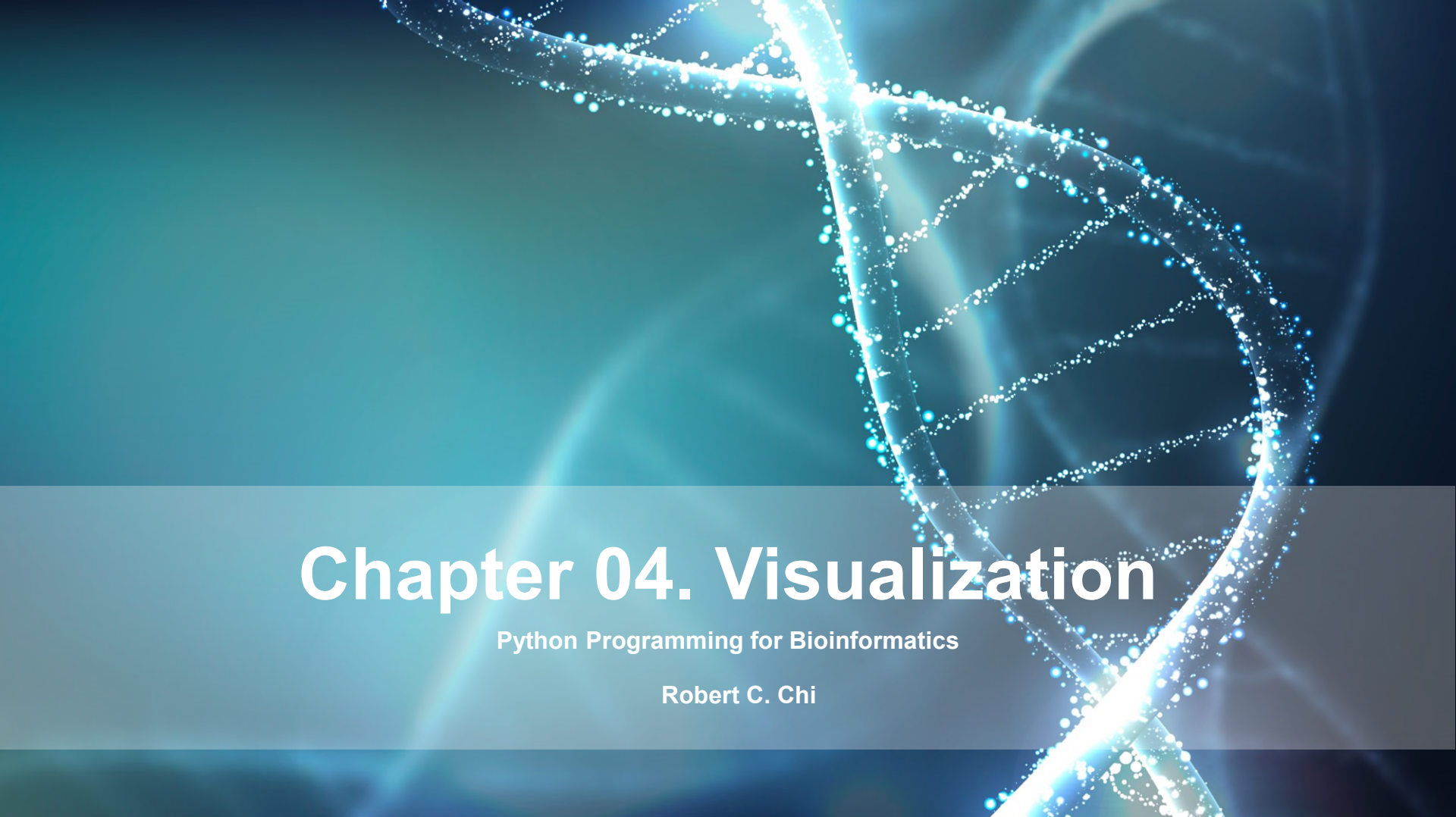


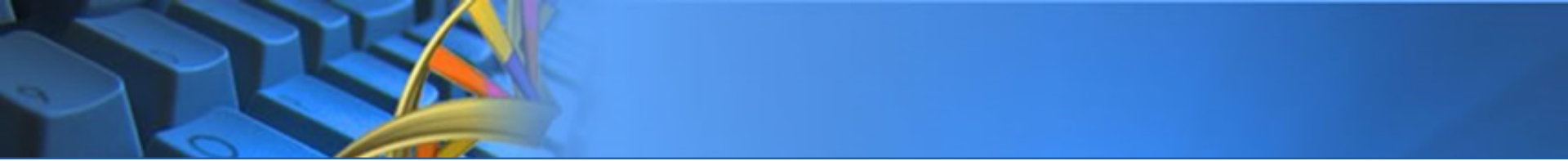
Chapter 04. Visualization

Python Programming for Bioinformatics

