

数理生物学演習

第11回 (Pythonによる) ゲノムデータ解析

中谷優介 (M2)

nakatani.yusuke.0919@gmail.com

数理生物学研究室 (佐々木グループ)

本日の内容

ゲノムデータの見方・解析手法を紹介

1. ゲノムデータ (の一部) を触ってみよう

Google Colabで系統樹作成

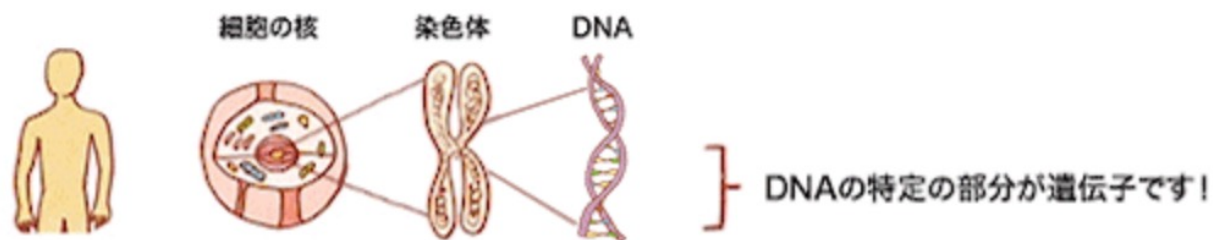
2. 研究紹介

オオセンチコガネの構造色のバリエーションを生み出す遺伝的基盤の探索

ゲノムとは？

- ゲノム: 各生物が持つ遺伝情報の全て
- 遺伝情報はDNAの4種類の塩基で構成
 - A, T, G, C
- ゲノムデータは4種類の文字列の並び

ゲノム



遺伝子(gene) + 染色体(chromosome) = ゲノム(genome)



ゲノムの何がすごいのか

- その生物の遺伝情報全てが載っている
 - 当然っちゃ当然やけど
 - 生物の形質を決める情報が詰まっている
- 進化の過程で変遷する
 - ゲノムを比べることで進化の過程を推定できる
- どの生物も同じ4文字を使っている
 - 違う生物種同士でも比較ができる

ゲノムデータは膨大

主な生物のゲノムの大きさと数

生物	ゲノム・サイズ(塩基の数)	遺伝子の数
大腸菌	460万	4100
トウモロコシ	23億	3万2000
イネ	3億8000万	3万2000
メダカ	8億	2万1000
カエル	17億	2万1000
ショウジョウバエ	1億8000万	1万4000
マウス	26億	2万6000
ヒト	30億	2万6000
サンショウウオ	400億	不明
肺魚	1300億	不明
デュビア(アメーバ)	6700億	不明

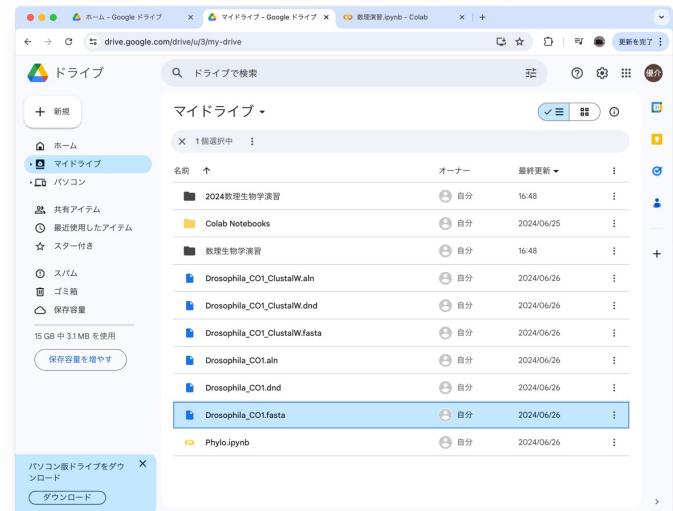
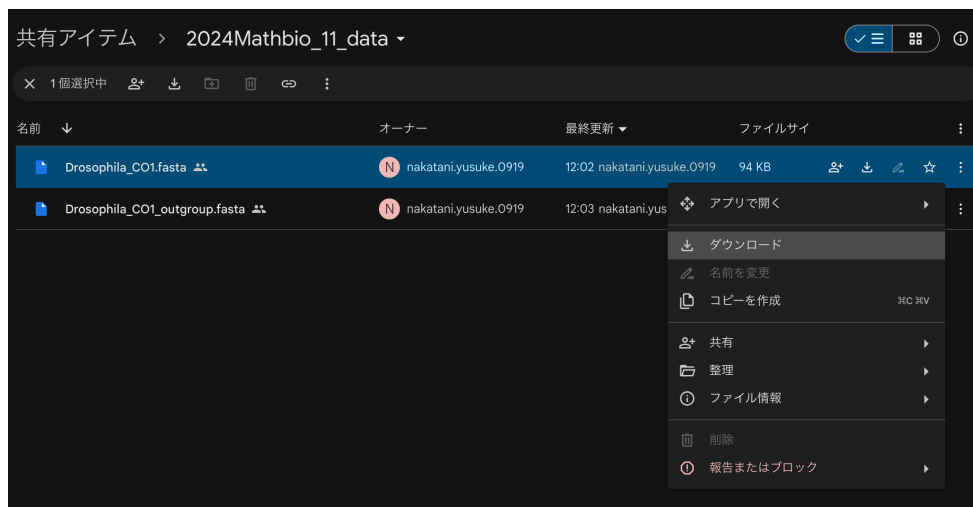
<https://www.nikkei.com/article/DGKDZO32302040V10C11A7M29000/>

全部は多すぎるので、ゲノムデータの一部だけ扱うこともある

ゲノムデータ (の一部) に触れてみよう

準備

- フォルダからDrosophila_CO1.fasta をダウンロード
<https://drive.google.com/drive/folders/1fuHI46OFE-dVIVaTIXSygzue3vMPRC4q?usp=sharing>
- Googleドライブにアップロード
 - 演習で使っているGoogleアカウントでログイン
 - マイドライブの直下に置いておくと後々やりやすい



今回扱うデータセット

- 日本産シヨウジョウバエの野外系統
- ミトコンドリアDNAのCOI(シトクロムオキシダーゼサブユニットI)領域の配列
- DNAバーコーディング (DNA配列からどの種かを調べる手法) でよく使われる領域
- FASTA形式のデータ

