Evolutionary Tree

Zidar Miha (63060317)

November 30, 2012

1 Introduction

The goal of this assignment was to download and compare a few mitochondrial sequences, based on *COX3* gene, with *Needleman-Wunsch* algorithm and to construct a dendrogram. The algorithm used a *Blosum50* table and a linear gap penalty.

2 Data

We used sequences of 14 different animals, show in the Animal table (see Table 1), that we downloaded from `http://www.ncbi.nlm.nih.gov/genbank/`. From the entire mitochondrial genome, we took out only *COX3* gene for comparison. We also used the *BloSum50* table, for getting the comparison costs for all Amino Acid pairs. You can find all *BloSum* tables at `ftp://ftp.ncbi.nih.gov/blast/matrices/`

Table 1: Table of animal species used.

Index	GeneBank id	English name	Latin name
1	NC_000845.1	pig	Sus scrofa
2	NC_004299.1	Fugu rubripes	Takifugu rubripes
3	AC_000022.2	Norway rat	Rattus norvegicus
4	NC_002083.1	Sumatran orangutan	Pongo abelii
5	NC_001643.1	chimpanzee	Pan troglodytes
6	NC_011137.1	Neandertal	Homo sapiens neanderthalensis
7	NC_012920.1	human	Homo sapiens
8	NC_001645.1	western gorilla	Gorilla gorilla
9	NC_002008.4	dog	Canis lupus familiaris
10	NC_006580.1	goldfish	Carassius auratus auratus
11	NC_012420.1	veiled chameleon	Chamaeleo calyptratus
12	NC_011391.1	Russell's viper	Daboia russellii
13	NC_012061.1	Longbeaked common dolphin	Delphinus capensis
14	NC_001640.1	horse	Equus caballus

3 Methods

We used *Needleman-Wunsch* algorithm with a fixed linear gap penalty $d = -11$ and *Blosum50* scoring matrix, for global alignment of two sequences.

Here we have a simple pseudocode implementation of the *Needleman-Wunsch* algorithm

```
for i=0 to len(s)
  F(i,0) := d*i
for j=0 to len(t)
  F(0,j) := d*j
for i=1 to len(s)
  for j=1 to len(t)
  {
    Match := F(i-1,j-1) + S(s[i], t[j])
    Delete := F(i-1, j) + d
    Insert := F(i, j-1) + d
    F(i,j) := max(Match, Insert, Delete)
  }
```

Where s and t are input strings, S is the blosum cost table and F is our cost Matrix. The final comparison score is stored in the last element of F matrix on $F[len(s), len(t)]$.

Testing the algorithm with different gap penalties didn't show any significant difference, if we kept the penalty withing the absolute bound of the maximum absolute value in the *Blosum50* matrix.

4 Results

Here we have a pairwise score comparions of all 14 animals (see Table 4) and a dendrogram (see Figure-1) showing the animals in groups in accordance with their similarity score.

1.	1814													
2.	1574	1826												
3.	1641	1545	1816											
4.	1592	1531	1586	1804										
5.	1613	1564	1643	1703	1820									
6.	1597	1535	1614	1702	1766	1816								
7.	1622	1546	1625	1713	1777	1798	1823							
8.	1603	1542	1611	1727	1757	1753	1764	1814						
9.	1665	1555	1672	1567	1603	1585	1608	1570	1832					
10.	1582	1743	1552	1551	1600	1574	1585	1583	1556	1824				
11.	1305	1305	1320	1264	1291	1278	1299	1307	1296	1305	1827			
12.	1401	1414	1405	1386	1409	1395	1406	1422	1381	1417	1290	1796		
13.	1698	1547	1628	1580	1622	1615	1640	1607	1682	1553	1329	1395	1826	
14.	1718	1556	1658	1585	1606	1604	1615	1608	1688	1564	1310	1410	1685	1815
	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.