Robert C. Chi

Agenda

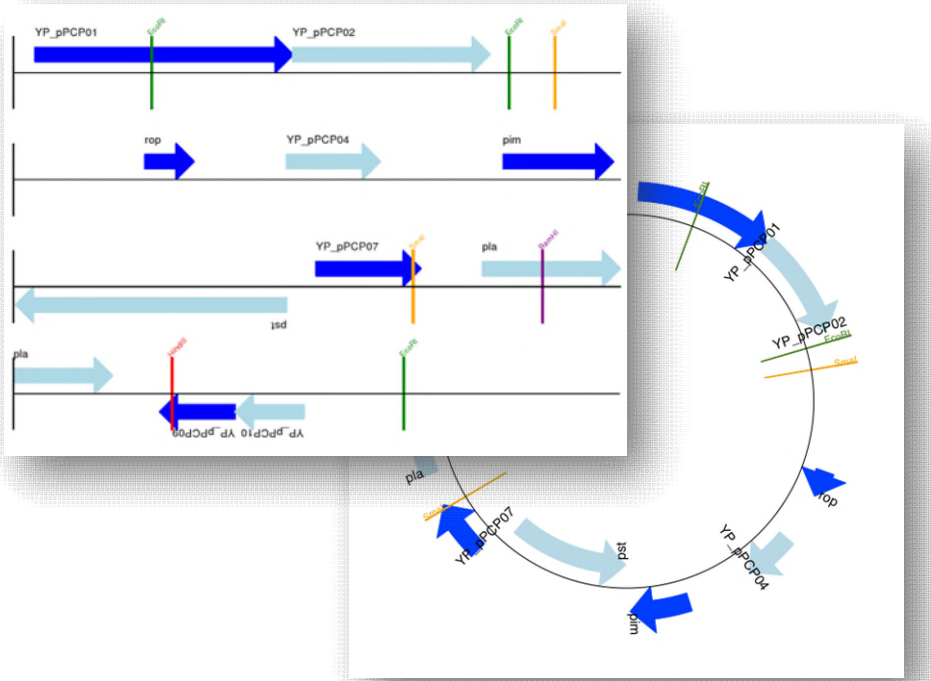
- Introduction
- Visualize Genome
- Visualize Chromosome
- Summary