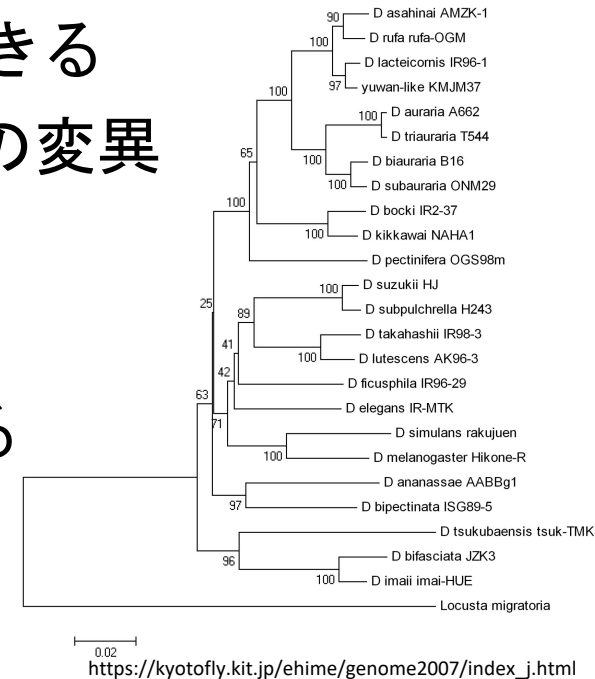
>配列ID
塩基配列

```
Drosophila_CO1.fasta
>D_hyde1_FK99
TCGGCACAAATGGTATTTTCAACAATCACAAGATATGGAACCTTATATTTATTTTCGGAGCTTGAGCGGGAATAGTGGGAACATCATTAAAGTATCTCAATCGAGCTGAACTGGGTATCTCTGGTCTTAAATGGAGACGACAAATTTATAAGTAACTGTCACCT
CACATGCTTTTGTAATAATTTTTATAGTATACCTTATAATCGGTGGGTTGGAAATGATTAGTTCCTTAAATATAGGAGCTCCGATATAGCTTCCACGAAATAAATAAAGATTTTGACTTCTCCCTGCTTATACCTTTATAGTAAGAAGTAT
AGTTGAAACGGAGCTGGACAGGGTGAACCTGTTACCCACTCTATCATCAGGAATGCCCAATGGAGGAGCTCTGATAGATTTAGCTAATTTTCTTCAATTTAGCCGGAATCTCTCAATTTAGGGGAGTAAATTTATACAACGTAAATTAATACGATCAACA
GGAATTAACCTCTGATCGTATACCTTTATTTGGTGGTCACTGTAATCTGCACTTTTATTTATTTATCTCTCCAGTTTACGGGGCTTACTAGTACTAACTGATCGAAATTAACAACATCATTTTTGACCTGCGAGGAGGAGTACCAATCCTTTACC
ACACTTATTTGATTTTTGGTCACTGAAATTTATTTAAATTTACCTGGGTTTGGAAATAATTCATATATTTAGTCAAGAATCAGGAAAGAAAGAACTTCGGATCTTAAAGGATAATTTACGCTATGCTGCAATTTGTTTACCTGGATTTGTTGAGC
TCATCATATTTCACTGAGGAATAGATGTTGATACTCGAGCTACTCACATCTGCAACTATAATTTGCTGTTCTACAGGATCAAAATTTTAAAGTATGCAACTTTACATGGGGCACAACTTACATACTCCCTGCTATTTATGGGCTTATAGGCTTGTATTT
TTATTCACAGTGGAGGTTAACAGGAGTGTATAGCTAATCTTCAGTGGATATTTTACATGATACATACAGCTAGTGTGCTAATTTTCACTATGTTTATCAATAGGAGCTGATTTGCAATTTAGGACAGGATTTATCAATGATATCTTTATTCAGGACTTA
CATTAATAATAAATGATAAAAAGTCAATTTATTTATTTATCGGAGTAAATCTTACTTTTTCCCAACATCTTTCTGGAATAGCGGGGATCCCTGCTGATCTGCACTACCCAGATGCTTACACAACATGAAATGAAATTTCAACAATTTGTTCTCATCTATT
ACTTTTAGGAATTTTATTTTTCTATATTTTGGAGAAGTCTGTATCACACGACGAAGTATTTACCTTATCAATTAATTTCAATTAAGTATGATACCAAAATCCCACTGCTGAACACAGATATTCGAGTTACCTTTTAACTAATTTAA
>D_hyde1_HK7K-2
TCGGCACAAATGGTATTTTCAACAATCACAAGATATGGAACCTTATATTTATTTTCGGAGCTTGAGCGGGAATAGTGGGAACATCATTAAAGTATCTCAATCGAGCTGAACTGGGTATCTCTGGTCTTAAATGGAGACGACAAATTTATAAGTAACTGTCACCT
CACATGCTTTTGTAATAATTTTTATAGTATACCTTATAATCGGTGGGTTGGAAATGATTAGTTCCTTAAATATAGGAGCTCCGATATAGCTTCCACGAAATAAATAAAGATTTTGACTTCTCCCTGCTTATACCTTTTATAGTAAGAAGTAT
AGTTGAAACGGAGCTGGACAGGGTGAACCTGTTACCCACTCTATCATCAGGAATGCCCAATGGAGGAGCTCTGATAGATTTAGCTAATTTTCTTCAATTTAGCCGGAATCTCTCAATTTAGGGGAGTAAATTTATACAACGTAAATTAATACGATCAACA
GGAATTAACCTCTGATCGTATACCTTTATTTGGTGGTCACTGTAATCTGCACTTTTATTTATTTATCTCTCCAGTTTACGGGGCTTACTAGTACTAACTGATCGAAATTAACAACATCATTTTTGACCTGCGAGGAGGAGTACCAATCCTTTACC
ACACTTATTTGATTTTTGGTCACTGAAATTTATTTAAATTTACCTGGGTTTGGAAATAATTCATATATTTAGTCAAGAATCAGGAAAGAAAGAACTTCGGATCTTAAAGGATAATTTACGCTATGCTGCAATTTGTTTACCTGGATTTGTTGAGC
TCATCATATTTCACTGAGGAATAGATGTTGATACTCGAGCTACTCACATCTGCAACTATAATTTGCTGTTCTACAGGATCAAAATTTTAAAGTATGCAACTTTACATGGGGCACAACTTACATACTCCCTGCTATTTATGGGCTTATAGGCTTGTATTT
TTATTCACAGTGGAGGTTAACAGGAGTGTATAGCTAATCTTCAGTGGATATTTTACATGATACATACAGCTAGTGTGCTAATTTTCACTATGTTTATCAATAGGAGCTGATTTGCAATTTAGGACAGGATTTATCAATGATATCTTTATTCAGGACTTA
CATTAATAATAAATGATAAAAAGTCAATTTATTTATTTATCGGAGTAAATCTTACTTTTTCCCAACATCTTTCTGGAATAGCGGGGATCCCTGCTGATCTGCACTACCCAGATGCTTACACAACATGAAATGAAATTTCAACAATTTGTTCTCATCTATT
ACTTTTAGGAATTTTATTTTTCTATATTTTGGAGAAGTCTGTATCACACGACGAAGTATTTACCTTATCAATTAATTTCAATTAAGTATGATACCAAAATCCCACTGCTGAACACAGATATTCGAGTTACCTTTTAACTAATTTAA
