



# TIN SINH HỌC ĐẠI CƯƠNG (Introduction to Bioinformatics)

PGS.TS. Trần Văn Lăng  
Email: langtv@vast.vn

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY


1



## Chương 6: THƯ VIỆN BIOPYTHON

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

2



## NỘI DUNG

- Giới thiệu về gói thư viện BioPython
- Sử dụng BioPython

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

3



## GIỚI THIỆU VỀ THƯ VIỆN BIOPYTHON

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

4



- Biopython is a set of freely available tools for biological computation written in [Python](#) by an international team of developers.
- The latest release is [Biopython 1.73](#), released on 18 December 2018.

5



- Sau khi đã install Python, dùng các lệnh sau để install thêm Biopython:
  - pip install biopython (cho version 2.x)
  - pip3 install biopython (cho version 3.x)

6



- Để thực thi, chẳng hạn với python3:

```
macOS:python lang$ python3
>>> import Bio
>>> print( Bio.__version__ )
1.73
>>>
```

7



- Để sử dụng các sequence

```
>>> from Bio.Seq import Seq
>>> my_seq = Seq("ACGTACGT")
>>> print(my_seq)
ACGTACGT
>>>
```

8

## VỀ CÁC TRÌNH TỰ DNA, RNA, PROTEIN

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

9

9

## Trình tự bổ sung và bổ sung đảo ngược

- Tìm trình tự bổ sung của 1 trình tự DNA và trình tự bổ sung đảo ngược của nó

```
>>> my_seq = Seq("ACGTACGT")
>>> print( my_seq )
ACGTACGT
>>> print( my_seq.complement() )
TGCATGCA
>>> print( my_seq.reverse_complement() )
ACGTACGT
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

10

10

- Lưu ý, với trình tự trong my\_seq chưa đề cập đó là trình tự gì, nên khi cần biết về loại trình tự, dùng:

```
>>> my_seq.alphabet
Alphabet()
>>>
```

- Kết quả đây là trình tự các ký tự alphabetic thông thường trong bảng chữ cái Latin

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

11

11


- Vì vậy, để cho biết trình tự đưa vào là trình tự DNA, phải có thêm:

```
>>> from Bio.Alphabet import generic_dna
>>> my_seq = Seq("ACGTACGT", generic_dna )
>>> my_seq.alphabet
DNAAlphabet()
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

12

12




- Nếu là trình tự protein, phải có thêm

```
>>> from Bio.Alphabet import generic_protein
>>> my_seq = Seq( "ACGTACGT", generic_protein )
>>> my_seq.alphabet
ProteinAlphabet()
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY 13

13




### Cộng hai trình tự

- Cộng trên cùng loại trình tự:

```
>>> from Bio.Alphabet import generic_dna, generic_protein
>>> my_dna = Seq( "ACGTACGT", generic_dna )
>>> my_pro = Seq( "ACGTBAGAT", generic_protein )
>>> seq = my_dna + my_dna
>>> seq
Seq('ACGTACGTACGTACGT', DNAAlphabet())
>>> seq = my_pro + my_pro
>>> seq
Seq('ACGTBAGATACGTBAGAT', ProteinAlphabet())
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY 14

14




- Nhưng không thể cộng 2 loại với nhau

```
>>> seq = my_dna + my_pro
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/local/lib/python3.6/site-packages/Bio/Seq.py", line
345, in __add__
    self.alphabet, other.alphabet))
TypeError: Incompatible alphabets DNAAlphabet() and
ProteinAlphabet()
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY 15

15



- Tương tự như vậy, có trình tự RNA

```
>>> from Bio.Alphabet import generic_rna
>>> my_rna = Seq( "ACGTACGT", generic_rna )
>>> my_rna.alphabet
RNAAlphabet()
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY 16

16

## Tìm trình tự DNA



- Cho biết vị trí bắt đầu của một trình tự trong trình tự nào đó:

```
>>> my_dna = Seq( "ACGTATGCACTG", generic_dna )
>>> my_dna.find( "ATG" )
4
>>> my_dna.find( "ACGT" )
0
>>> my_dna.find( "ATC" )
-1
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

17

17

## Đếm số nucleotide hoặc oligomer



- Cho biết có bao nhiêu nucleotide trong trình tự my\_dna cho trước