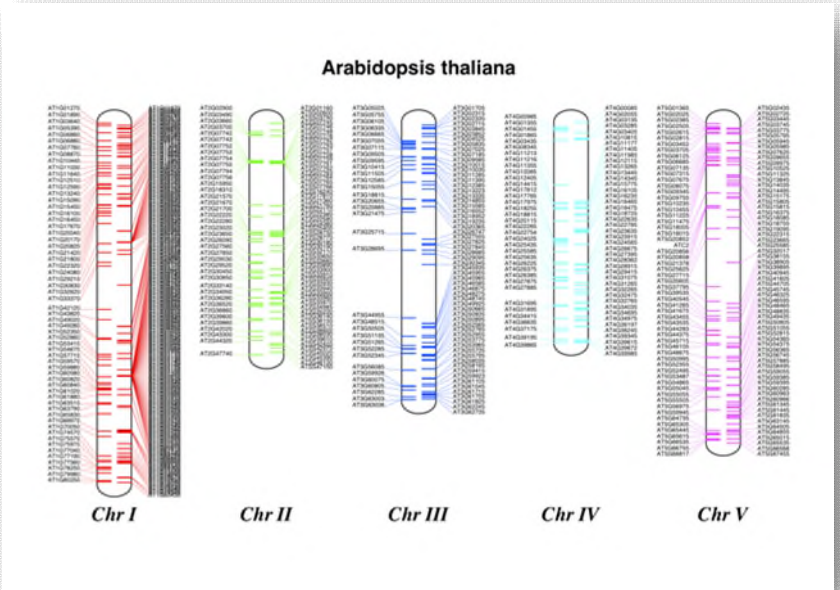
INTRODUCTION

What can it do?

- Visualize **Genome**



- Visualize **Chromosome**

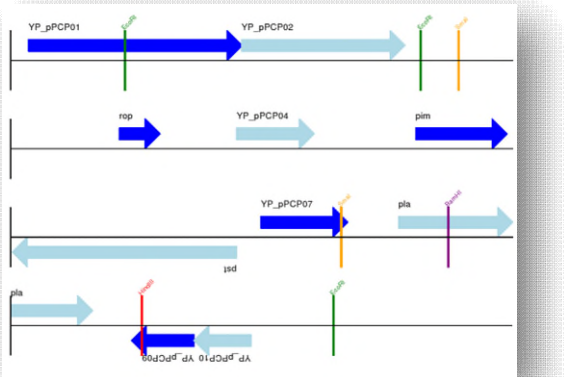


What Packages are Required?