>D_hyde1_MTY2
TCGGCACAAATGGTATTTTCAACAATCACAAGATATGGAACCTTATATTTATTTTCGGAGCTTGAGCGGGAATAGTGGGAACATCATTAAAGTATCTCAATCGAGCTGAACTGGGTATCTCTGGTCTTAAATGGAGACGACAAATTTATAAGTAACTGTCACCT
CACATGCTTTTGTAATAATTTTTATAGTATACCTTATAATCGGTGGGTTGGAAATGATTAGTTCCTTAAATATAGGAGCTCCGATATAGCTTCCACGAAATAAATAAAGATTTTGACTTCTCCCTGCTTATACCTTTTATAGTAAGAAGTAT
AGTTGAAACGGAGCTGGACAGGGTGAACCTGTTACCCACTCTATCATCAGGAATGCCCAATGGAGGAGCTCTGATAGATTTAGCTAATTTTCTTCAATTTAGCCGGAATCTCTCAATTTAGGGGAGTAAATTTATACAACGTAAATTAATACGATCAACA
GGAATTAACCTCTGATCGTATACCTTTATTTGGTGGTCACTGTAATCTGCACTTTTATTTATTTATCTCTCCAGTTTACGGGGCTTACTAGTACTAACTGATCGAAATTAACAACATCATTTTTGACCTGCGAGGAGGAGTACCAATCCTTTACC
ACACTTATTTGATTTTTGGTCACTGAAATTTATTTAAATTTACCTGGGTTTGGAAATAATTCATATATTTAGTCAAGAATCAGGAAAGAAAGAACTTCGGATCTTAAAGGATAATTTACGCTATGCTGCAATTTGTTTACCTGGATTTGTTGAGC
TCATCATATTTCACTGAGGAATAGATGTTGATACTCGAGCTACTCACATCTGCAACTATAATTTGCTGTTCTACAGGATCAAAATTTTAAAGTATGCAACTTTACATGGGGCACAACTTACATACTCCCTGCTATTTATGGGCTTATAGGCTTGTATTT
TTATTCACAGTGGAGGTTAACAGGAGTGTATAGCTAATCTTCAGTGGATATTTTACATGATACATACAGCTAGTGTGCTAATTTTCACTATGTTTATCAATAGGAGCTGATTTGCAATTTAGGACAGGATTTATCAATGATATCTTTATTCAGGACTTA
CATTAATAATAAATGATAAAAAGTCAATTTATTTATTTATCGGAGTAAATCTTACTTTTTCCCAACATCTTTCTGGAATAGCGGGGATCCCTGCTGATCTGCACTACCCAGATGCTTACACAACATGAAATGAAATTTCAACAATTTGTTCTCATCTATT
ACTTTTAGGAATTTTATTTTTCTATATTTTGGAGAAGTCTGTATCACACGACGAAGTATTTACCTTATCAATTAATTTCAATTAAGTATGATACCAAAATCCCACTGCTGAACACAGATATTCGAGTTACCTTTTAACTAATTTAA
>D_immigrans_MT19-1
TCGGCACAAATGGTATTTTCAACAATCACAAGATATGGAACCTTATATTTATTTTCGGAGCTTGAGCGGGAATAGTGGGAACATCATTAAAGTATCTCAATCGAGCTGAACTGGGTATCTCTGGTCTTAAATGGAGACGACAAATTTATAAGTAACTGTCACCT
CTGATGCTTATTAATAATTTTTATAGTATGCAATTAATAATGGAGGTTTGGAAATGATTAGTTCCTTAAATATAGGAGCTCCGATATAGCTTCCACGAAATAAATAAAGATTTTGACTTCTCCCTGCTTATACCTTTTATAGTAAGAAGTAT
GGTGAATAATGGAGGAACTGGGTGAACAGTGTACCTCTTATCATCAGGGAATGCATGGTGGGCTCAGTAGATTTTGTCAATTTTTCTTCAATTTAGCCGGAATTTCTCAATTTAGGGGCTGTAATTTTATACAACAGATTTAATTAACGATCTCTCT
GGAATTAACCTTAGCAGGAATCTTATTTGTTGATCAGTTGTAATCTGCAATTTATTTATTTATCTCTACCTGTTTACGGGAGCAATTAACAATTTATGACAGACGAAATTAACAACATCATTTTTGACCTGCGAGGAGGAGGAGCCTATTTTAT
AACATTTCTGATTTTTGGGCATCCAGAAGTTTATTTTTAAATCTCTCGGATTCGGAATAATTTTCAATATATTTAGTCAAGAATCAGGTAAGAAAGAAACATTTGGGCTTAAAGGATAATTTATGCAATATGCTATTTGGGTTATTTGTTGAGC
TCACCATATTTTACAGTGGGATAGATGTTGATACAGAGCTTATTTACTCGCAACTATAATTTGCTGTTACCAACAGGAATTAATTTTCAAGTGTATGCAACTTTACATGGAGCTCAACTTACATACTCACCAGCAATTTATGAGCTTATAGGATTTGATTT
TTATTTACTGAGGAGGATTAAGTGGATTTGTTAGCTAATCTGCTAGATTTATTTCTCATGATATCTTACTAGTATGAGCTGCTCCTATGATGATTTATCAATAGGAGCTGATTTGCTATTTATAGCAGGATTTATGCTATTTATAGCAGGATTTATGATCTTAT
CATTAATAATAAATGATAAAAAGTCAATTTATTTATTTATTTGGGGTAAATTTACCTTTTTCCCAACATCTTTTGGATAGCTGCGGAATACCCGAGTATTTCTGATTAACCTGAGCTTACACTCTTGAATGATTTCAACAATTTGTTCTCATCTATT
CTTTTAGGAATTTTATTTTTCTTATATTTTGGAGAAGTGTAGTGTCTCAACGACAGTAAATTTCAAAATCAATTAATTTCTGATTAAGTATGATACCAAAATCCCACTGCTGAACACAGATATTCGAACTCTTTTAACTAATTTAA
```


