Pristian Budi Dharmawan_2501983105_LA20 – BB20_Computational Biology_Session 2_DNA Composition Analysis

Exercise Session 2

Google Collab Link: Click Here or
https://colab.research.google.com/drive/1pYOEngPHNn6k5Pr98QaPGksjEo3nDfIk?usp=sharing

A. Exercise PPT Example

Tasks

1. Which of the following sequence will have the highest GC?
2. Which of the sequence will have the higher melting temperature?
3. Which of the sequence will have the highest Molecular Weight?

Sequence

- Sequence 1 = 'ATGCATGGTGCGCGA'
- Sequence 2 = 'ATTTGTGCTCCTGGA'

Answer

Basic installation

```
%pip install biopython
from Bio.Seq import Seq
from Bio.SeqUtils import GC
from Bio.SeqUtils import MeltingTemp as mt
from Bio.SeqUtils import molecular_weight
```

```
def at_content(seq):
    result = float(seq.count('A') + seq.count('T'))/len(seq) * 100
    return result
```

```
ex1 = Seq('ATGCATGGTGCGCGA')
ex2 = Seq('ATTTGTGCTCCTGGA')
```

```
def get_metrics(seq):
    gc = GC(seq)
    at = at_content(seq)
    melting = mt.Tm_GC(seq)
    weight = molecular_weight(seq)
    result = "GC:{}, AT:{}, Temp:{}, Weight:{}".format(gc,at,melting,weight)
    return result
```

Steps

GC Value

```
[27]
gc_ex1 = GC(ex1)
gc_ex1
✓ 0.0s
... 60.0
```

```
[6]
gc_ex2 = GC(ex2)
gc_ex2
✓ 0.0s
... 46.666666666666664
```

AT Value

```
[7]
at_ex1 = at_content(ex1)
at_ex1
✓ 0.1s
... 40.0
```

```
[8]
at_ex2 = at_content(ex2)
at_ex2
✓ 0.0s
... 53.333333333333336
```

Melting Temperature

```
[9]
mt.Tm_Wallace(ex1)
mt_ex1 = mt.Tm_GC(ex1)
mt_ex1
✓ 0.0s
... 44.5029020719779
```

```
[10]
mt.Tm_Wallace(ex2)
mt_ex2 = mt.Tm_GC(ex2)
mt_ex2
✓ 0.0s
... 39.03623540531123
```

Molecular Weight

```
[11]
mw_ex1 = molecular_weight(ex1)
mw_ex1
✓ 0.0s
... 4712.995199999999
```

```
[12]
mw_ex2 = molecular_weight(ex2)
mw_ex2
✓ 0.0s
... 4653.9565
```

Final Answer

```
[13]
get_metrics(ex1)
✓ 0.0s
... 'GC:60.0, AT:40.0, Temp:44.5029020719779, Weight:4712.995199999999'
```

```
[14]
get_metrics(ex2)
✓ 0.0s
... 'GC:46.666666666666664, AT:53.333333333333336, Temp:39.03623540531123, Weight:4653.9565'
```

Answer for the tasks

```python
# 1. Which of the following sequence will have the highest GC?
if gc_ex1 > gc_ex2:
    print("GC Value of Seq 1 is bigger than Seq 2")
else:
    print("GC Value of Seq 2 is bigger than Seq 1")
```

[15] ✓ 0.0s

```
GC Value of Seq 1 is bigger than Seq 2
```

```python
# 2. Which of the sequence will have the higher melting temperature
if mt_ex1 > mt_ex2:
    print("Melting Temperature Value of Seq 1 is bigger than Seq 2")
else:
    print("Melting Temperature Value of Seq 2 is bigger than Seq 1")
```

[16] ✓ 0.0s

```
Melting Temperature Value of Seq 1 is bigger than Seq 2
```

```python
# 3. Which of the sequence will have the highest Molecular Weight
if mw_ex1 > mw_ex2:
    print("Seq 1 is heavier than Seq 2")
else:
    print("Seq 2 is heavier than Seq 1")
```