```
>>> my_dna = Seq( "ACGTATGCACTG", generic_dna )
>>> my_dna.count( "A" )
3
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

18

18

## dimer (2), tetramer (3), pentamer (4), hexamer (5), ...



```
>>> my_dna = Seq( "ACGTATGCACTG", generic_dna )
>>> my_dna.count( "A" )
3
>>> my_dna.count( "AC" )
2
>>> my_dna.count( "ACG" )
1
>>> my_dna.count( "ACGT" )
1
>>> my_dna.count( "ATGCA" )
1
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

19

19

- Cũng có thể viết trực tiếp:

```
>>> "ACGTATGCACTG".count("AC")
2
>>> Seq("ACGTATGCACTG", generic_dna).count("AC")
2
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

20

20

## Trình tự bắt cặp của DNA



```
>>> from Bio.Seq import Seq
>>> from Bio.Alphabet import generic_dna
>>> my_dna = Seq("AGTACACTGGT", generic_dna)
>>> my_dna
Seq('AGTACACTGGT', DNAAlphabet())
>>> my_dna.complement()
Seq('TCATGTGACCA', DNAAlphabet())
>>> my_dna.reverse_complement()
Seq('ACCAAGTGTACT', DNAAlphabet())
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

21

21

## Phiên mã sang RNA



```
>>> my_dna = Seq("AGTACACTGGT", generic_dna)
>>> my_dna
Seq('AGTACACTGGT', DNAAlphabet())
>>> my_dna.transcribe()
Seq('AGUACACUGGU', RNAAlphabet())
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

22

22

## • Hoặc

```
>>> my_dna = Seq("AGTACACTGGT", generic_dna)
>>> my_rna = my_dna.transcribe()
>>> my_rna
Seq('AGUACACUGGU', RNAAlphabet())
>>> my_rna.back_transcribe()
Seq('AGTACACTGGT', DNAAlphabet())
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

23

23

## Dịch mã từ mRNA sang protein



```
• Dịch mã sang protein có stop codon:
>>> from Bio.Seq import Seq
>>> from Bio.Alphabet import generic_rna
>>> m_rna =
Seq("AUGGCCAUUGUAAUGGGCCGCUGAAAGGGUGCCCGAU
AG", generic_rna)
>>> m_rna.translate()
Seq('MAIVMGR*KGAR*', HasStopCodon(ExtendedIUPACProtein(),
'*'))
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

24

24



- Để dịch mã sang protein không có stop codon

```
>>> m_rna =
Seq("AUGGCCAUUGUAAUGGGCCGCUGAAAGGGUGCCCGAU
AG", generic_rna)
```

```
>>> m_rna.translate(to_stop=True)
Seq('MAIVMGR', ExtendedIUPACProtein())
```

- Để thay stop codon bằng một ký hiệu khác

```
>>> m_rna.translate(stop_symbol="@")
Seq('MAIVMGR@KGAR@',
HasStopCodon(ExtendedIUPACProtein(), '@'))
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

25

25



- Để dịch mã sang protein, nhưng lấy cho đến khi gặp stop codon thứ II:

```
>>> m_rna =
Seq("AUGGCCAUUGUAAUGGGCCGCUGAAAGGGUGCCCGAU
AG", generic_rna)
```

```
>>> m_rna.translate(table=2)
Seq('MAIVMGRWKGAR*',
HasStopCodon(ExtendedIUPACProtein(), '*'))
>>> m_rna.translate(table=2,to_stop=True)
Seq('MAIVMGRWKGAR', ExtendedIUPACProtein())
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

26

26

## Lưu ý



- Biopython sử dụng ký hiệu, cũng như danh pháp hóa học theo quy ước của **IUPAC** (*International Union of Pure and Applied Chemistry* - Liên minh Quốc tế về Hóa học thuần túy và Hóa học ứng dụng - <http://www.sbcs.qmul.ac.uk/iupac/>) với các đối tượng DNA, RNA và Protein.