分子系統樹の作成

- DNAに蓄積する変異は一定の割合で起きる
- ほとんどは自然選択とは無関係な中立の変異

- DNA配列がどれくらい似ているか
→進化的にどの程度近縁かを推定できる



- 全ゲノムを使うこともあるが、ここでは一部の遺伝子
（マーカー遺伝子）配列のみを扱う
→データが大きすぎず扱いやすい

Google Colabを立ち上げる

```
# 11-01 必要なツールのインストール
```

```
!apt-get install clustalw
```

```
!pip install biopython
```

```
!pip install scipy
```

```
# 11-02 モジュールとパッケージの読み込み
```

```
from Bio.Align.Applications import ClustalwCommandline
```

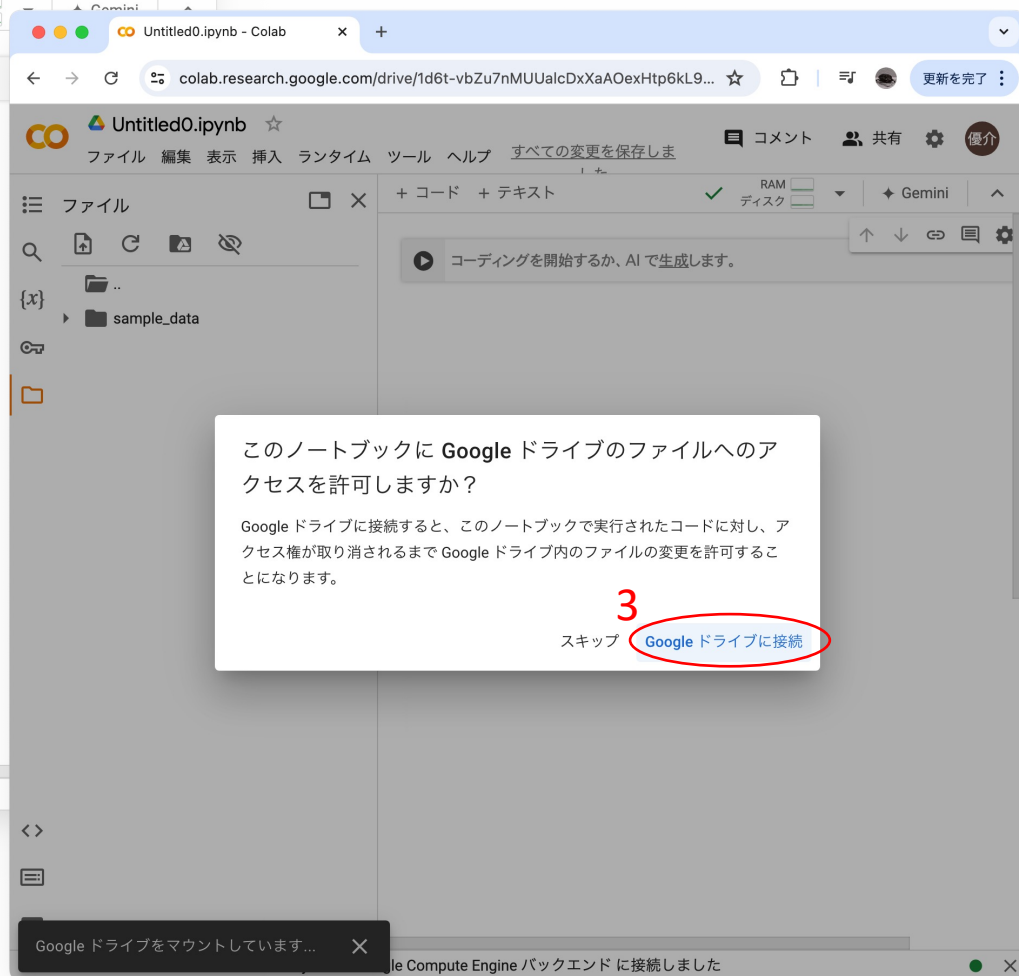
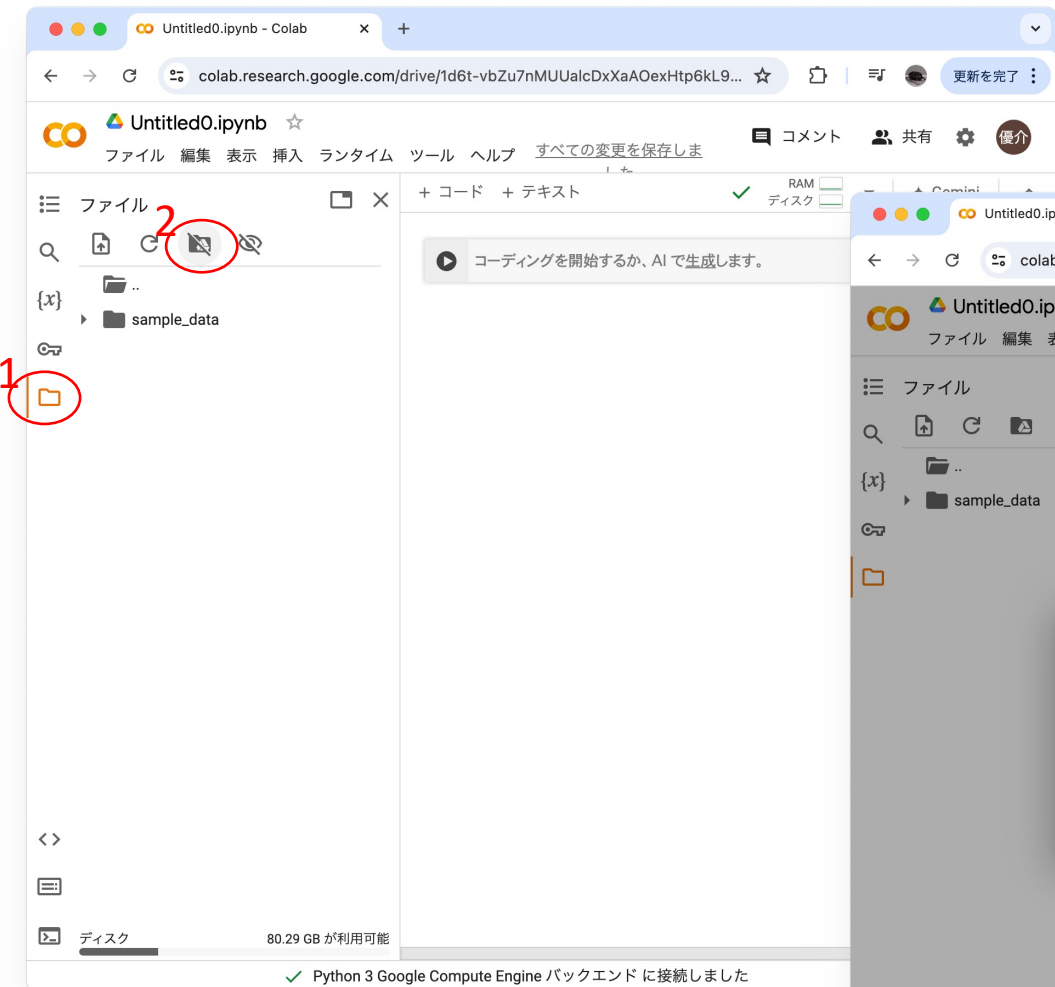
```
from Bio import AlignIO, Phylo
```

```
from Bio.Phylo.TreeConstruction import
```

```
DistanceCalculator, DistanceTreeConstructor
```

```
import matplotlib.pyplot as plt
```

Google ドライブのファイルへのアクセスを許可



パス（ファイルの住所）を通す

```
# 11-03 ファイルとツールのパスを通す
```

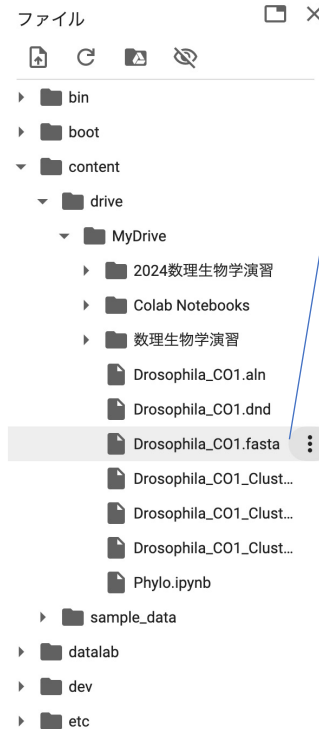
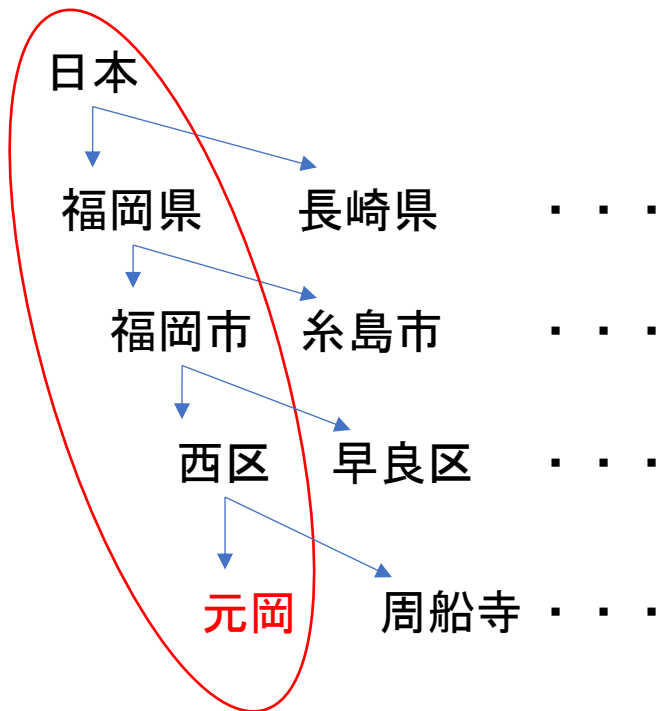
```
# FASTAファイル
```

```
fasta_file = '/content/drive/MyDrive/Drosophila_CO1.fasta'
```

```
# ClustalW
```

```
clustalw_exe = "/usr/bin/clustalw"
```

```
clustalw_cline = ClustalwCommandline(clustalw_exe, infile=fasta_file)
```



