

A7_ZHANG_20158140

Qiuhan Zhang

2023-02-28

My repository (<https://github.com/qiuhan1008/Assignment7.git>)

Import Nexus Data

```
library(dplyr)
library(Biostrings)
library(ape)
DragonNexus<-read.nexus.data("input/DragonMatrix.nex")
head(DragonNexus)
```

```

## $`0.1FishXXX`
## [1] "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"
## [20] "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"
## [39] "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"
## [58] "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0" "0"
## [77] "0" "0"
##
## $`0.2SnakeXX`
## [1] "1" "1" "1" "1" "0" "0" "0" "0" "1" "1" "1" "0" "0" "1" "1" "1" "0" "1" "1"
## [20] "0" "0" "0" "0" "0" "0" "0" "1" "1" "0" "0" "0" "0" "0" "0" "1" "0" "0" "0"
## [39] "1" "0" "0" "0" "1" "0" "0" "0" "1" "0" "0" "0" "0" "0" "0" "1" "1" "1" "1"
## [58] "1" "1" "0" "0" "0" "0" "0" "0" "0" "0" "1" "0" "0" "0" "0" "0" "0" "0" "0"
## [77] "0" "0"
##
## $`0.3MammalX`
## [1] "1" "0" "0" "0" "0" "0" "0" "0" "0" "0" "1" "1" "1" "0" "0" "0" "0" "0" "0"
## [20] "0" "0" "0" "1" "0" "0" "0" "1" "1" "1" "0" "0" "0" "0" "0" "1" "1" "0" "0"
## [39] "1" "1" "0" "0" "1" "1" "0" "0" "0" "0" "0" "0" "0" "0" "0" "1" "0" "0" "0"
## [58] "0" "0" "0" "1" "1" "0" "0" "0" "0" "0" "1" "0" "0" "0" "0" "0" "1" "0" "0"
## [77] "0" "0"
##
## $`1GermanXXX`
## [1] "0" "1" "0" "0" "1" "1" "1" "0" "0" "1" "1" "1" "1" "0" "0" "0" "0" "1" "1"
## [20] "0" "0" "0" "0" "1" "0" "1" "1" "1" "1" "1" "0" "0" "0" "0" "1" "1" "0" "1"
## [39] "0" "0" "0" "0" "1" "1" "0" "1" "1" "0" "0" "0" "0" "0" "0" "1" "1" "1" "0"
## [58] "0" "0" "0" "1" "0" "0" "0" "1" "0" "0" "1" "0" "?" "?" "?" "?" "0" "0" "1"
## [77] "1" "1"
##
## $`2FrenchXXX`
## [1] "0" "1" "0" "0" "1" "1" "0" "1" "0" "1" "1" "0" "0" "1" "1" "1" "0" "1" "0"
## [20] "0" "0" "0" "0" "0" "0" "1" "1" "1" "1" "1" "0" "0" "0" "0" "1" "1" "0" "0"
## [39] "1" "0" "0" "0" "1" "1" "0" "0" "0" "0" "0" "0" "0" "0" "0" "1" "1" "0" "0"
## [58] "0" "0" "1" "0" "0" "1" "0" "0" "0" "1" "1" "0" "0" "0" "1" "1" "0" "0" "1"
## [77] "1" "0"
##
## $`3FrenchXXX`
## [1] "0" "1" "0" "0" "1" "1" "1" "0" "0" "0" "1" "1" "0" "0" "0" "0" "0" "0" "?"
## [20] "?" "?" "1" "0" "1" "0" "0" "1" "1" "1" "1" "0" "0" "0" "0" "1" "1" "0" "1"
## [39] "1" "1" "0" "1" "1" "1" "0" "1" "0" "0" "0" "0" "0" "0" "0" "1" "1" "1" "0"
## [58] "0" "0" "0" "1" "0" "0" "0" "1" "0" "0" "1" "0" "0" "0" "1" "1" "0" "0" "1"
## [77] "1" "1"

