EEOB_563_assignment4

Devin Molnau

February 26, 2019

EEOB563 assignment 4

Question 1:

To get a Jukes Cantor distance matrix we can read in the data as sequences. We can use the library ape and the function dist.dna() to get the JC matrix.

```
sequences<-read.FASTA("sequences_1a.txt", type="DNA")
dist.dna(sequences,model="JC69")
```

```
##             Dog       Cat     Mouse       Pig
## Cat   0.1468084
## Mouse 0.1468084 0.2326162
## Pig   0.1073256 0.1073256 0.1468084
## Human 0.3295250 0.3831192 0.3831192 0.3831192
```

To calulate the distances manually you will need to count the number of distances between every pair of sequences and then from that you can calculate the distances.

```
seq1a<-read.dna("sequences_1a.txt", format = "fasta", as.character = TRUE, as.matrix = TRUE)
comparing<-function(string1, string2){
  counter=0
  for (i in 1:length(sequences[[1]])){
    if (string1[i]!=string2[i]){
      counter=counter+1
    }
  }
  return(counter)
}

get_distances<-function(matrix_of_interest){
  l<-length(row.names(matrix_of_interest))
  a<-matrix(data=NA,nrow=length(row.names(matrix_of_interest)),
            ncol=length(row.names(matrix_of_interest)),
            dimnames = list(row.names(matrix_of_interest),
                            row.names(matrix_of_interest)))
  for (i in 1:l){
    for (j in 1:l){
    value<-comparing(matrix_of_interest[i,],matrix_of_interest[j,])
    a[i,j]<-value
    a[j,i]<-value
    }
  }
  return(a)
  }
a<-get_distances(seq1a)

cat("Differences in amino acid sequences Question#1")
```

```
## Differences in amino acid sequences Question#1
```

```
cat("\n")
```

```
a #Differences in amino acid sequences Question#1
```

```
##       Dog Cat Mouse Pig Human
## Dog     0   4     4   3     8
## Cat     4   0     6   3     9
## Mouse   4   6     0   4     9
## Pig     3   3     4   0     9
## Human   8   9     9   9     0
```

```
a_prob<-a/(length(seq1a[1,]))
#a_prob<-formatC(a_prob, digits = 5, format = "f")
distance_matrix<-(-3/4)*log(1-((4/3)*(a_prob)))
cat("------------------------------------------\n")
```

```
## ------------------------------------------
```

```
cat("Distance matrix Question#1")
```

```
## Distance matrix Question#1
```

```
cat("\n")
```

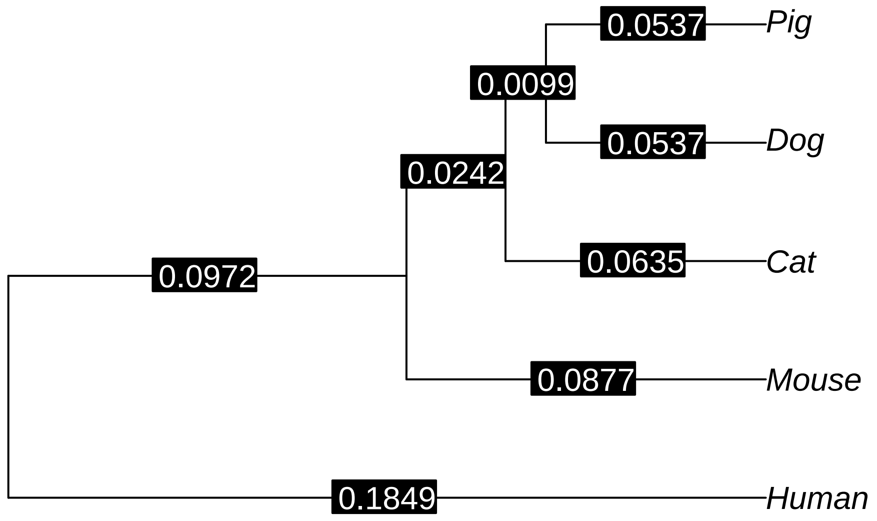
```
distance_matrix #Distance matrix Question#1
```

```
##             Dog       Cat     Mouse       Pig     Human
## Dog   0.0000000 0.1468084 0.1468084 0.1073256 0.3295250
## Cat   0.1468084 0.0000000 0.2326162 0.1073256 0.3831192
## Mouse 0.1468084 0.2326162 0.0000000 0.1468084 0.3831192
## Pig   0.1073256 0.1073256 0.1468084 0.0000000 0.3831192
## Human 0.3295250 0.3831192 0.3831192 0.3831192 0.0000000
```