[17] ✓ 0.0s

```
Seq 1 is heavier than Seq 2
```

B. Exercise WGS Example

Tasks

1. Calculate the temperature of AT and CG
2. Calculate the molecular weight of AT and CG

Sequence

- Sequence = https://www.ncbi.nlm.nih.gov/genbank/examples.wgs/#partialcds

Answer

Basic Installation

```python
%pip install biopython
from Bio.Seq import Seq
from Bio.SeqUtils import GC
from Bio.SeqUtils import MeltingTemp as mt
from Bio.SeqUtils import molecular_weight
```

```python
def at_content(seq):
    result = float(seq.count('A') + seq.count('T'))/len(seq) * 100
```

```
    return result
```

```
WGS =
Seq('TGcaaagtGGAATTCCAATTTCAACACCAGTTTTTGATGGCGCAAAAGAGCAAGATGTAACAAATATGTTAGAGCTTG
CATCATTACCAAAATCTGGTCAAACAAAATTGTGGGATGGTAGAACAGGTGAAAAATTTGATAGAGAAGTCACAGTTGGCACT
ATTTATATGTTAAAATTACACCATCTTGTAGAAGATAAAATACACGCAAGATCTACAGGTCCTTATAGTTTAGTTACACAACA
ACCTCTTGGTGGTAAGGCTCAATTGGGAGGTCAACGATTTGGAGAAATGGAAGTTTGGGCTCTGGAAGCTTATGGGGCTTCTT
ATACTTTACAAGAAATTTTAACAGTAAAATCTGATGATGTTGCTGGTAGAGTTAAAGTTTATGAAACAATAGTAAAAGGTGAA
GAGAATTTCGAGTCAGGAATACCTGAGTCATTTAATGTTTTAGTAAAAGAAATCAAAGCGCTAGCTCTTAATGTGGAGTTAAA
TTAAAATGAAAAAAGATATTAAAGATTTTTTTAAAGAAACTGCCATATCAGACTCTCAAAATTTTAATAGTATTAAAATTACT
TTAGCAAGCCCTGAAAAGATAAAGTCATGGACTTATGGAGAAATAAAAAAACCCGAAACTATTAATTATAGAACTTTCAGACC
TGAAAAAGACGGCCTATTTTGTGCGAGAATATTTGGTCCAATAAAAGATTACGAATGTTTATGTGGAAAATATAAAAGAATGA
AGTTCAGAGGAATTATTTGTGAGAAGTGTGGCGTAGAGGTTACTAAATCAAATGTTCGTAGAGAAAGAATGGGGCACATCAAT
TTATCAACCCCAGTTGCACATATTTGGTTTTTAAAATCTTTACCAAGTAGAATTTCACTAGCTATTGATATGAAGCTTAAAGA
GGTTGAAAGAGTTCTATACTTTGAAAGTTTTATTGTTATAGAGCCTGGATTAACTAGTCTTAAAAAAAATCAACTTTTAAACG
AAGATGAATTAAATAAATATCAAGAGGAGTTTGGTGAAGAATCCTTTACTGCAGGAATAGGAGCAGAGGCGATACTAGAGATT
TTAAAATCTATAGACTTGAATAAAGAGAGAGAAATTTTATTAAAAAATATAAATGAGACAAAATCAAAGGTTGCTGAAGAAAG
ATCTATAAAAAGATTAAAACTGATCGATTCATTTATTGAAACTGGTAACAAACCAGAATGGATGATTTTAACTACTATACCTG
TAATACCACCAGAGTTAAGGCCACTTGTTCCTCTAGATGGAGGTAGATTTGCAACATCAGATCTAAACGATTTGTATAGAAGA
GTTATAAATAGAAATAATAGATTGAAAAGATTAATGGATCTTAAAGCTCCAGATATAATTATTAGAAATGAAAAACGAATGTT
GCAAGAGTCAGTGGATGCTTTATTCGATAATGGCAGAAGAGGCAGAGTAATTACAGGAACTGGTAAACGTCCATTAAAATCTT
TGGCTGAAATGCTTAAAGGAaaacaaG')
```

```
def get_metrics(seq):
    gc = GC(seq)
    at = at_content(seq)
    melting = mt.Tm_GC(seq)
    weight = molecular_weight(seq)
    result = "GC:{}, AT:{}, Temp:{}, Weight:{}".format(gc,at,melting,weight)
    return result
```

Steps

GC Value

```python
gc_WGS = GC(WGS)
gc_WGS
```

[22] ✓ 0.1s

```
32.05804749340369
```

AT Value

```python
at_WGS = at_content(WGS)
at_WGS
```

[23] ✓ 0.1s

```
67.34828496042216
```

Melting Temperature

```python
mt.Tm_Wallace(WGS)
mt_WGS = mt.Tm_GC(WGS)
mt_WGS
```

[24] ✓ 0.0s

```
72.65092318015732
```

Molecular Weight

```python
mw_WGS = molecular_weight(WGS)
mw_WGS
```

[25] ✓ 0.0s

```
471222.4697000047
```

Final Answer

```python
get_metrics(WGS)
```

[26] ✓ 0.0s

```
'GC:32.05804749340369, AT:67.34828496042216, Temp:72.65092318015732, Weight:471222.4697000047'
```