パスをコピーする

Drosophila_CO1.fastaのパスはコピペすると良い

The image displays two sequential screenshots of a Google Colab notebook interface, illustrating the correct way to copy a file path for a FASTA file.

Left Screenshot: The file explorer on the left shows a directory structure under 'MyDrive' containing a folder '2024数理生物学演習' and a file 'Drosophila_CO1.fasta'. A red circle highlights the three-dot menu icon next to the file name.

Right Screenshot: The same file explorer is shown, but the context menu is open over 'Drosophila_CO1.fasta'. A red circle highlights the option 'パスをコピー' (Copy path), which is the recommended action.

The code editor in the background of both screenshots contains the following Python code:

```
from Bio.Align.Applications import ClustalwCommandl
from Bio import SeqIO, AlignIO, Phylo
from Bio.Phylo.TreeConstruction import DistanceCalculator, DistanceTreeConstructor
import matplotlib.pyplot as plt
```

Below the code, a warning message is displayed:

```
Due to the on going maintenance burden of keeping command line application wrappers up to date, we have decided to deprecate and eventually remove the modules.

We instead now recommend building your command line and invoking it directly with the subprocess module.

warnings.warn(
```

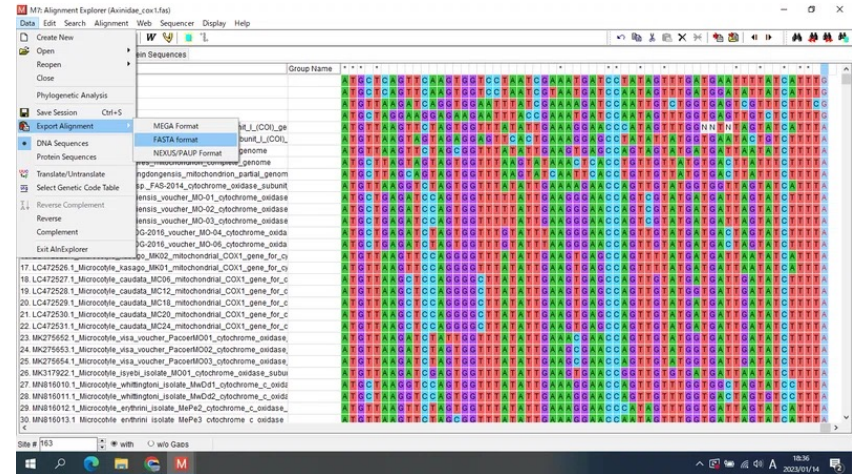
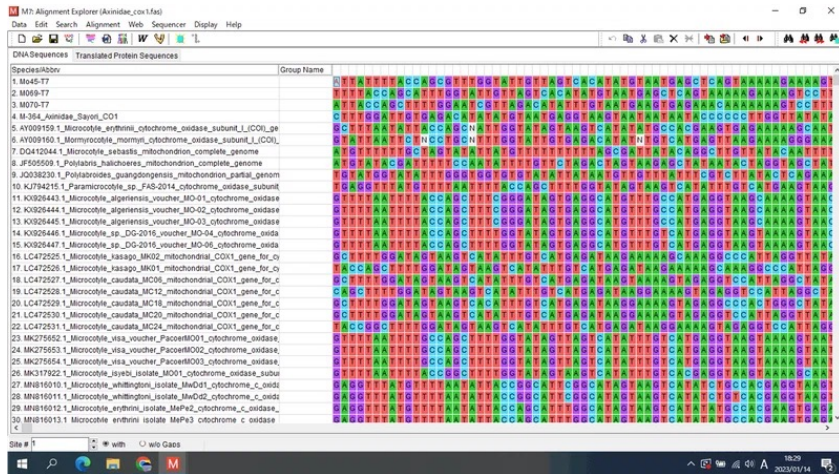
ClustalWでアライメント

※やや時間がかかります (2分くらい)

11-04 アライメントの実行, 結果の読み込み

```
stdout, stderr = clustalw_cline()
```

```
alignment = AlignIO.read(fasta_file.replace(".fasta", ".aln"), "clustal")
```



https://note.com/fish_and_worms/n/n6f64cc75447a

配列を整列させる

系統樹作成の準備

配列間の距離を計算する方法を指定

identity method (同一性法)

配列間の一致する位置の割合に基づいて距離を計算

```
# 11-05 距離行列の計算, 系統樹の作成
```

```
# 距離行列の計算
```

```
calculator = DistanceCalculator('identity')
```

```
dm = calculator.get_distance(alignment)
```

```
# 系統樹の作成
```

```
constructor = DistanceTreeConstructor()
```

```
tree = constructor.nj(dm) # neighbor-joining method
```

距離行列 (dm) をもとに近隣結合法を使用して系統樹 (tree) を構築

近隣結合法(NJ法)の原理

例: 4つの配列A, B, C, Dの間の距離行列

	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
<i>A</i>	0	5	9	9
<i>B</i>	5	0	10	10
<i>C</i>	9	10	0	8
<i>D</i>	9	10	8	0

- 1.最も近いペア (AとB) を見つけて結合する
- 2.新しいクレードAB以外の配列C, Dへの距離を計算する
- 3.距離行列を更新し、再び最も近いペアを見つけて結合する
- 4.これを繰り返して系統樹を構築する