Question 2:

Using this distance matrix we can build the UPGMA tree.

```
tree_upgma<-upgma(D=distance_matrix)
plot(tree_upgma, type="phylogram", use.edge.length = TRUE, show.tip.label = TRUE, show.node.label = TRU
edgelabels(formatC(tree_upgma$edge.length, digits = 4, format = "f"), bg="black", col="white", font=0.2
```



```
#tree_upgma$edge.length
```

To do this by hand you would combine the columns with the smallest distance together and then recalculate the distances after that each time. Work attached to end of document.

Question 3:

To create a neighbor joining tree we will take the distance matrix tree from question 1. We will average the row values and then generate a new matrix of the intersect of the cell-the row average- the row average for the other intersected taxa. Using the minimum of this new matrix (called U) we will new which two taxa are the nearest neighbor of one another. We will then recalculate the distance between our grouped taxa and the other taxa and restart the process again.

```
#length(distance_matrix[1,])
average_dist_matrix<-function(dist_mat){
  average_dist<-c()
  for (i in 1:length(dist_mat[1,])){
    average_dist[i]<-sum(dist_mat[i,])/((length(dist_mat[1,]))-2)
  }
  #average_dist
  dist_mat_alter<-cbind(dist_mat,average_dist)
  return(dist_mat_alter)
}

dist_mat_alter<-average_dist_matrix(distance_matrix)
cat("------------------------------------------\n")
```

```
## ------------------------------------------
```

```r
dist_mat_alter #The new distance matrix with the row averages added
```

```
##             Dog       Cat     Mouse       Pig     Human average_dist
## Dog   0.0000000 0.1468084 0.1468084 0.1073256 0.3295250    0.2434892
## Cat   0.1468084 0.0000000 0.2326162 0.1073256 0.3831192    0.2899565
## Mouse 0.1468084 0.2326162 0.0000000 0.1468084 0.3831192    0.3031174
## Pig   0.1073256 0.1073256 0.1468084 0.0000000 0.3831192    0.2481930
## Human 0.3295250 0.3831192 0.3831192 0.3831192 0.0000000    0.4929609
```

```r
cat("------------------------------------------\n")
```

```
## ------------------------------------------
```

```r
u_matrix<-function(dist_mat){
  new_dim_names<-unlist(list(row.names(dist_mat)))
  #new_dim_names<-unlist(list(row.names(dist_mat_alter)))
  dimension<-length(dist_mat[,1])
  #dimension<-length(dist_mat_alter[,1])
  which_next<-matrix(NA,nrow=dimension,ncol = dimension,
       dimnames = list(new_dim_names,new_dim_names))
which_next
for (i in 1:length(which_next[1,])){
  for (j in 1:length(which_next[1,])){
    if (i==j){
      which_next[i,j]<-0
    }
    else{
      value<-dist_mat[i,j]-
        dist_mat[i,length(dist_mat[1,])]-
        dist_mat[j,length(dist_mat[1,])]
      which_next[i,j]<-value
    }
  }
}
return(which_next)
}

u_matrix(dist_mat_alter) #This return the matrix to tell you which branch to cluster next
```

```
##              Dog        Cat      Mouse        Pig      Human
## Dog    0.0000000 -0.3866372 -0.3997982 -0.3843565 -0.4069251
## Cat   -0.3866372  0.0000000 -0.3604577 -0.4308238 -0.3997982
## Mouse -0.3997982 -0.3604577  0.0000000 -0.4045020 -0.4129591
## Pig   -0.3843565 -0.4308238 -0.4045020  0.0000000 -0.3580346
## Human -0.4069251 -0.3997982 -0.4129591 -0.3580346  0.0000000
```

Because the smallest element of the u matrix is the intersection between Cat and Pig, those taxa will be the first to merge. I create a new distance matrix with of those two columns merged and then start the process again.