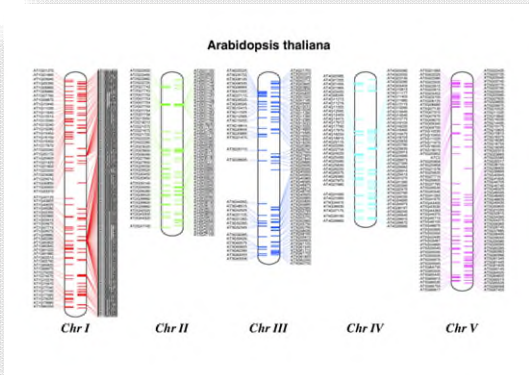
ReportLab

pip install **reportlab**
pip install **biopython**

Bio.Graphics.GenomeDiagram

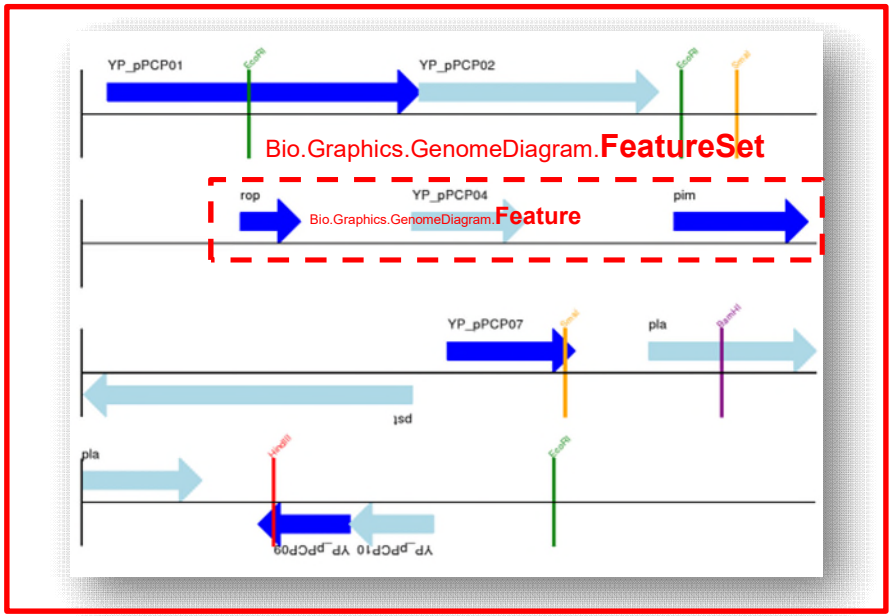


Bio.Graphics.BasicChromosome



Data Structure of GenomeDiagram

Bio.Graphics.GenomeDiagram.**Diagram**



Bio.Graphics.GenomeDiagram.**Track**

Bio.Graphics.GenomeDiagram.**Track**

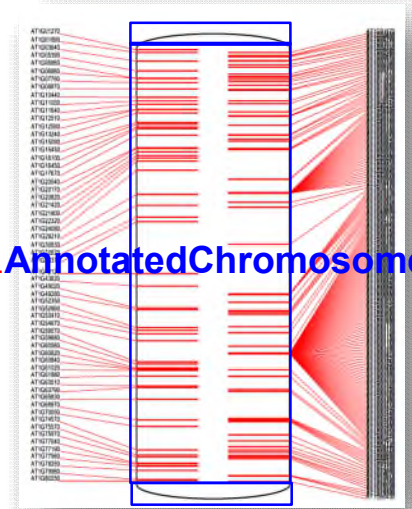
Bio.Graphics.GenomeDiagram.**Track**

Bio.Graphics.GenomeDiagram.**Track**

Data Structure of BasicChromosome

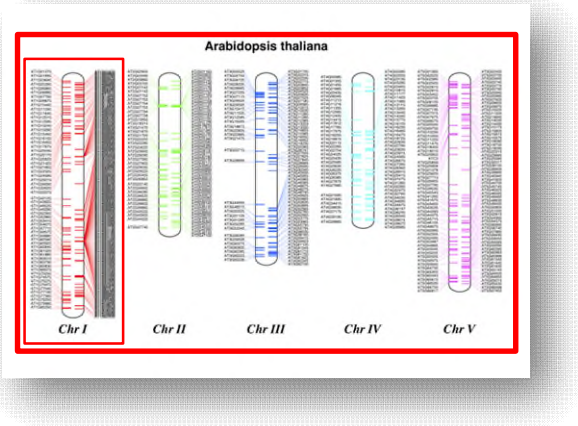
Bio.Graphics.BasicChromosome.**TelomereSegment**

Bio.Graphics.BasicChromosome.**AnnotatedChromosomeSegment**

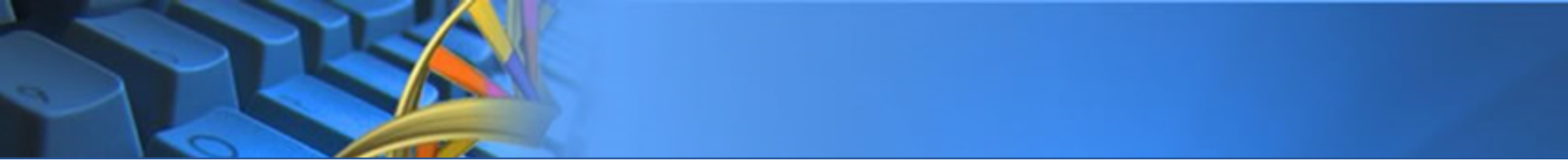


Bio.Graphics.BasicChromosome.**TelomereSegment**

Bio.Graphics.BasicChromosome.**Organism**



Bio.Graphics.BasicChromosome.**Chromosome**



VISUALIZE GENOME

Environment Preparation

- **Install Packages**

```
!pip install reportlab
!pip install biopython
```

- **Download the File**

```
# GenBank file for the pPCP1 plasmid from Yersinia pestis biovar Microtus
import os
if not os.path.isfile("NC_005816.gb"):
  os.system("wget https://raw.githubusercontent.com/biopython/biopython/master/Tests/GenBank/NC_005816.gb")
```