系統樹を描画する

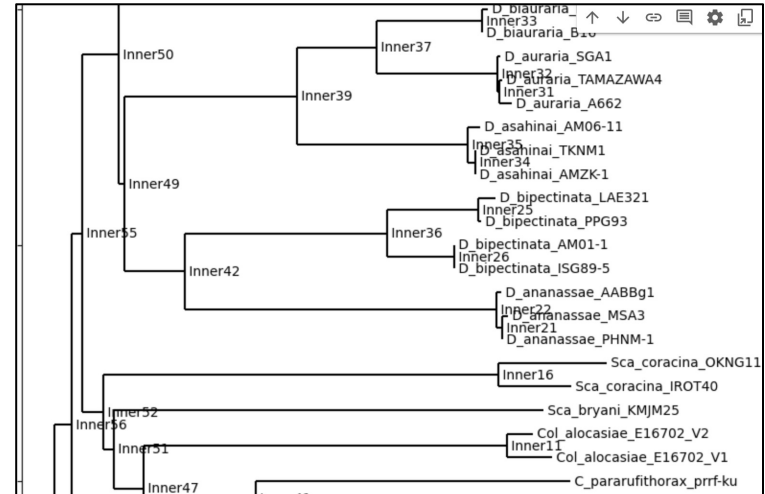
```
# 11-06 系統樹の描画
```

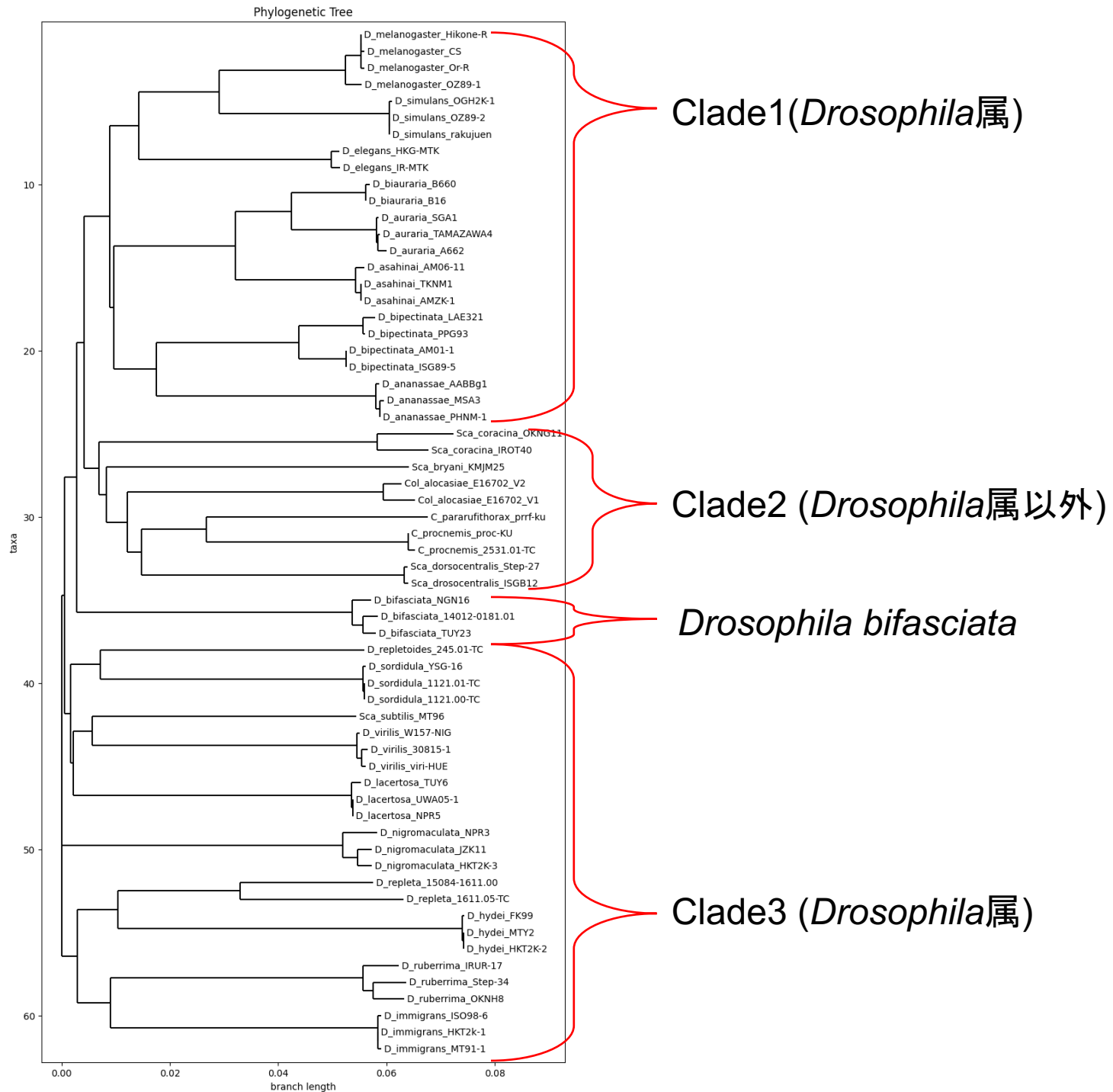
```
# 内部ノードの名前を削除
```

```
for clade in tree.find_clades():  
    if not clade.is_terminal():  
        clade.name = ""
```

```
# 系統樹の描画
```

```
fig = plt.figure(figsize=(10, 20))  
ax = fig.add_subplot(1, 1, 1)  
Phylo.draw(tree, axes=ax, do_show=False)  
plt.title('Phylogenetic Tree')  
plt.show()
```





課題

Normal

1. 今回作成した系統樹の中のClade1に入る種とClade2に入る種それぞれの共通点を教えてください。
(ヒント: どちらもDrosophila属だが、亜属は同じ?)
2. 1. で答えたの共通点をもとに考えると、*Drosophila bifasciata*はClade1,2のどちらに属するか?
3. 質問・感想・要望をどうぞ!

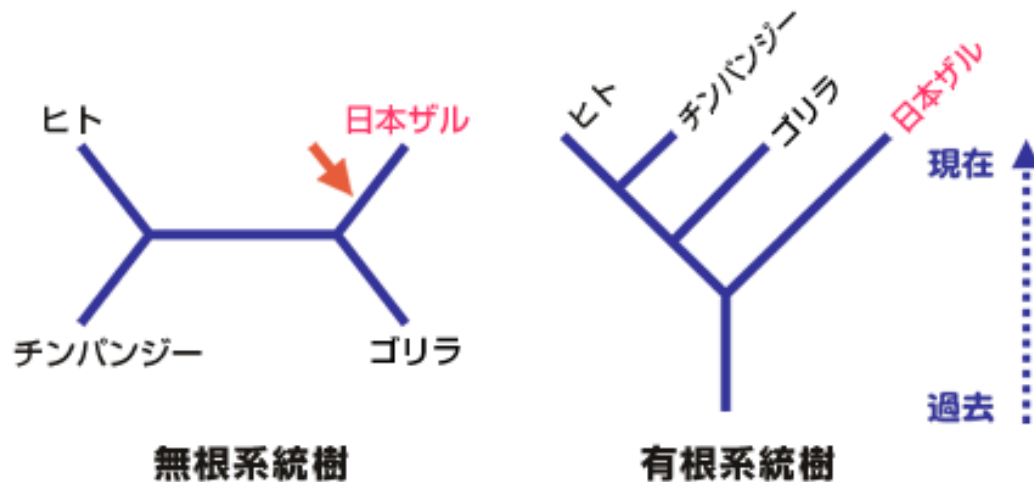
Hard

1. 以下の2点のどちらか（もしくは両方）を実施し、系統樹の図を添付してください。
 - 系統樹に外群を追加する
 - 近隣結合法以外で系統樹を作成する

応用編: 外群(outgroup)を追加

外群: 対象の配列よりも遠縁の特定の配列

今回のHard課題ではトノサマバッタの配列を使用
対象の配列に入りそうなものはNG (*Drosophila*属とか)
遠縁すぎるものもNG (シロイヌナズナとか)