```

names(DragonNexus)

```
## [1] "0.1FishXXX" "0.2SnakeXX" "0.3MammalX" "1GermanXXX" "2FrenchXXX"
## [6] "3FrenchXXX" "4DutchXXXX" "5EnglishXX" "6AmericanX" "7FrenchXXX"
## [11] "8EnglishXX" "9FrenchXXX" "10FrenchXX" "11SpanishX" "12Japanese"
## [16] "13Japanese" "14Japanese" "15Japanese" "16Japanese" "17Japanese"
## [21] "18Japanese" "19Japanese" "20Japanese" "21Japanese" "22Japanese"
## [26] "23Japanese" "24Japanese" "25Japanese" "26Japanese" "27Japanese"
## [31] "28Japanese" "29Japanese" "30ItalianX" "31ItalianX" "32ItalianX"
## [36] "33XXXXXXXX" "34GermanXX" "35EnglishX" "36GermanXX" "37DutchXXX"
## [41] "38SpanishX" "39ItalianX" "40ItalianX" "41EnglishX" "42ItalianX"
## [46] "43SpanishX" "44ItalianX" "45ItalianX" "46EnglishX" "47ItalianX"
## [51] "48DutchXXX" "49IndianXX" "50Japanese" "51Japanese" "52Japanese"
## [56] "53Japanese" "54IranianX" "55IranianX" "56IranianX" "57IranianX"
## [61] "58TurkishX" "59IranianX" "60IranianX" "61TurkishX" "62TurkishX"
## [66] "63UkraineX" "64UkraineX" "65RussiaXX" "66UkraineX" "67RussiaXX"
## [71] "68GreeceXX" "69ItalianX" "70American" "71BritishX" "72BritishX"
## [76] "73BritishX" "74BritishX" "75Toothles" "76SpikeXXX" "77Dragonit"
```

##Distance matrix

```
DragonNexusDF<-data.frame(matrix(unlist(DragonNexus), ncol=78,byrow=T))
row.names(DragonNexusDF)<-names(DragonNexus)
head(DragonNexusDF)
```

```
##          X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13 X14 X15 X16 X17 X18 X19
## 0.1FishXXX 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 0.2SnakeXX 1 1 1 1 0 0 0 0 1 1 1 0 0 1 1 1 0 1 1
## 0.3MammalX 1 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0
## 1GermanXXX 0 1 0 0 1 1 1 0 0 1 1 1 1 0 0 0 0 1 1
## 2FrenchXXX 0 1 0 0 1 1 0 1 0 1 1 0 0 1 1 1 0 1 0
## 3FrenchXXX 0 1 0 0 1 1 1 0 0 0 1 1 0 0 0 0 0 0 ?
##          X20 X21 X22 X23 X24 X25 X26 X27 X28 X29 X30 X31 X32 X33 X34 X35 X36
## 0.1FishXXX 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 0.2SnakeXX 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 0
## 0.3MammalX 0 0 0 1 0 0 0 0 1 1 1 0 0 0 0 0 1 1
## 1GermanXXX 0 0 0 0 1 0 1 1 1 1 1 1 0 0 0 0 1 1
## 2FrenchXXX 0 0 0 0 0 0 0 1 1 1 1 1 0 0 0 0 1 1
## 3FrenchXXX ? ? 1 0 1 0 0 1 1 1 1 1 0 0 0 0 1 1
##          X37 X38 X39 X40 X41 X42 X43 X44 X45 X46 X47 X48 X49 X50 X51 X52 X53
## 0.1FishXXX 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 0.2SnakeXX 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 0
## 0.3MammalX 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0
## 1GermanXXX 0 1 0 0 0 0 1 1 0 1 1 1 0 0 0 0 0
## 2FrenchXXX 0 0 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0
## 3FrenchXXX 0 1 1 1 0 1 1 1 0 1 0 0 0 0 0 0 0
##          X54 X55 X56 X57 X58 X59 X60 X61 X62 X63 X64 X65 X66 X67 X68 X69 X70
## 0.1FishXXX 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
## 0.2SnakeXX 1 1 1 1 1 1 0 0 0 0 0 0 0 0 1 0 0
## 0.3MammalX 1 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0
## 1GermanXXX 1 1 1 0 0 0 0 0 1 0 0 0 1 0 0 1 ?
## 2FrenchXXX 1 1 0 0 0 0 1 0 0 1 0 0 0 1 1 0 0
## 3FrenchXXX 1 1 1 0 0 0 0 0 1 0 0 0 1 0 0 1 0
##          X71 X72 X73 X74 X75 X76 X77 X78
## 0.1FishXXX 0 0 0 0 0 0 0 0
## 0.2SnakeXX 0 0 0 0 0 0 0 0
## 0.3MammalX 0 0 0 1 0 0 0 0
## 1GermanXXX ? ? ? 0 0 1 1 1
## 2FrenchXXX 0 1 1 0 0 1 1 0
## 3FrenchXXX 0 1 1 0 0 1 1 1
```

```
DragonDist<-dist(DragonNexusDF,method='binary')
```

```
## Warning in dist(DragonNexusDF, method = "binary"): NAs introduced by coercion
```

```
DragonDistMat<-as.matrix(DragonDist)
```

```
#Vsualize the matrix
```

```
WeightsDat<-read.csv("input/Weights.csv")
```