–Nên từ đây về sau chúng ta luôn luôn chỉ định một bảng chữ cái rõ ràng, không nhập nhằng (unambiguous alphabet) khi tạo ra trình tự

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

27

27



- Khi đó, cần thực hiện các câu lệnh:

```
>>> import Bio
>>> from Bio.Seq import Seq
>>> from Bio.Seq import IUPAC
```

- Tạo ra trình tự DNA

```
>>> my_seq = Seq("AGTACACTGGT", IUPAC.unambiguous_dna)
>>> my_seq
Seq('AGTACACTGGT', IUPACUnambiguousDNA())
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

28

28

## Tính tỷ lệ phần trăm GC%



- Sử dụng hàm len() để lấy chiều dài của trình tự

```
>>> my_seq =  
Seq("GATCGATGGGCCTATATAGGATCGAAAATCGC",  
IUPAC.unambiguous_dna )  
>>> 100 * float(my_seq.count("G") + my_seq.count("C")) /  
len(my_seq)  
46.875  
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

29

29

- Hoặc sử dụng hàm GC()

```
>>> from Bio.SeqUtils import GC  
>>> GC( my_seq )  
46.875  
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

30

30

## Bóc tách trình tự



- Tách trình tự có màu tím

```
>>> my_seq =  
Seq("GATCGATGGGCCTATATAGGATCGAAAATCGC",  
IUPAC.unambiguous_dna )  
>>> my_seq[4:12]  
Seq('GATGGGCC', IUPACUnambiguousDNA())  
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

31

31

## Gộp các trình tự lại



- Tạo một List của các sequence (một tập hợp)

```
>>> seqs = [Seq("ACGT", IUPAC.unambiguous_dna),  
Seq("AACC", IUPAC.unambiguous_dna), Seq("GGTT",  
IUPAC.unambiguous_dna)]
```

- Khởi tạo trình tự kết quả sumSeq

```
>>> sumSeq = Seq( "", IUPAC.unambiguous_dna )  
>>> for s in seqs:  
...     sumSeq += s  
...  
>>> sumSeq  
Seq('ACGTAACCGGTT', IUPACUnambiguousDNA())  
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

32

32





- Cũng có thể sử dụng hàm sum()

```
>>> seqs = [Seq("ACGT", IUPAC.unambiguous_dna),
Seq("AACC", IUPAC.unambiguous_dna), Seq("GGTT",
IUPAC.unambiguous_dna)]
>>> sum( seqs, Seq("", unambiguous_dna) )
Seq('ACGTAACCGTT', IUPACUnambiguousDNA())
>>>
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

33

33

## Ví dụ với trình tự hoàn chỉnh CDS



- Trình tự hoàn chỉnh (complete coding sequence - CDS) là trình tự nucleotide có start codon, stop codon và bên trong đó không có stop codon.
- Chẳng hạn, sau đây là một CDS

**ATG**AAAAAGATGCAATCTATCGTACTCGCACTTTCCCTGGTTCT  
GGTCGCTCCCATGGCAGCACAGGCTGCGGAAATTACGTTAGT  
CCCGTCAGTAAAATTACAGATAGGCGATCGTGATAATCGTGCG  
TATTACTGGGATGGAGGTCAGTGGCGCGACCACGGCTGGTGG  
AAACAACATTATGAATGGCGAGGCAATCGCTGGCACCTACACG  
GACCGCCGCCACCGCCGCGCCACCATAAGAAAGCTCCTCATG  
ATCATCACGGCGGTCATGGTCCAGGCAAACATCACCGCT**TAA**

