bio334_day3

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 Day3 Final

VCF file, Dictionary, and nucleotide diversity and Tajima's D

[ ]:
```
import IPython.display
IPython.display.Audio("voice/day3_final.mp3")
```

Today, the final session in this module, I will give you a more advanced exercise, but if you have not finished the exercise of calculating Tajima's D yet, please continue it.

If everything is all clear until yesterday, then please go on the final exercise today.

3 Advanced exercise1

Calculate nucleotide diversity of *A. thaliana genomes* http://www.1001genomes.org

Use only chromosome1 of MPICWang2013 accessions

[ ]:
```
IPython.display.Audio("voice/day3_exercise.mp3")
```

This is also real data that has been already published. It is the 1001 genome project on Arabidopsis thaliana.

You can get a lot of SNP data trom the 1001 project site. The SNP data are uploaded as follows.

4 SNP file format

```
#samp chr     pos ref sub qual
108   Chr1    83  T   C   40
108   Chr1    92  A   C   40
108   Chr1    262 C   G   40
```

- The length of Chr1: 30,427,671 [bp]

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

As for the SNP data, as a defacto standard, we use VCF file format, VCF: Variant Calling Format.

It is a tab-separated value format. Each column is separated by tab space.

The first line shows the column header.

The first column is the sample name or accession ID, the second column is the chromosome number, the third column is the nucleotide position in the chromosome, the fourth column is the reference nucleotide, and the fifth column is the alternative nucleotide in that accession.

So, this is not the FASTA format that we used until yesterday.

It contains only the SNP information, in other words, position information and alternative nucleotide information, in the VCF file.

The basic idea is the same as before, but the implementation should change to adjust to the file format.

Namely, you have to change, how to count the segregating sites, how to compare each accession, how to count the pairwise snip based on this file format.

It is the final goal of this module to think about the algorithm by yourself. You should have all pieces of the code, and you understand the nucleotide diversity and Tajima's D and how to calculate it mathematically. The problem in programming should be how to construct the algorithm by assembling the knowledge.

5 Dictionary

- Similar to List
- Curly brackets for the entire set of elements
- Square brackets for each elements
- Index does not have to be sequential integer

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

There are several solutions, but I just give you one hint.

If you use *Dictionary* data structure, it could rather simply deal with the VCF file data.

The *Dictionary* is similar to *List* object.

The *List* has an index and value, but the *Dictionary* has **key** and **value**.

The key can be not only an *Integer* but a *String* or any object.

Please look at the example below.

[ ]:
```
# Dictionary
# Example

dic = {}
```

```
dic[1] = "aaa"
dic["a"] = 111
print(dic) # => {1: 'aaa', 'a': 111}
print(dic.keys()) # => [1, 'a']
print(dic.values()) # => ['aaa', 111]
print(dic.items()) # => [(1, 'aaa'), ('a', 111)]
```

[ ]:
```
# day3_example1.py

import glob

samples = []
for file_name in glob.glob("*.vcf"):
    f = open(file_name)
    print(file_name)
    snps = {}
    for line in f:
        print(line, end="")
        pos = line.split()[2]
        nuc = line.split()[4]
        snps[pos] = nuc
    f.close()
    samples.append(snps)
    print()

print("samples=", samples)
```

Position	sample1	sample2	sample3	SNP?
1	A	A	A	No
2	N	N	N	No?
3	G	N	G	Yes?
4	C	T	T	Yes

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

Now you can use the *Dictionary* data structure to detect a SNP from a VCF file.

Let's look at the example above. Assuming that this is a part of the same chromosome. There is position information in the first column, and three sample information.

Look at the first and second positions. These are not SNPs, because sample1, sample2, and sample3 have the same nucleotide information.

N means uncertain nucleotide due to some sequencing error, low quality, or something like that.

The third and fourth positions are SNP sites, because one of them is different from the others.

How do you implement this SNP detection in Python?

Actually, there are several ways to check whether it is SNP or not.

I give you one hit, and the *Set* operation may help you below.

6 Set operation

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

Set is another data structure.

It is also similar to *List* object, but two important things.

It does not have an index, and it does not allow the same elements in it. Please look at the example below.

List can be converted into a *set* object by *set()* function.

And then the duplicated elements become only one in the *Set* object.

I just give you another hint, a little bit more realistic example in the next cell, but I stop the explanation. Please look at the code and think about what it does.

[ ]:
```
list1 = [1,1,2,3,3,4]
set1 = set(list1)
print(set1)
```

[ ]:
```
# day3_example2.py

snps1 = {1: 'A', 3: 'G', 4: 'C'}
snps2 = {1: 'A', 4: 'T'}

print("Sample1: ", snps1)
print("Sample2: ", snps2)
pos1 = set(snps1.keys())
pos2 = set(snps2.keys())
common_pos = pos1 & pos2
exclor_pos = pos1 ^ pos2

diff = len(exclor_pos)
for pos in common_pos:
    if not snps1[pos] == snps2[pos]:
        diff += 1

print("The number of segregating sites = ", diff)
```

7 Final exercise

Nucleotide diversity & Tajima's D from VCF file

- https://gist.github.com/masaomi/1397a32c4b870f7ab7e92f479770788d

8 Optional exercise1

- If you completely understand everything, let's generate the final exam problem using an LLM tool in markdown format.
  - Please send me the link to your GitHub repository.
  - If it's a good problem, I might use it.

9 Optional Exercise 2

- You may abel to modify and update the teaching materials in a more appropriate way using an LLM or AI agent for next year's lecture.
  - Please send me the link to your GitHub repository.
  - If it's a good problem, I might use it and credit you by name.