Practice

- **Visualize Genome: Environment Preparation**
  - Write and Run the following codes on a Colab page called "VisualizeGenome.ipynb":

Install Biopython

```
[9]  1  !pip install reportlab
     2  !pip install biopython
```

Download the Related File

```
[6]  1  # GenBank file for the pPCP1 plasmid from Yersinia pestis biovar Microtus
     2  import os
     3  if not os.path.isfile("NC_005816.gb"):
     4    os.system("wget https://raw.githubusercontent.com/biopython/biopython/master/Tests/GenBank/NC_005816.gb")
```

(Solution URL of this Practice)

Get Data from the File

```python
# Create a list to store all parsed SeqRecords
seq_records = []

# Parse one Record a time and add into seq_records
from Bio import SeqIO
for rec in SeqIO.parse("NC_005816.gb", "genbank"):
  seq_records.append(rec)

# Show how many records have been parsed
print("Total Number of Records:", len(seq_records))
print()

# Show the first record to prove the parsing was successful
print("--- First Record ---")
print(seq_records[0])
```

```
Total Number of Records: 1

--- First Record ---
ID: NC_005816.1
Name: NC_005816
Description: Yersinia pestis biovar Microtus str. 91001 plasmid pPCP1, complete sequence
Database cross-references: Project:58037
Number of features: 41
/molecule_type=DNA
/topology=circular
/data_file_division=BCT
/date=21-JUL-2008
/accessions=['NC_005816']
/sequence_version=1
/gi=45478711
/keywords=['']
/source=Yersinia pestis biovar Microtus str. 91001
/organism=Yersinia pestis biovar Microtus str. 91001
/taxonomy=['Bacteria', 'Proteobacteria', 'Gammaproteobacteria', ... 'Yersinia']
/references=[Reference(title='Genetics of metabolic variations ...'),
             Reference(title='Complete genome sequence ...'),
             Reference(title='Direct Submission', ...),
             Reference(title='Direct Submission', ...)]
/comment=PROVISIONAL REFSEQ: This record has not yet been subject to final
NCBI review. The reference sequence was derived from AE017046.
COMPLETENESS: full length.
Seq('TGTAACGAACGGTGCAATAGTGATCCACACCCAACGCCTGAAATCAGATCCAGG...CTG')
```

Practice

- **Visualize Genome: Get Data**
  - Write and Run the following codes on a Colab page called "VisualizeGenome.ipynb":

```
# Create a list to store all parsed SeqRecords
seq_records = []

# Parse one Record a time and add into seq_records
from Bio import SeqIO
for rec in SeqIO.parse("NC_005816.gb", "genbank"):
  seq_records.append(rec)

# Show how many records have been parsed
print("Total Number of Records:", len(seq_records))
print()

# Show the first record to prove the parsing was successful
print("--- First Record ---")
print(seq_records[0])
```

(Solution URL of this Practice)

Prepare Data Structures

```
1   # Get the first record as example
2   rec = seq_records[0]
3
4   # Create a Diagram
5   from Bio.Graphics import GenomeDiagram
6   diag = GenomeDiagram.Diagram(name=rec.id)
7
8   # Create a Track
9   trac = diag.new_track(1, name="Annotated Features")
10
11  # Create a Feature Set
12  feat_set = trac.new_set()
```

(1) Get the first record

(2) Create the Diagram

- name= String. Identifier for the diagram.

(3) Create the First Track

- name= String describing the track.