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

34

34



- Có thể dịch mã sang protein như sau:

```
>>> gene = Seq(
"ATGAAAAAGATGCAATCTATCGTACTCGCACTTTCCCTGGTTC
TGGTCGCTCCCATGGCAGCACAGGCTGCGGAAATTACGTTAG
TCCCGTCAGTAAAATTACAGATAGGCGATCGTGATAATCGTG
CTATTACTGGGATGGAGGTCAGTGGCGCGACCACGGCTGGT
GAAACAACATTATGAATGGCGAGGCAATCGCTGGCACCTACAC
GGACCGCCGCCACCGCCGCGCCACCATAAGAAAGCTCCTCAT
GATCATCACGGCGGTCATGGTCCAGGCAAACATCACCGCTAA"
, IUPAC.unambiguous_dna )
>>> print( gene.translate( to_stop=True ) )
MKKMQSIVLALSLVLVAPMAAQAAEITLVPSVKLQIGDRDNRGYY
WDGGHWRDHGWWKQHYEWRGNRWHLHGPPPPRHHKKAPH
DHHGGHGP GKHHR
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

35

35

## Ví dụ: gene yaaX của E.coli K12



- Trong mã di truyền của vi khuẩn, start codon là GTG. Chẳng hạn, với vi khuẩn K12, gene yaaX là trình tự:

**GTG**AAAAAGATGCAATCTATCGTACTCGCACTTTCCCTGGTTC  
TGGTCGCTCCCATGGCAGCACAGGCTGCGGAAATTACGTTAG  
TCCCGTCAGTAAAATTACAGATAGGCGATCGTGATAATCGTG  
CTATTACTGGGATGGAGGTCAGTGGCGCGACCACGGCTGGT  
GAAACAACATTATGAATGGCGAGGCAATCGCTGGCACCTACAC  
GGACCGCCGCCACCGCCGCGCCACCATAAGAAAGCTCCTCAT  
GATCATCACGGCGGTCATGGTCCAGGCAAACATCACCGCT**TAA**

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

36

36



- Nên khi biểu diễn ra protein là:

```
>>> gene.translate()
Seq('VKKMQSIVLALSLVLVAPMAAQAAEITLVPSVKLQIGDRDNR
GYYWDGGHWRDH...HR*', HasStopCodon(IUPACProtein(), '*'))
>>> gene.translate( to_stop=True )
Seq('VKKMQSIVLALSLVLVAPMAAQAAEITLVPSVKLQIGDRDNR
GYYWDGGHWRDH...HHR', IUPACProtein())
>>>
```

37



- Hoặc sử dụng hàm print() để xuất toàn bộ

```
>>> print(gene.translate( to_stop=True ))
VKKMQSIVLALSLVLVAPMAAQAAEITLVPSVKLQIGDRDNRGYY
WDGGHWRDHGWWKQHYEWRGNRWHLHGPPPPPRHHKKAPH
DHHGGHGP GKHHR
```

38



- Trong mã di truyền của vi khuẩn, GTG là một codon khởi đầu hợp lệ (bình thường đó là Valine),
- Khi đó, codon GTG muốn được dịch là Methionine, thì phải chỉ ra trình tự này là một CDS như sau

39



```
>>> print( gene.translate( to_stop=True ) )
VKKMQSIVLALSLVLVAPMAAQAAEITLVPSVKLQIGDRDNRGYY
WDGGHWRDHGWWKQHYEWRGNRWHLHGPPPPPRHHKKAPH
DHHGGHGP GKHHR
>>> print( gene.translate( table="Bacterial", cds=True ) )
MKKMQSIVLALSLVLVAPMAAQAAEITLVPSVKLQIGDRDNRGYY
WDGGHWRDHGWWKQHYEWRGNRWHLHGPPPPPRHHKKAPH
DHHGGHGP GKHHR
>>>
```

40

## Bảng mã Amino Acid



- Xem 2 bảng mã  
(<ftp://ftp.ncbi.nlm.nih.gov/entrez/misc/data/gc.prt>)
  - Standard translation
  - Vertebrate Mitochondrial (ty lập thể của động vật có xương sống)

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

41

41



```
>>> from Bio.Data import CodonTable
>>> standard_table = CodonTable.unambiguous_dna_by_id[1]
>>> mito_table = CodonTable.unambiguous_dna_by_id[2]
>>> print( standard_table )
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

42

42



```
>>> standard_table.start_codons
['TTG', 'CTG', 'ATG']
>>> standard_table.stop_codons
['TAA', 'TAG', 'TGA']
>>>
```

Table 1 Standard, SGC0					
	T	C	A	G	
T	TTT F	TCT S	TAT Y	TGT C	T
T	TTC F	TCC S	TAC Y	TGC C	C
T	TTA L	TCA S	TAA Stop	TGA Stop	A
T	TTG L(s)	TCG S	TAG Stop	TGG W	G
C	CTT L	CCT P	CAT H	CGT R	T
C	CTC L	CCP P	CAC H	CGC R	C
C	CTA L	CCA P	CAA Q	CGA R	A
C	CTG L(s)	CCG P	CAG Q	CGG R	G
A	ATT I	ACT T	AAT N	AGT S	T
A	ATC I	ACC T	AAC N	AGC S	C
A	ATA I	ACA T	AAA K	AGA R	A
A	ATG M(s)	ACG T	AAG K	AGG R	G
G	GTT V	GCT A	GAT D	GGT G	T
G	GTC V	GCC A	GAC D	GGC G	C
G	GTA V	GCA A	GAA E	GGA G	A
G	GTG V	GCG A	GAG E	GGG G	G