NJ distance matrix with four taxa groups now.

```r
new_dim_names<-list("Dog","Pig/Cat","Mouse","Human")
nj_matrix_1<-matrix(data = c(0,0.0735,0.147,0.329,0.0735,0,0.1365,0.3295,0.147,0.1365,0,0.383,0.329,0.32
                    nrow = 4, ncol = 4, dimnames = list(new_dim_names,new_dim_names))
cat("NJ distance matrix with 4 taxa\n")
```

```
## NJ distance matrix with 4 taxa
```

```
nj_matrix_1 #NJ distance matrix with 4 taxa
```

```
##            Dog Pig/Cat  Mouse  Human
## Dog     0.0000  0.0735 0.1470 0.3290
## Pig/Cat 0.0735  0.0000 0.1365 0.3295
## Mouse   0.1470  0.1365 0.0000 0.3830
## Human   0.3290  0.3295 0.3830 0.0000
```

```
aver_nj_mat_1<-average_dist_matrix(nj_matrix_1)
cat("------------------------------------------\n")
```

```
## ------------------------------------------
```

```
cat("Altered distance 4 taxa matrix with row averages\n")
```

```
## Altered distance 4 taxa matrix with row averages
```

```
aver_nj_mat_1 #Altered distance 4 taxa matrix with row averages
```

```
##            Dog Pig/Cat  Mouse  Human average_dist
## Dog     0.0000  0.0735 0.1470 0.3290      0.27475
## Pig/Cat 0.0735  0.0000 0.1365 0.3295      0.26975
## Mouse   0.1470  0.1365 0.0000 0.3830      0.33325
## Human   0.3290  0.3295 0.3830 0.0000      0.52075
```

```
cat("------------------------------------------\n")
```

```
## ------------------------------------------
```

```
cat("U matrix of 4 taxa distance matrix\n")
```

```
## U matrix of 4 taxa distance matrix
```

```
u_matrix(aver_nj_mat_1) # U matrix of 4 taxa distance matrix
```

```
##             Dog Pig/Cat   Mouse   Human
## Dog      0.0000 -0.4710 -0.4610 -0.4665
## Pig/Cat -0.4710  0.0000 -0.4665 -0.4610
## Mouse   -0.4610 -0.4665  0.0000 -0.4710
## Human   -0.4665 -0.4610 -0.4710  0.0000
```

```
branch_length_nj<-function(mat,i,j){
  value<-((1/2)*(mat[i,j]))+((1/2)*(aver_nj_mat_1[i,length(aver_nj_mat_1[1,])]-aver_nj_mat_1[j,length(a
  return(value)
}
#branch_length_nj(aver_nj_mat_1,2,1)
#branch_length_nj(aver_nj_mat_1,1,2)
#branch_length_nj(aver_nj_mat_1,3,4)
#branch_length_nj(aver_nj_mat_1,4,3)
```

Next JC distance matrix with 3 taxa groups now.

```
new_dim_names<-list("Dog","Pig/Cat","Mouse/Human")
nj_matrix_2<-matrix(data = c(0,0.0735,0.044,0.0735,0,0.04175,0.044,0.04175,0),
                    nrow = 3, ncol = 3, dimnames = list(new_dim_names,new_dim_names))
cat("NJ distance matrix with 3 taxa groups\n")
```

```
## NJ distance matrix with 3 taxa groups
```

```r
nj_matrix_2 #NJ distance matrix with 3 taxa groups
```

```
##                Dog Pig/Cat Mouse/Human
## Dog         0.0000 0.07350     0.04400
## Pig/Cat     0.0735 0.00000     0.04175
## Mouse/Human 0.0440 0.04175     0.00000
```

```r
cat("-------------------------------------------\n")
```

```
## -------------------------------------------
```

```r
aver_nj_mat_2<-average_dist_matrix(nj_matrix_2)
cat("NJ distance matrix with 3 taxa groups and row averages\n")
```

```
## NJ distance matrix with 3 taxa groups and row averages
```

```r
aver_nj_mat_2 #NJ distance matrix with 3 taxa groups and row averages
```

```
##                Dog Pig/Cat Mouse/Human average_dist
## Dog         0.0000 0.07350     0.04400      0.11750
## Pig/Cat     0.0735 0.00000     0.04175      0.11525
## Mouse/Human 0.0440 0.04175     0.00000      0.08575
```

```r
cat("-------------------------------------------\n")
```

```
## -------------------------------------------
```

```r
u_matrix(aver_nj_mat_2)
```

```
##                  Dog  Pig/Cat Mouse/Human
## Dog          0.00000 -0.15925    -0.15925
## Pig/Cat     -0.15925  0.00000    -0.15925
## Mouse/Human -0.15925 -0.15925     0.00000
```

```r
#branch_length_nj(nj_matrix_2,1,2) #distance between dog and the internal node pig/cat
#branch_length_nj(nj_matrix_2,2,1) #distance between pig/cat internal node and dog
```