https://www.brh.co.jp/research/formerlab/miyata/2005/post_000008.php

外群を追加すると有根系統樹が作成できる
共通祖先からどのように分岐したか推定可能

外群付きの系統樹作成

Drosophila_CO1_outgroup.fasta をダウンロード

```
Drosophila_CO1_outgroup.fasta - 編集済み
ACTTTTCATTACGTTTTATCAATAGGAGCTGATTTGCTATTATAGCAGGATTCATTGATGATCCCTTTACAGGATTAACAATAAATCAAATAATTAATAAAGTCAATTTATTTATTTGGAGTTAAT
TTAACATTTTTCCCTCAACATTTTTAGGATTAGCAGGTATACCTCGTCGATATTCAGATTACCAGATGCTTACTTTCATGAAATATTTTCAACAATTTGGATCATCTATTTCTTTAGGAAATTTTTTTTTCT
TTATATTTTGGAGAAAGTTAGTTTCCACAGCTCAAGTAATTTATCCAGTCAACTAAATTCATCTTATGAAATGATACAAAATACCACACCTGCTGAACATAGTTATTCGAAATACCAATTTTAACTAAATA
>Co1_alocas.iae_E16702_V1
TCGCGCAATGGTTTCTACGAATCATAAAGATTCGGAACCTTTATTTATTTTCGGGCGTAGCCGGAATAATGGAACTTCATTAAGAATTTAATTCGAGCAGAATAGGACCTCGGAGCTTTAATTTGG
AAATGAAACAATTTATAGCTAATTTGTAAGTAATTTTTATTTGTAATCAATATAATTTGGAAATTCGGAATTCGGAATTTGATTTAGTTCCTTAAATTTAGGAGCTCTGATATGGCATTCCAC
GTAATAAATAATAGATTTTGACTACTCCCTGCTCTTCTCTTTATAGTAAGAAGAATAGTTGAAATGGAGCTGGAACAGGATGAACAGTTTACCCTGCTTATCAGCAGGAATTTGCTCATGGAGGAGCTCT
GTTGATTTAGCTATCTTTCTCTTCTTATAGCAGGAATTTTCATCAATTTAGGGCGAGTAAATTTTACTACAGTAATAATACGATCAACCGGAATTTAGCAGGAATTTGCTGATCAGTAGT
AATACAGCTATTTTACTTTTATCTTTCATGATTTAGCAGGAGCAATTAACAATACTTAAACAGACGAAATTTAAATCTTCTTTGACCTGCGGAGGAGGAGGACCAATTTATATCAACATTTATTTT
GATTTTTGGTCATCCAGAAATATATATTTAAATTTTACCTGGATTTGGGATAATTTCTCATATTTAGGCAAGAATCAGGAAAAAGGAAACTTTGGTCTTTAGGAAATTTATGCTATCTTTGGTTTTA
TTAGGATTTATGATGAGCTCATATATTTACAGTAGGAATAGATGTTGATACAGAGCTTACTTTCATCGGCTACAATAATTTTGCAGTACCAACAGGAATTTAAATTTTATGTTGATAGCACTTTACAGT
AACCAATTAATATTTCCAGCTATTTTATGAGCTTTAGGATTCGATTTTTTTCACAGTAGGAGGATTTAACAGGAGTTGTTTTAGCAAAATTCATCAATTTGATATTTATACATGATACATATTTAGTAGCTC
ATTTTCATTTATGTTTATCAATGGGAGCTGATTTGCTATTATAGCAGGATTTGATACATGATACCTTTAATCAGGATTAACATAAATAAATGATTTAAAGTCAATTTATTTATTTTGGAGTTAAT
TTAACCTTTTCCCTCAACATTTTTAGGATTAGCTGGAATACCCGAGACTACCTGAGTTATCCAGATGCTTATACCTCATGAAATTTATTTCTCAATTTGGTTCATCTTTTATTTAGGAAATTTTTTTTTTT
TTATATTTTGGAGAAAGTTAGTTTCCACAGCAGAAATTTATCTTCAATTAATAATTTCTTATGAAATGATACAAAATCTCCACAGCTGAACATAGTTATTCGAAATACCAATTTTAAATAAATA
>Co1_alocas.iae_E16702_V2
TCGCGCAATGATTTCTCAAAATCATAAAGATTTGGAACCTTTTACTTTATTTTGGAGCTAGCCGGAATAGTCGGAACCTTCATTAAGAATTTAATTCGAGCAGAATAGGACCTCGGAGCTTTAATTTGG
AAATGAAACAATTTTAACTGATTTGTAAGCTAATTTTTGTAATAATTTTTTATGGTAATCAAAATATAATTTGGAGGATTCGGAATTTGATTTAGTTCCTTAAATTTAGGAGCTCTGATATGGCATTCCAC
GTAATAAATAATAGATTTTGACTACTCCCTGCTCTTCTCTTTATAGTAAGAAGAATAGTTGAAATGGAGCTGGAACAGGATGAACAGTTTACCCTGCTTATCAGCAGGAATTTGCTCATGGAGGAGCTCT
GTTGATTTAGCTATCTTTCTCTTCTTATAGCAGGAATTTTCATCAATTTAGGGCGAGTAAATTTTACTACAGTAATAATACGATCAACCGGAATTTAGCAGGAATTTGCTGATCAGTAGT
AATACAGCTATTTTACTTTTATCTTTCATGATTTAGCAGGAGCAATTAACAATACTTAAACAGACGAAATTTAAATCTTCTTTGACCTGCGGAGGAGGAGGACCAATTTATATCAACATTTATTTT
GATTTTTGGTCATCCAGAAATATATATTTAAATTTTACCTGGATTTGGGATAATTTCTCATATTTAGCAAGAATCAGGAAAAAGGAAACTTTGGTCTTTAGGAAATTTATGCTATCTTTGGTTTTA
TTAGGATTTATGATGAGCTCATATATTTACAGTAGGAATAGATGTTGATACAGAGCTTACTTTTACCTCGGCTACAATAATTTTGCAGTACCAACAGGAATTTAAATTTTATGTTGATAGCACTTTACAGT
AACCAATTAATATTTCCAGCTATTTTATGAGCTTTAGGATTCGATTTTTTTCACAGTAGGAGGATTTAACAGGAGTTGTTTTAGCAAAATTCATCAATTTGATATTTATACATGATACATATTTAGTAGCTC
ATTTTCATTTATGTTTTATCAATGGGAGCTGATTTGCTATTATAGCAGGATTTGATACATGATACCTTTATTTACAGGATTAACATAAATAAAGTCAATTTTATTTATTTTGGAGTTAAT
TTAACCTTTTCCCTCAACATTTTTAGGATTAGCTGGAATACCCGAGACTACCTGAGTTATCCAGATGCTTATACCTCATGAAATTTAATTTCTCAATTTGGTTCATCTTTTATTTAGGAAATTTTTTTTTT
TTATATTTTGGAGAAAGTTAGTTTCCACAGCAGAAATTTATCTTCAATTAATAATTTCTTATGAAATGATACAAAATCTCCACAGCTGAACATAGTTATTCGAAATACCAATTTTAAATAAATA
>Locusta_migratoria
TTATTTCTCAACAACACAGGACATTTGGTACACTGATTTTATATTTTCGGGCGTAGCTGGAATAGTAGGAACATCAATAAGAATAATTTTCGAGCTGAATTTAGCTCAACAGGAACAATAATTAACGATGATCAAGT
ATAATAGTAACTTACAGCCACAGCATTTGTTATAATTTCTCATGGTTATGCAAAATAAATTTGGAGGATTCGGAATTTGATTTAGTACCATTAATAATTTGGAGCTCAGATATAGCTTTCCAGGAATAAATAACA
TAAGATTTGATTTTACCACCATCATTAACACTCTCACTAATGCTCTCTGATAGATAGATAGGAGCTGTTACAGGATGAACAGTATATCCCCACTAGCCAGAGTCTTCTGATAGGAGGATCTCTGATAGATTTAGCA
ATTTTTTCTTACACTAGCAGGTTCTCCTCAATTTTAGGAGCTTTAATTTCTTACACAGCAATCAATATACGATCAAAATAATAAACCCTTGATCAACACCATTTATTTGATCAGTAGCAATTTACAGCT
ATTACTTTTATATCATTCAGCTATTTAGCTGGAGCAATTAATTTATTTAACCTGATCGAACTTTAATACGTCATTTCTGACCTCGCAGGAGGAGGTTGACCAATTTTATATCAACACTATTTTGTCTTTGGTC
ACCAGAGATTTATTTTTAAATTTACAGGATTTGGTATTTTTCACATTCGTTGTCAGAAAGAGGAAATTTGAATCATTTGGAACTTTGGTAAATTTATGCAATATTTCAATTTGGATTTATTTGATTTATTT
GATGAGCTCATCAATTTTACAGTAGGAATAGCAGTACAGCTTATTTTCCCTCAGCAACAATAATTTGCTGATCAACAGGATTTAGGATTTTCAATTTGATTTTATTTAGGAAATTTTATTTTATTTT
ATTCAACCCACTTATTTAGGATTTAGGATTTATTTTCTTATTTACATAGGAGGTTTAACTGGCTTGTACTTGAATTCATCTAGATTTGTTTACATGACACATTTATGATGAGCAGCTTTCCACTTTG
TATTTCTATAGGAGGATTTTGAATTTAGGAGGATTTTCAATGATACCTTTTATTTACAGGATTAACATAAATAGTAAATGACTAAAAATTTCAATTTACTTATTTATTTGGATTTAACTTTAACTTTT
CCTCAACATTTCTAGGATTTAGCAGGAATACCAAGCAGATTTCTGATTTCTGAGCTTATACATCATGAAATTTATCTCCAGAAATTTGGATCAACAATTTCAATCAGAGGAAATTTATTTATTTTATTTAT
AGAAAGAATAATTAAGCAACGAATGATATTTTGAACAATAATAAGAAGATCAACAGAAATGATTACAAAATAATCCACAGCAGAACACAGACTCAGAACTCACCAATTTATAGAT
```