1. Create a single vector of weights.

```
Weights<-paste0(WeightsDat$Weight,collapse="")
Weights<-strsplit(Weights,split="")[[1]]
```

2. Convert each letter to a value.

```
LETTERS # See what LETTERS is (see also letters)
```

```
## [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S"
## [20] "T" "U" "V" "W" "X" "Y" "Z"
```

```
which(LETTERS=="G")
```

```
## [1] 7
```

```
WeightsNum<-rep(NA,length(Weights))
for(i in 1:length(WeightsNum)){
  if(Weights[i] %in% LETTERS){
    WeightsNum[i]<-which(LETTERS==Weights[i])+9
  } else {
    WeightsNum[i]<-Weights[i]
  }
}
WeightsNum<-as.numeric(WeightsNum)
```

```
length(WeightsNum)
```

```
## [1] 78
```

3. Multiply the weight value by the trait vector for each dragon.

```
WtDragonNexus<-DragonNexus # Make a new weighted data frame object
for (i in 1:length(DragonNexus)){
  RepWeight<-DragonNexus[[i]]==1
  WtDragonNexus[[i]][RepWeight]<-WeightsNum[RepWeight]
  RepWeight<-NA
}
```

4. Re-calculate our distance matrix

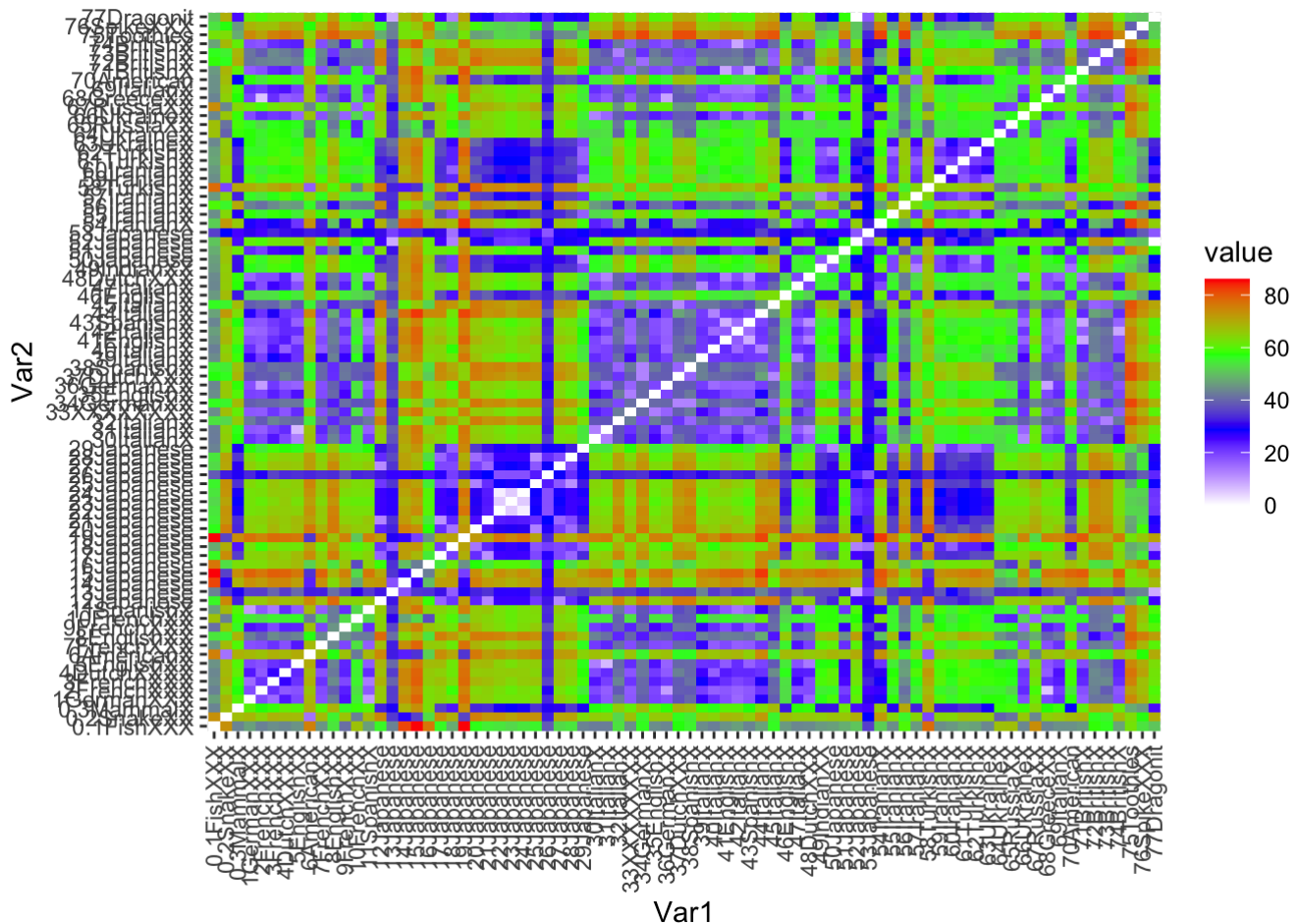
```
WtDragonNexusDF<-data.frame(matrix(unlist(WtDragonNexus),ncol=78,byrow=T))
row.names(WtDragonNexusDF)<-names(WtDragonNexus)