Table 2: Table comparison scores for all animal pairs. See Animal table 1 for index description

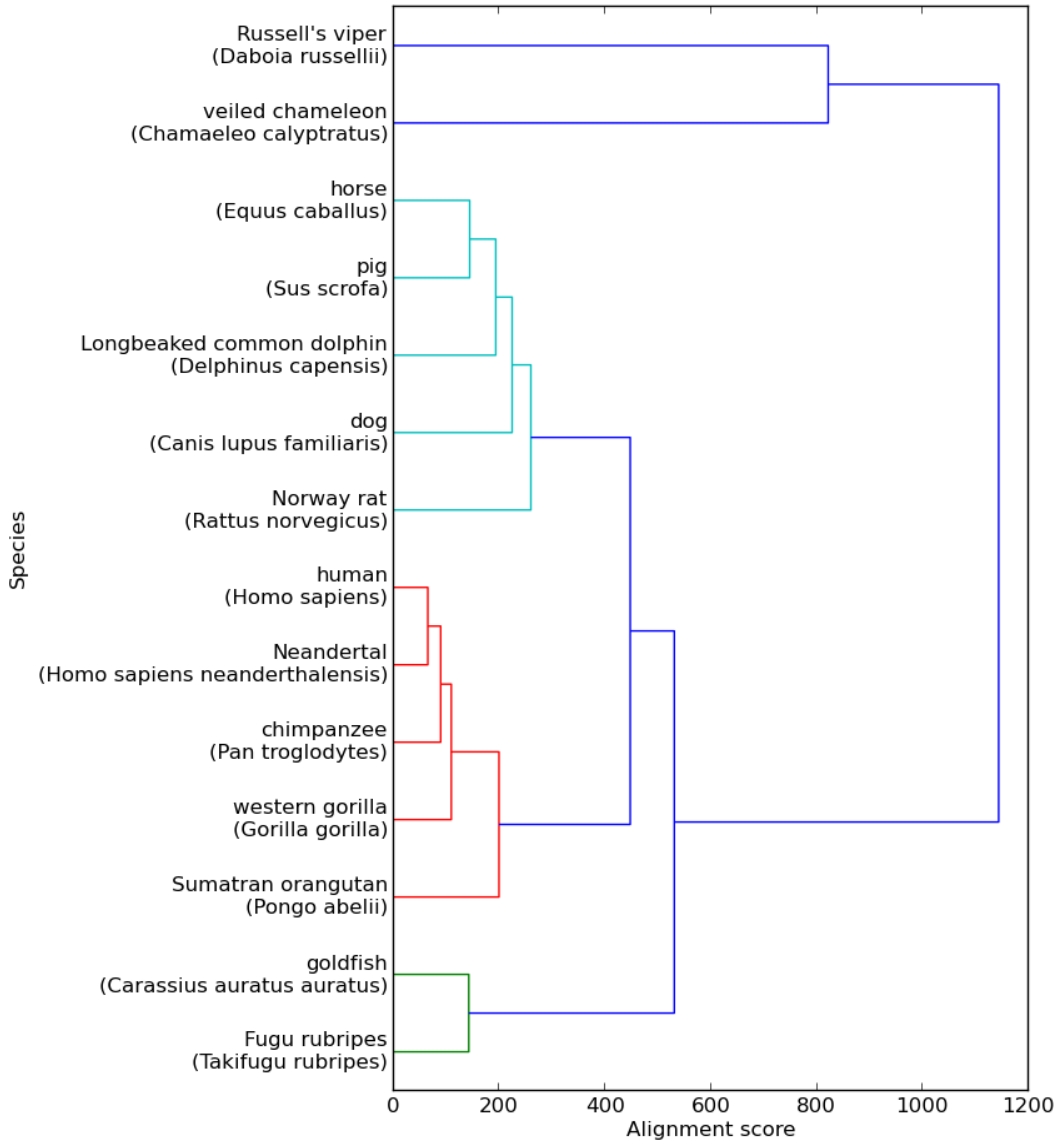

Figure 1: Every figure should include a caption with a figure description.

Honor Code

My answers to homework are my own work. I did not make solutions or code available to anyone else. I did not engage in any other activities that will dishonestly improve my results or dishonestly improve/hurt the results of others.