bio334_day2_part5

May 13, 2025

1 Bio334 Practical Bioinformatics

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

1.1 Masaomi Hatakeyama

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

2 Day2 Part5

Batch process, module, and *A. kamchatica* homoeolog Tajima's D

[ ]:
```
import IPython.display
IPython.display.Audio("voice/day2_part5.mp3")
```

As the final part today, I will give you another technique to improve reusability again, before introducing the exercise.

But if you have not finished the Tajima's D implementation yet, please focus on and continue it.

In the actual situation, normally, we calculate a lot of samples, not only one dataset of sequences.

In such a situation, we have to use the same programming code many times for different datasets. You could imagine 100 times executions of the same programming code.

There is a typical way to run 100 jobs at once. It is called the batch process.

I will show you two examples to do the batch process here. The first example uses a shell script, another example uses a Python script.

3 Why programming?

- **Reusability**: Reproducibility, you can calculate with just one command type again
- **Batch process**: Automation, computer can work while you are sleeping
- **Understanding**: It helps you to learn algorithms
- **Computer simulation**: testing many possible conditions/assumptions

You will learn what the batch process is in this part

4 Batch process (Shell script)

```
#!/bin/sh
python script.py job1
python script.py job2
python script.py job3
```

- Executing several jobs sequentially

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

The first example is using a shell script to run three different jobs but it uses the same Python script.

In other words, once you make a shell script, a lot of Python jobs can run at once.

The difference in each job is the input FASTA file.

5 Example1

batch shell script

```
$ cat day2_5_example1.sh
python day2_1_exercise1.py athal_chr1.fa
python day2_1_exercise1.py athal_chr2.fa
python day2_1_exercise1.py athal_chr3.fa
```

Execution example

```
$ sh day2_5_example1.sh
genome size = 30427671 bp
genome size = 19698289 bp
genome size = 23459830 bp
```

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

The second example is using a Python script to do the same thing as the first example.

In order to execute a Python script in another Python script, you can use *os.system()* method.

This method can call an external Unix command in a Python script.

6 Example2

batch python script

```
$ cat day2_5_example2.py
import glob
import os
for file in glob.glob("athal_chr*.fa"):
  command = "python3 day2_1_exercise1.py " + file
  os.system(command)
```

Execution

```
$ python3 day2_5_example2.py
genome size = 30427671 bp
genome size = 19698289 bp
genome size = 23459830 bp
```

7 *os* module

- Execute an external command (in a Python script)

```
import os
os.system("pwd")
```

8 *glob* module

Get a file path (matching a pattern)

```
import glob
glob.glob("dir/*.fa")
```

returns a List object that has all fasta file paths in directory

9 Mini-Summary

- **Batch process**: several jobs (commands) run at once
- The simplest way is to use a shell script
- It can be done also in a Python script

which means that it is possible to call a Python script in another Python script

[ ]:
```
IPython.display.Audio("voice/exercise_day2_part5.mp3")
```

I will explain the final exercise today.

It is the real data published in 2016. The target species is *Arabidopsis kamchatica*. *Arabidopsis kamchatica* is an allotetraploid species that is polyploidized from *Arabidopsis halleri*, diploid, and *Arabidopsis lyrata*, diploid.

So, *Arabidopsis kamchatica* has two homologous gene sets derived from different species. Such a gene is called **homeolog**.

Arabidopsis halleri is called hyperaccumulator and it can live in a highly toxic heavy metal environment such as mine, while *Arabidopsis lyrata* cannot live in such a heavy metal environment.

We investigated the nucleotide diversity around HMA4 gene, heavy metal ATP-ase4, in *Arabidopsis kamchatica*, particularly, compared the nucleotide diversity of the sub-genomes derived from *Arabidopsis halleri* and *lyrata*.

Samples were collected mainly from Japan but including Alaska in the north, and Taiwan in the south.

In this exercise, you calculate the nucleotide diversity of several genes around HMA4 gene in each subgenome.

The gene sequences have been already aligned and saved in different files. So, you have to run several jobs using the same Python code.

Please integrate and assemble all the knowledge that you have learned in this module, and solve the problem.

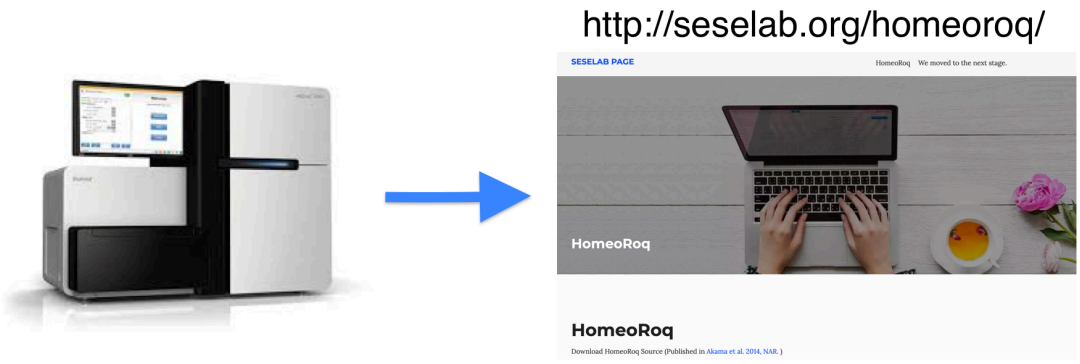
10 *Arabidopsis kamchatica*

11 Samples

12 Homoeolog

Duplicated gene pair

13 Pipeline HomeoRoq



- http://seselab.org/homeoroq/
- Note: HomeoRoq is obsolete. Now we switched to use EAGLE-RC
- Reference: Tim, Masa, et. al, MBE., 2016

14 Appendix, module

14.0.1 Another tips to use an external python code

You can call functions defined in an external Python file, which is called *module*.

```
my_module.py
    def hello():
        print("Hello, world")

main1.py
```

```
    import my_module
    my_module.hello()

main2.py
    import my_module as m       # you can rename the module
    m.hello()

main3.py
    from my_module import hello as gruezie # you can remane a function
    gruezie()


result
    $ python main1.py
    Hello, World

    $ python main2.py
    Hello, World

    $ python main3.py
    Hello, World
```