWS Training and Certification](#)

Activate Windows
Go to Settings to activate Windows.