(4) Create the Feature Set

Practice

- **Visualize Genome: Create Data Structures**
  - Write and Run the following codes on a Colab page called "VisualizeGenome.ipynb":

```
# Get the first record as example
rec = seq_records[0]

# Create a Diagram
from Bio.Graphics import GenomeDiagram
diag = GenomeDiagram.Diagram(name=rec.id)

# Create a Track
trac = diag.new_track(1, name="Annotated Features")

# Create a Feature Set
feat_set = trac.new_set()
```

(Solution URL of this Practice)

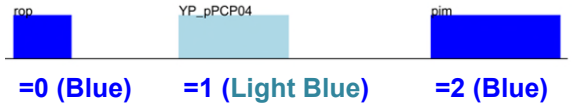
Create each Features

```python
# Find the Features == "gene" in the first record
from reportlab.lib import colors

for feat in rec.features:
  # Skip all the types not equal to "gene"
  if feat.type != "gene":
    continue

  # If the current index of Feature Set is even, set the color to Blue
  # If the current index of Feature Set is odd, set the color to Light Blue
  if len(feat_set) % 2 == 0:
    color = colors.blue
  else:
    color = colors.lightblue

  # Add this feature to the Feature Set
  feat_set.add_feature(feat, color=color,
                       label=True, label_size=14, label_angle=0)
```

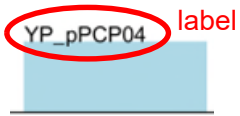
(1) Only fetch those features as “gene”

```
FEATURES    Location/Qualifiers
   gene     87..1109
            /locus_tag="YP_pPCP01"
            /db_xref="GeneID:2767718"
```

(2) Even as “Blue”, Odd as “Light Blue”



(3) Add Feature into Feature Set



Practice

- **Visualize Genome: Create each Feature**
  - Write and Run the following codes on a Colab page called “VisualizeGenome.ipynb”:

```
# Find the Features == "gene" in the first record
from reportlab.lib import colors

for feat in rec.features:
  # Skip all the types not equal to "gene"
  if feat.type != "gene":
    continue

  # If the current index of Feature Set is even, set the color to Blue
  # If the current index of Feature Set is odd, set the color to Light Blue
  if len(feat_set) % 2 == 0:
    color = colors.blue
  else:
    color = colors.lightblue

  # Add this feature to the Feature Set
  feat_set.add_feature(feat, color=color,
                       label=True, label_size=14, label_angle=0)
```

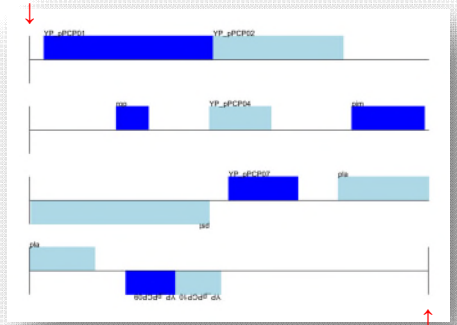
(Solution URL of this Practice)

Output the Result

```
1  # Draw the Diagram with Linear Format as PNG File
2  diag.draw(format="linear", orientation="landscape", pagesize='A4',
3                  fragments=4, start=0, end=len(rec))
4  diag.write("plasmid_linear.png", "PNG")
```

start=0

fragments=4

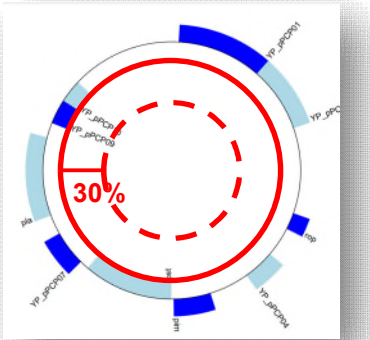


end=len(rec)

If the genome is circular in reality

pixel x pixel

```
6  # Draw the Diagram with Circular Format as PDF File
7  from reportlab.lib.units import cm
8  diag.draw(format="circular", circular=True, pagesize=(20*cm,20*cm),
9                  start=0, end=len(rec), circle_core=0.7)
10 diag.write("plasmid_circular.pdf", "PDF")
```