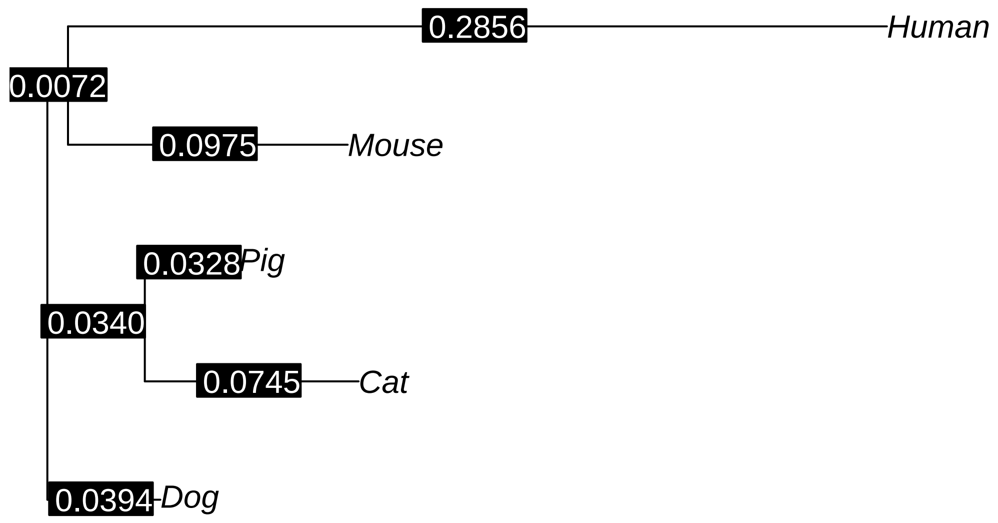
To get the absolute last matrix, there is no difference is merging dog with cat and pig or merging dog with human and mouse. I am chosing to merge dog with cat and pig.

```r
new_dim_names<-list("Dog/Pig/Cat","Mouse/Human")
nj_matrix_3<-matrix(data = c(0,0.006125, 0.006125,0),
                    nrow = 2, ncol = 2, dimnames = list(new_dim_names,new_dim_names))
nj_matrix_3
```

```
##             Dog/Pig/Cat Mouse/Human
## Dog/Pig/Cat    0.000000    0.006125
## Mouse/Human    0.006125    0.000000
```

To verify our results, we can build the NJ tree using the ape package's NJ() function.

```r
neighbor_joining_tree<-NJ(distance_matrix)
plot(neighbor_joining_tree,type="phylogram")
edgelabels(formatC(neighbor_joining_tree$edge.length, digits = 4, format = "f"), bg="black", col="white
```



##Question 4: The rest of this assignment is done in the commandline on an hpc.The multiple sequence alignment is first generated using MAFFT and saved in a phylip format. Then we load in phylip and generate a distance matrix.Using the program dnadist within phylip, we can create several different distance matrices: Jukes-Cantor, Kimura, Log and F84.With these distance matrices we can create a neighbor joining tree of both and then create a strict consensus tree using consense.

```
module load mafft
mafft --nuc --phylipout cob_nt.fasta > cob_nt_aln.phylip
module load  phylip
dnadist cob_nt_aln.phylip
neighbor
consense #strict, root
```

Question 5:

Majority consensus tree

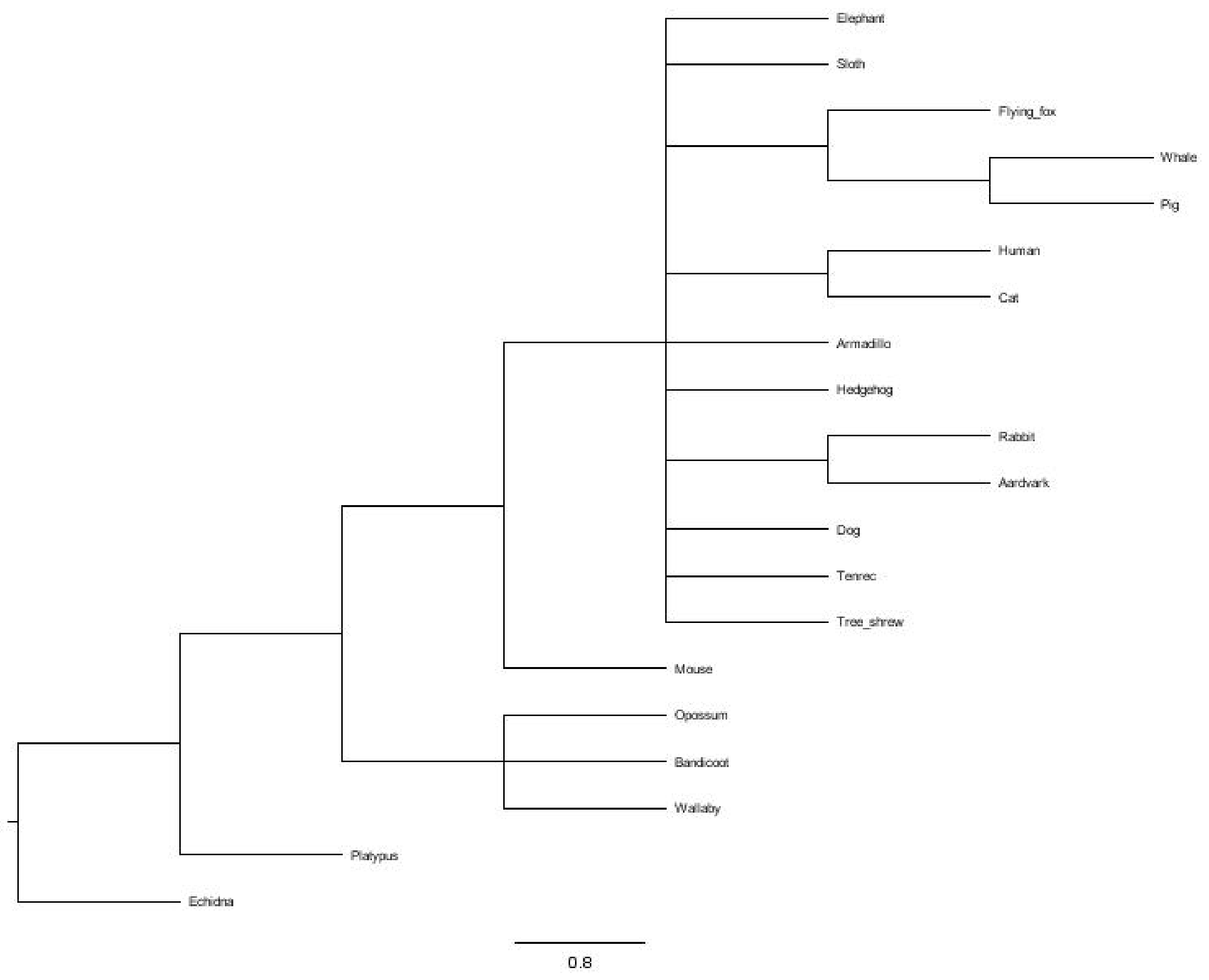
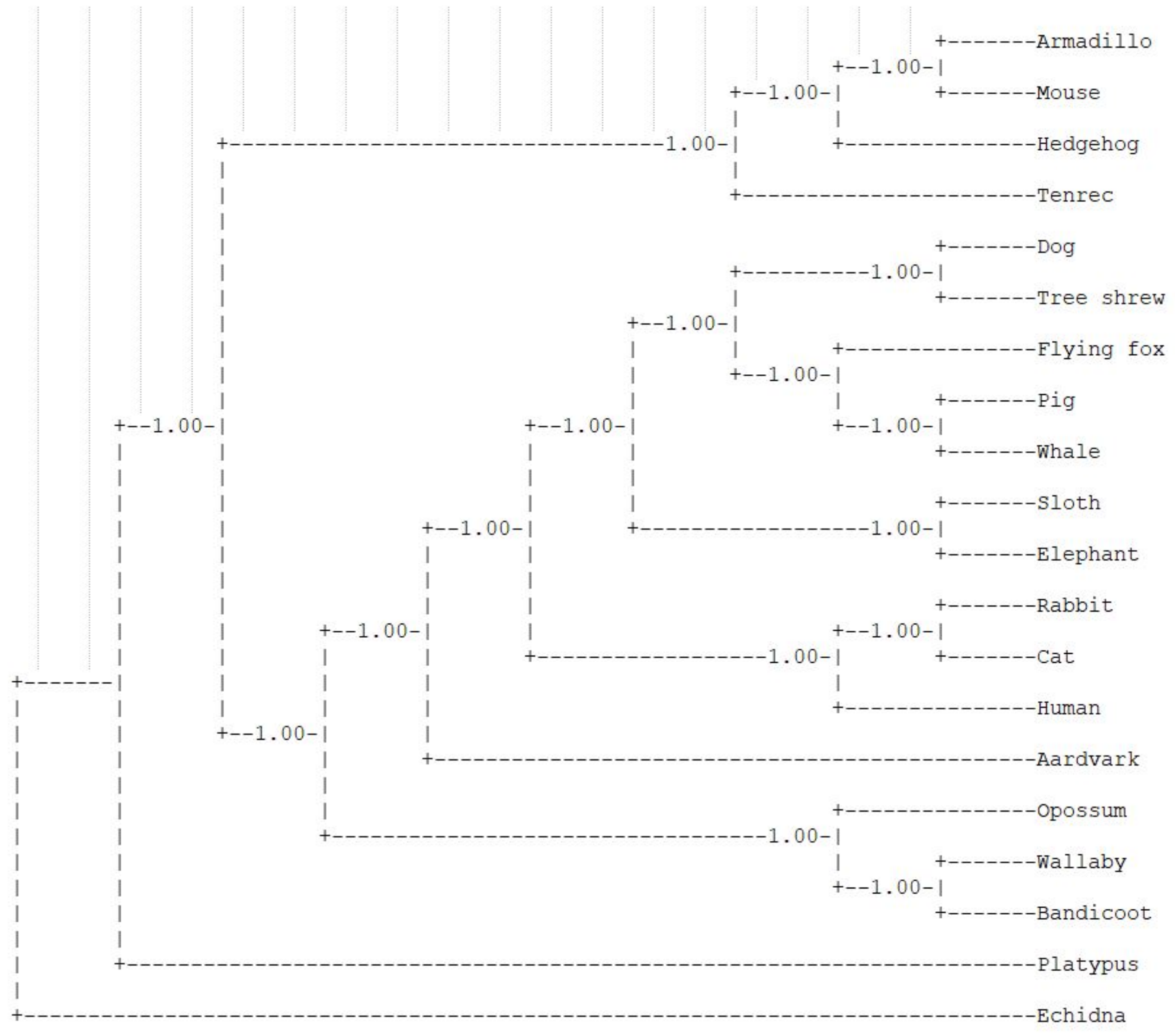