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

43

43



```
>>> print( mito_table )
>>> mito_table.start_codons
['ATT', 'ATC', 'ATA', 'ATG', 'GTG']
>>> mito_table.stop_codons
['TAA', 'TAG', 'AGA', 'AGG']
>>>
```

Table 2 Vertebrate Mitochondrial, SGC1					
	T	C	A	G	
T	TTT F	TCT S	TAT Y	TGT C	T
T	TTC F	TCC S	TAC Y	TGC C	C
T	TTA L	TCA S	TAA Stop	TGA W	A
T	TTG L	TCG S	TAG Stop	TGG W	G
C	CTT L	CCT P	CAT H	CGT R	T
C	CTC L	CCP P	CAC H	CGC R	C
C	CTA L	CCA P	CAA Q	CGA R	A
C	CTG L	CCG P	CAG Q	CGG R	G
A	ATT I(s)	ACT T	AAT N	AGT S	T
A	ATC I(s)	ACC T	AAC N	AGC S	C
A	ATA M(s)	ACA T	AAA K	AGA Stop	A
A	ATG M(s)	ACG T	AAG K	AGG Stop	G
G	GTT V	GCT A	GAT D	GGT G	T
G	GTC V	GCC A	GAC D	GGC G	C
G	GTA V	GCA A	GAA E	GGA G	A
G	GTG V(s)	GCG A	GAG E	GGG G	G

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

44

44

## Trình tự đột biến (Mutable Sequence)



- Trong Python không cho phép gán một phần tử trong chuỗi ký tự bằng một giá trị mới.
- Chẳng hạn,

```
>>> my_seq =
Seq("GCCATTGTAATGGGCCGCTGAAAGGGTGCCCGA",
IUPAC.unambiguous_dna)
>>> my_seq[3] = "T"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'Seq' object does not support item assignment
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

45

45

- Để có thể làm được việc này, cần dùng hàm **tomutable()** để chuyển trình tự thành trình tự đột biến:

```
>>> mutable_seq = my_seq.tomutable()
>>> mutable_seq[3] = "T"
```

- Xuất để kiểm tra
- ```
>>> my_seq
Seq('GCCATTGTAATGGGCCGCTGAAAGGGTGCCCGA',
IUPACUnambiguousDNA())
```
- ```
>>> mutable_seq
MutableSeq('GCCTTTGTAATGGGCCGCTGAAAGGGTGCCCG
A', IUPACUnambiguousDNA())
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

46

46

## Bắt cặp trình tự DNA



- Dùng thư viện Align
- Ví dụ bắt cặp toàn cục 2 trình tự:

```
import Bio
from Bio import Align

aligner = Align.PairwiseAligner()
aligner.mode = 'global'
aligner.match = 2
aligner.mismatch = -1
aligner.gap_score = -2

alignments = aligner.align("CATGT", "ACGCTG")
for a in alignments:
    print("Score = %.1f" % a.score)
    print(a)
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

47

47

## Bắt cặp trình tự Protein



- Ngoài thư viện Align, cần dùng thêm thư viện MatrixInfo trong SubsMat

```
>>> from Bio import Align
>>> from Bio.SubsMat import
MatrixInfo
```

- Dùng ma trận BLOSUM62 như bên cạnh

```
import Bio
from Bio import Align
from Bio.SubsMat import MatrixInfo
aligner = Align.PairwiseAligner()

aligner.substitution_matrix = MatrixInfo.blosum62
alignments = aligner.align("ACGCDG", "CADGD")

print("Score = %.1f" % alignments[0].score)
for a in alignments:
    print(a)
```

A.Prof. Tran Van Lang, PhD, VIETNAM ACADEMY OF SCIENCE AND TECHNOLOGY

48

48