session3_clustering_demo

September 20, 2019

1 Machine Learning Demo - Depict a Phylogenetic Tree with Hierarchical clustering

Kim Hee (Graduate research assistant) Universitätsmedizin Mannheim, Mannheim (UMM)

- This is prepared for `Data analysis tools (Datenanalysewerkzeuge)` at MIRACUM summer school 2019

1.1 How can we relate different species together?

In the decades before DNA sequencing was reliable, the scientists struggled to answer a seemingly simple question: Are giant pandas closer to bears or racoons? (Ref: https://t.ly/Ld1p8)

1.1.1 Protocol:

1. Import required libraries 2. Load RNA data (download) 3. Sequence alignment - calculate the similarities of sequence 4. Visualize the result in dendrogram (phylogenetic tree)

[39]:
```
import os
from Bio import SeqIO, Phylo
from Bio.Seq import Seq
from Bio.SeqRecord import SeqRecord
from Bio.Alphabet import generic_rna, generic_protein
from Bio.Align.Applications import ClustalwCommandline
```

Biopython is a freely available tool for biological computation written in Python by an international team of developers.

1.1.2 Protocol:

1. Import required libraries 2. Load RNA data (download) 3. Sequence alignment - calculate the similarities of sequence 4. Visualize the result in dendrogram (phylogenetic tree)

[53]:
```
root_dir = "data"
filename = 'rRNA'
FILE_PATH = os.path.join(root_dir, f'{filename}.fasta')
```

```
records = list(SeqIO.parse(FILE_PATH, "fasta"))
records
```

[53]:
```
[SeqRecord(seq=Seq('ACCCAAAGCUAGCCCAAGCAACAAUGACUAGUAAAACCAUUAUGAAACAUUCAA…UUU
', SingleLetterAlphabet()), id='panda', name='panda', description='panda
Species_panda', dbxrefs=[]),
 SeqRecord(seq=Seq('AACUAAAACUAGCCCAAACAACAAUCAAUUAAAACUACUACACACAACAAUUAA…CUU
', SingleLetterAlphabet()), id='raccoon', name='raccoon', description='raccoon
Species_raccoon', dbxrefs=[]),
 SeqRecord(seq=Seq('GCCCAGAGCUAGCCCAGACAAUAACCAAUCAAACUACCACAGGCCAAUUAAAUA…UUU
', SingleLetterAlphabet()), id='black', name='black', description='black bear
Species_black bear', dbxrefs=[])]
```

FASTA Format * standard text-based format in bioinformatics for representing either nucleotide/ gene sequences or amino acid/ protein sequences using one letter code * describes one/ more sequence entries * each sequence entry includes a header (starting with ">") and the actual sequence, optional comments (starting with "#")

1.1.3 Protocol:

1. Import required libraries 2. Load RNA data (download) 3. Sequence alignment - calculate the similarities of sequence 4. Visualize the result in dendrogram (phylogenetic tree)

Clustal * it is a series of widely used computer programs used in Bioinformatics for multiple sequence alignment. There have been many versions of Clustal over the development of the algorithm that are listed below. * it is a general purpose DNA or protein multiple sequence alignment program for three or more sequences.

e.g. unit edit distance (Levenshtein distance) between "kitten" and "sitting" is 3 1. kitten → sitten (substitution of "k" by "s" at pos 1) 2. sitten → sittin (substitution of "e" by "i" at pos 5) 3. sittin → sitting (insertion of "g" at the end)

[46]:
```
clustalw_cline = ClustalwCommandline("clustalw2", infile=FILE_PATH)
stdout, stderr = clustalw_cline()
print(stdout)
```

```


 CLUSTAL 2.1 Multiple Sequence Alignments


Sequence format is Pearson
Sequence 1: panda       1583 bp
Sequence 2: raccoon     1587 bp
Sequence 3: black       1582 bp
Start of Pairwise alignments
```

```
Aligning…

Sequences (1:2) Aligned. Score:  86
Sequences (1:3) Aligned. Score:  90
Sequences (2:3) Aligned. Score:  87
Guide tree file created:   [data/rRNA.dnd]

There are 2 groups
Start of Multiple Alignment

Aligning…
Group 1: Sequences:   2      Score:27941
Group 2: Sequences:   3      Score:26454
Alignment Score 22117

CLUSTAL-Alignment file created  [data/rRNA.aln]
```

1.1.4 Protocol:

1. Import required libraries 2. Load RNA data (download) 3. Sequence alignment - calculate the similarities of sequence 4. Visualize the result in dendrogram (phylogenetic tree)

[51]:
```
newick_path = os.path.join(root_dir, f'{filename}.dnd')
tree = Phylo.read(newick_path, "newick")
Phylo.draw_ascii(tree)
```

```
  ____________________________________________ panda
 |
_|_____________________________________________________________________ raccoon
 |
 |_____________________________________ black
```

[47]:
```
# import matplotlib.pyplot as plt
# %matplotlib inline
# tree.rooted = True
# Phylo.draw(tree, branch_labels=lambda c: c.branch_length)
```