Figure 1: Consensus tree (question 4)



```
                                                                        +-------Armadillo
                                                                +--1.00-|
                                                        +--1.00-|       +-------Mouse
                                                        |       |
                +----------------------------------1.00-|       +---------------Hedgehog
                |                                       |
                |                                       +-----------------------Tenrec
                |
                |                                                       +-------Dog
                |                                       +----------1.00-|
                |                                       |               +-------Tree shrew
                |                               +--1.00-|
                |                               |       |       +---------------Flying fox
                |                               |       +--1.00-|
                |                               |               |       +-------Pig
        +--1.00-|                       +--1.00-|               +--1.00-|
        |       |                       |       |                       +-------Whale
        |       |                       |       |
        |       |                       |       |                       +-------Sloth
        |       |               +--1.00-|       +------------------1.00-|
        |       |               |       |                               +-------Elephant
        |       |               |       |
        |       |               |       |                               +-------Rabbit
        |       |       +--1.00-|       |                       +--1.00-|
+-------|       |       |       |       +------------------1.00-|       +-------Cat
|       |       |       |       |                               |
|       |       |       |       |                               +---------------Human
|       |       +--1.00-|       |
|       |               |       +-----------------------------------------------Aardvark
|       |               |
|       |               |                                       +---------------Opossum
|       |               +----------------------------------1.00-|
|       |                                                       |       +-------Wallaby
|       |                                                       +--1.00-|
|       |                                                               +-------Bandicoot
|       |
|       +-----------------------------------------------------------------------Platypus
|
+-------------------------------------------------------------------------------Echidna
```

Figure 2: Majority tree (question 5)