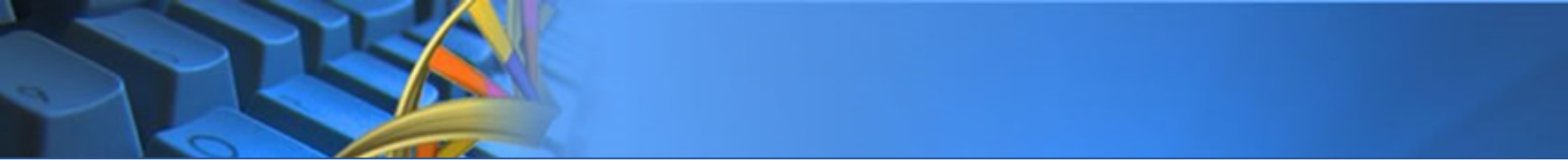


Practice

- **Visualize Genome: Output the Result**
  - Write and Run the following codes on a Colab page called "VisualizeGenome.ipynb":

```
# Draw the Diagram with Linear Format as PNG File
diag.draw(format="linear", orientation="landscape", pagesize='A4',
                fragments=4, start=0, end=len(rec))
diag.write("plasmid_linear.png", "PNG")

# Draw the Diagram with Circular Format as PDF File
from reportlab.lib.units import cm
diag.draw(format="circular", circular=True, pagesize=(20*cm,20*cm),
                start=0, end=len(rec), circle_core=0.7)
diag.write("plasmid_circular.pdf", "PDF")
```

(Solution URL of this Practice)



VISUALIZE CHROMOSOME

Environment Settings

Install Biopython

```
!pip install reportlab
!pip install biopython
```

Download the Related File

```
[3]
# bio is a tool to download GenBank file
!pip install bio

# Since the files are big, we only download the first chromosome
!bio fetch NC_003070 > NC_003070.gbk
#bio fetch NC_003071 > NC_003071.gbk
#bio fetch NC_003074 > NC_003074.gbk
#bio fetch NC_003075 > NC_003075.gbk
#bio fetch NC_003076 > NC_003076.gbk
```

Practice

- **Visualize Chromosome: Environment Settings**
  - Write and Run the following codes on a Colab page called "VisualizeChromosome.ipynb":

Install Biopython

```
!pip install reportlab
!pip install biopython
```

Download the Related File

```
# bio is a tool to download GenBank file
!pip install bio

# Since the files are big, we only download the first chromosome
!bio fetch NC_003070 > NC_003070.gbk
#bio fetch NC_003071 > NC_003071.gbk
#bio fetch NC_003074 > NC_003074.gbk
#bio fetch NC_003075 > NC_003075.gbk
#bio fetch NC_003076 > NC_003076.gbk
```

(Solution URL of this Practice)

Settings of Parameters

```
from reportlab.lib.units import cm
from Bio import SeqIO
from Bio.Graphics import BasicChromosome

# The chromosome name vs. its file
entries = [("Chr I", "NC_003070.gbk"),
#                    ("Chr II", "NC_003071.gbk"),
#                    ("Chr III", "NC_003074.gbk"),
#                    ("Chr IV", "NC_003075.gbk"),
#                    ("Chr V", "NC_003076.gbk")
           ]
# Set the scale factor of each chromosome
# Otherwise each chromosome will have same size visually
max_len = 30432563 #Could compute this
telomere_length = 1000000 #For illustration

chr_diagram = BasicChromosome.Organism()
chr_diagram.page_size = (29.7*cm, 21*cm) #A4 landscape
```

Practice

- **Visualize Chromosome: Settings of Parameters**
  - Write and Run the following codes on a Colab page called "VisualizeChromosome.ipynb":

```
from reportlab.lib.units import cm
from Bio import SeqIO
from Bio.Graphics import BasicChromosome

# The chromosome name vs. its file
entries = [("Chr I", "NC_003070.gbk"),
#                    ("Chr II", "NC_003071.gbk"),
#                    ("Chr III", "NC_003074.gbk"),
#                    ("Chr IV", "NC_003075.gbk"),
#                    ("Chr V", "NC_003076.gbk")
           ]
# Set the scale factor of each chromosome
# Otherwise each chromosome will have same size visually
max_len = 30432563 #Could compute this
telomere_length = 1000000 #For illustration

chr_diagram = BasicChromosome.Organism()
chr_diagram.page_size = (29.7*cm, 21*cm) #A4 landscape
```