外群:

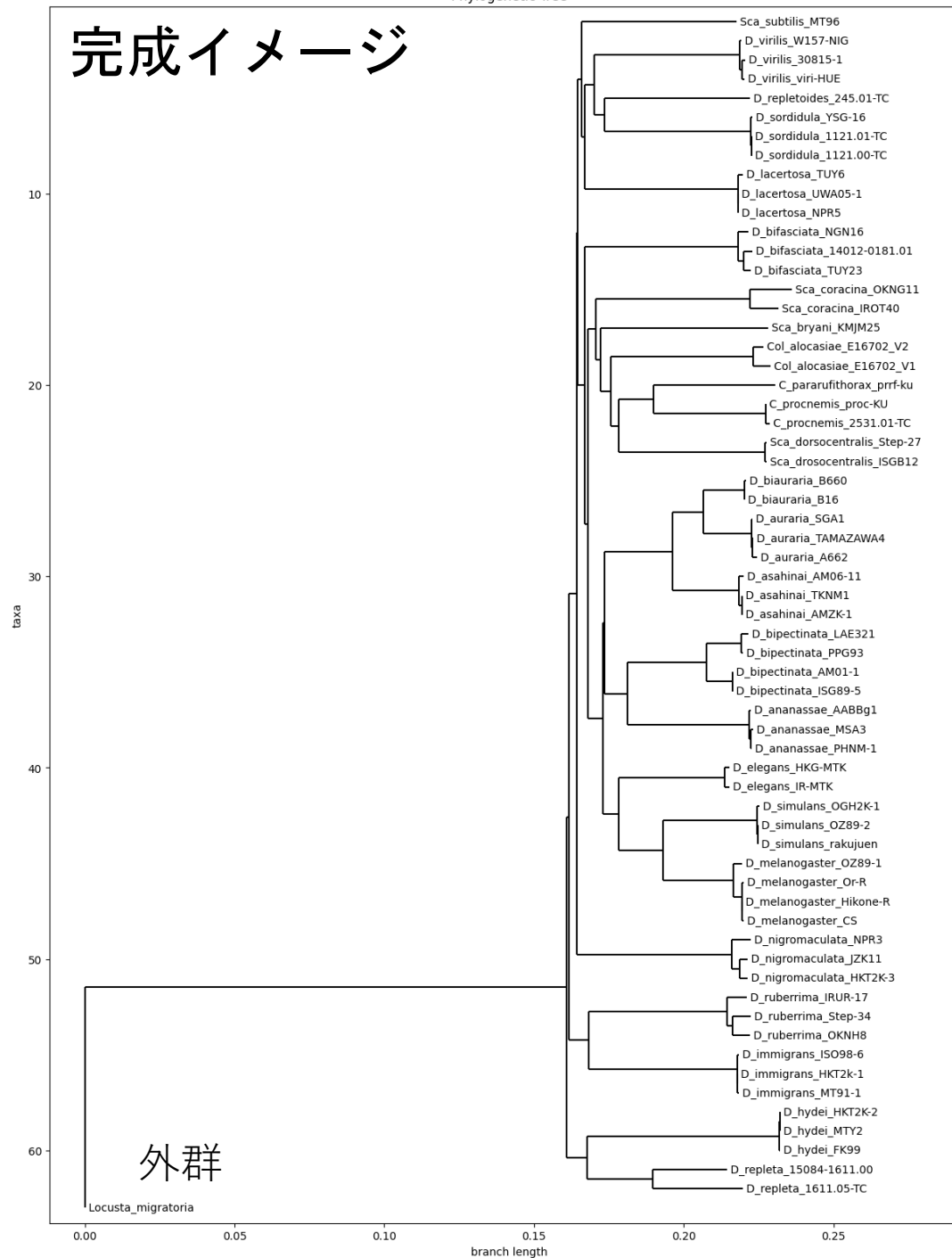
Locusta migratoria
(トノサマバツタ)

基本的なコードは同じだが、途中で外群を指定する

```
# 外群の指定 (外群となる配列IDを指定)
outgroup = 'Locusta_migratoria'
tree.root_with_outgroup(outgroup)
```

※どの段階で外群を指定すればいいかは考えてみてください

完成イメージ



外群

応用編: 他の手法で系統樹作成

系統樹の作成方法は他にもある

- 距離行列法
 - 近隣結合法（NJ法）→今回の演習で使用
 - 平均距離法（UPGMA法）→同じようにできる
- 形質状態法
 - 最節約法（MP法）
 - 最尤法（ML法）
(↑これらはちょっと大変です)

完成イメージ(UPGMA法)