WtDragonDist<-dist(WtDragonNexusDF,method='euclidean')
```

```
## Warning in dist(WtDragonNexusDF, method = "euclidean"): NAs introduced by
## coercion
```

```
WtDragonDistMat<-as.matrix(WtDragonDist)
```

```
library(reshape2)
PDat<-melt(DragonDistMat)
```

```
library(ggplot2)
# rearrange the data from an n×n matrix to a n2×3 matrix
WtPDat <- melt(WtDragonDistMat)
# plot the matrix
ggplot(data = WtPDat, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradientn(colours = c("white","blue","green","red")) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```



Dragon Phylogeny

Tree Building

```
WtDragonTree<-fastme.bal(WtDragonDist)
WtDragonTreeNJ <- nj(WtDragonDist)
```

```
# check 'tip.labels'
head(WtDragonTree$tip.label)
```

```
## [1] "0.1FishXXX" "26Japanese" "15Japanese" "19Japanese" "14Japanese"
## [6] "0.2SnakeXX"
```

```
# remove leading numbers
Country <- gsub("[0-9\\.]+(^[^X]+)X*", "\\1", WtDragonTree$tip.label)
# replace the unwanted '3' with 'Unknown'
Country <- gsub("\\d", "Unknown", Country)
# group 'tip.labels' by their corresponding country
CountryGroups <- split(WtDragonTree$tip.label, Country)
names(CountryGroups)
```

