bio334_day1_part2

May 13, 2025

1 Bio334 Practical Bioinformatics

The 2nd module, 14-16, May, 2025

1.1 Masaomi Hatakeyama

- GitHab https://github/masaomi/bio334_2025
- TAs: Narjes Yousefi, Kenji Yip Tong

2 Day1 Part2

Compare two lists

[ ]:
```python
import IPython.display
IPython.display.Audio("voice/day1_part2.mp3")
```

In this part, I will shortly introduce how to compare two lists. This technique will be applied to the comparison of two nucleotide sequences. Please look at the simple example below.

[ ]:
```python
list1 = [1, 2, 3, 4, 5, 6]
list2 = [1, 2, 0, 4, 5, 0]

for i in range(0, len(list1)):
    print("%s %s : " % (list1[i], list2[i]), end = "")
    if list1[i] == list2[i]:
        print("True")
    else:
        print("False")
```

[ ]:
```python
IPython.display.Audio("voice/day1_part2_example1_1.mp3")
```

This example compares two lists. More precisely speaking, it compares each element at the same position of the two lists.

If you do not understand what happens, then let's look at the code one by one for each element.

I would say, this is the deductive thinking way. To understand the code, deduction is important.

3 Compare two Lists

Break down the complicated part

`for i in range(0, len(list1)):`

Q: What does this do?

```
[ ]: IPython.display.Audio("voice/day1_part2_example1_2.mp3")
```

Probably you understand the first two lines. That is just definitions of two lists.

What happens in the third line with for-loop?

Let's break it down into smaller parts. Just please execute the following examples.

```
[ ]: print(len(list1))
```

```
[ ]: print(range(0, 6))
```

```
[ ]: for i in range(0, len(list1)):
         print(i)
```

```
[ ]: for i in [0,1,2,3,4,5]:
         print(i)
```

```
[ ]: IPython.display.Audio("voice/day1_part2_example1_3.mp3")
```

Hopefully, you have understood what happens.

This for-loop repeats the processes 6 times by incrementing the variable, *i*, starting from 0 to (the length of the list -1).

4 Compare two Lists

```
list1 = [1, 2, 3, 4, 5, 6]
list2 = [1, 2, 0, 4, 5, 0]

for i in [0, 1, 2, 3, 4, 5]:
  print("%s %s : " % (list1[i], list2[i]),
  end = "")
```

- What does the *print()* do?

The *print()* method just prints the each elements of the two lists

```
[ ]: list1 = [1, 2, 3, 4, 5, 6]
     list2 = [1, 2, 0, 4, 5, 0]

     for i in [0, 1, 2, 3, 4, 5]:
         print("%s %s : " % (list1[i], list2[i]), end = "")
```

5 Compare two Lists

```
print("%s %s : " % (list1[i], list2[i]), end = "")
```

- *print(“%s” % variable)*: shows the variable at the %s part
- *print(“XXX”, end=“ “)*: without line break

6 Compare two Lists

```
list1 = [1, 2, 3, 4, 5, 6]
list2 = [1, 2, 0, 4, 5, 0]

for i in range(0, len(list1)):
    if list1[i] == list2[i]:
```

- What does this do?

[ ]:

```
list1 = [1, 2, 3, 4, 5, 6]
list2 = [1, 2, 0, 4, 5, 0]

for i in range(0, len(list1)):
    if list1[i] == list2[i]:
        print(i, list1[i], list2[i], "True")
```

[ ]:

```
IPython.display.Audio("voice/day1_part2_example2_1.mp3")
```

This for-loop process compares each element of the two lists,

and if the elements at the same position are same, the print method show the elements

7 Compare two Lists

```
list1 = [1, 2, 3, 4, 5, 6]
list2 = [1, 2, 0, 4, 5, 0]

if list1[0] == list2[0]:
if list1[1] == list2[1]:
if list1[2] == list2[2]:
if list1[3] == list2[3]:
if list1[4] == list2[4]:
if list1[5] == list2[5]:
```

- Compare each element

[ ]:

```
IPython.display.Audio("voice/day1_part2_example2_2.mp3")
```

If you are confused with the abstracted processes by for-loop, please break it down without for-statement.

The process is actually same.

8 Compare two Lists

```
list1 = [1, 2, 3, 4, 5, 6]
list2 = [1, 2, 0, 4, 5, 0]

if list1[0] == list2[0]:
    print("True")
else:
    print("False")
```

- Repeated 6 times for each element

9 Compare two Lists

```
list1 = [1, 2, 3, 4, 5, 6]
list2 = [1, 2, 0, 4, 5, 0]

for i in range(0, len(list1)):
    print("%s %s: %s " % (list1[i], list2[i], list1[i] == list2[i])
```

- What is the difference?

```
[ ]: list1 = [1, 2, 3, 4, 5, 6]
     list2 = [1, 2, 0, 4, 5, 0]

     for i in range(0, len(list1)):
         print("%s %s: %s " % (list1[i], list2[i], list1[i] == list2[i]))
```

10 Mini-Summary

generalization

In order to compare the elements in **two lists**, use **for** and **if**

```
for <variable> in range(0, len(<list>)):
  if <compare elements> :
    # process for each condition
```

```
[ ]: IPython.display.Audio("voice/summary_day1_part2.mp3")
```

Just let me summarize this part.

I showed an example to compare two lists. In order to do that, I used the for-loop and if-branching flow.

This pattern will be used a lot in other situations in the case of using the *List* data structure.

Here, I can come back to the tips. you can generalize the idea.

When you need to compare two lists, you can use this pattern, for-loop and if-branching.

Now you have got the generalized idea to rewrite a lot of similar repeated processes into one or two code lines.

It is the generalization.

Then, let's do the exercise.