(Solution URL of this Practice)

Visualize Chromosomes

```
for index, (name, filename) in enumerate(entries):
    # Fetch all records & features of this chromosome
    record = SeqIO.read(filename,"genbank")
    length = len(record)
    features = [f for f in record.features if f.type=="tRNA"]

    # Set the color of each chromosome
    #Record an Artemis style integer color in the feature's qualifiers,
    #1 = Black, 2 = Red, 3 = Green, 4 = blue, 5 =cyan, 6 = purple
    for f in features: f.qualifiers["color"] = [index+2]

    # Create a Chromosome
    cur_chromosome = BasicChromosome.Chromosome(name)
    # Set the scale to the MAXIMUM length plus the two telomeres in bp,
    # Allows each chromosome to be of unequal length, reflecting their true length
    cur_chromosome.scale_num = max_len + 2 * telomere_length

    #Add an opening telomere
    start = BasicChromosome.TelomereSegment()
    start.scale = telomere_length
    cur_chromosome.add(start)

    #Add a body - again using bp as the scale length here.
    body = BasicChromosome.AnnotatedChromosomeSegment(length, features)
    body.scale = length
    cur_chromosome.add(body)

    #Add a closing telomere
    end = BasicChromosome.TelomereSegment(inverted=True)
    end.scale = telomere_length
    cur_chromosome.add(end)

    #This chromosome is done
    chr_diagram.add(cur_chromosome)

chr_diagram.draw("tRNA_chrom.pdf", "Arabidopsis thaliana")
```

Practice

```
for index, (name, filename) in enumerate(entries):
    # Fetch all records & features of this chromosome
    record = SeqIO.read(filename,"genbank")
    length = len(record)
    features = [f for f in record.features if f.type=="tRNA"]

    # Set the color of each chromosome
    #Record an Artemis style integer color in the feature's qualifiers,
    #1 = Black, 2 = Red, 3 = Green, 4 = blue, 5 =cyan, 6 = purple
    for f in features: f.qualifiers["color"] = [index+2]

    # Create a Chromosome
    cur_chromosome = BasicChromosome.Chromosome(name)
    # Set the scale to the MAXIMUM length plus the two telomeres in bp,
    # Allows each chromosome to be of unequal length, reflecting their true length
    cur_chromosome.scale_num = max_len + 2 * telomere_length

    #Add an opening telomere
    start = BasicChromosome.TelomereSegment()
    start.scale = telomere_length
    cur_chromosome.add(start)

    #Add a body - again using bp as the scale length here.
    body = BasicChromosome.AnnotatedChromosomeSegment(length, features)
    body.scale = length
    cur_chromosome.add(body)

    #Add a closing telomere
    end = BasicChromosome.TelomereSegment(inverted=True)
    end.scale = telomere_length
    cur_chromosome.add(end)

    #This chromosome is done
    chr_diagram.add(cur_chromosome)

chr_diagram.draw("tRNA_chrom.pdf", "Arabidopsis thaliana")
```

- **Visualize Chromosome**
  - Write and Run the following codes on a Colab page called "VisualizeChromosome.ipynb":

(Solution URL of this Practice)

Summary

- **Bio.Graphics.GenomeDiagram**
  - Bio.Graphics.GenomeDiagram.Diagram
  - Bio.Graphics.GenomeDiagram.Track
  - Bio.Graphics.GenomeDiagram.FeatureSet
  - Bio.Graphics.GenomeDiagram.Feature

- **Bio.Graphics.BasicChromosome**
  - Bio.Graphics.BasicChromosome.Organism
  - Bio.Graphics.BasicChromosome.Chromosome
  - Bio.Graphics.BasicChromosome.TelomereSegment
  - Bio.Graphics.BasicChromosome.AnnotatedChromosomeSegment