```
## [1] "American" "British" "Dragonit" "Dutch" "English" "Fish"
## [7] "French" "German" "Greece" "Indian" "Iranian" "Italian"
## [13] "Japanese" "Mammal" "Russia" "Snake" "Spanish" "Spike"
## [19] "Toothles" "Turkish" "Ukraine" "Unknown"
```

```
library(ggtree)
```

```
## ggtree v3.7.1.002 For help: https://yulab-smu.top/treedata-book/
##
## If you use the ggtree package suite in published research, please cite
## the appropriate paper(s):
##
## Guangchuang Yu, David Smith, Huachen Zhu, Yi Guan, Tommy Tsan-Yuk Lam.
## ggtree: an R package for visualization and annotation of phylogenetic
## trees with their covariates and other associated data. Methods in
## Ecology and Evolution. 2017, 8(1):28-36. doi:10.1111/2041-210X.12628
##
## Guangchuang Yu. Data Integration, Manipulation and Visualization of
## Phylogenetic Trees (1st edition). Chapman and Hall/CRC. 2022,
## doi:10.1201/9781003279242
##
## Shuangbin Xu, Lin Li, Xiao Luo, Meijun Chen, Wenli Tang, Li Zhan, Zehan
## Dai, Tommy T. Lam, Yi Guan, Guangchuang Yu. Ggtree: A serialized data
## object for visualization of a phylogenetic tree and annotation data.
## iMeta 2022, 1(4):e56. doi:10.1002/imt2.56
```

```
##
## Attaching package: 'ggtree'
```

```
## The following object is masked from 'package:ape':
##
##      rotate
```

```
## The following object is masked from 'package:Biostrings':
##
##      collapse
```

```
## The following object is masked from 'package:IRanges':
##
##      collapse
```

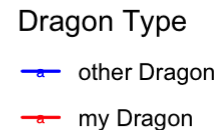
```
## The following object is masked from 'package:S4Vectors':
##
##      expand
```

```
# use the groupOTU function to apply the grouping information for plotting
WtDTcol <- groupOTU(WtDragonTree, CountryGroups)
str(WtDTcol)
```

```
## List of 4
## $ edge      : int [1:157, 1:2] 81 81 82 83 84 84 85 86 87 87 ...
## $ edge.length: num [1:157] 25.24 5.27 5.43 3.64 1.63 ...
## $ tip.label  : chr [1:80] "0.1FishXXX" "26Japanese" "15Japanese" "19Japanese" ...
## $ Nnode      : int 78
## - attr(*, "class")= chr "phylo"
## - attr(*, "order")= chr "cladewise"
## - attr(*, "group")= Factor w/ 22 levels "American","British",...: 6 13 13 13 13 16 1
  20 15 21 ...
```

Visualize the Phylogenetic Tree

```
# again use the groupOTU function to highlight the selected three groups
WtDTcolHi <- groupOTU(WtDTcol, c(CountryGroups$Toothles,
                                CountryGroups$Spike,
                                CountryGroups$Dragonit))
ggtree(WtDTcolHi, layout = "circular", aes(color = group)) +
  geom_tiplab(size = 2, aes(angle = angle)) +
  scale_colour_manual(name = "Dragon Type",
                      values = c("blue", "red"),
                      labels = c("other Dragon", "my Dragon"))
```

Report

A black dragon with large, dark wings and a long, spiky tail. It has yellow eyes and a wide, open mouth showing its tongue. The dragon is shown in a flying pose against a plain white background.

Toothless from How To Train Your Dragon. Credit
to:[https://howtotrainyourdragon.fandom.com/wiki/Toothless_\(Franchise\)](https://howtotrainyourdragon.fandom.com/wiki/Toothless_(Franchise))
([https://howtotrainyourdragon.fandom.com/wiki/Toothless_\(Franchise\)](https://howtotrainyourdragon.fandom.com/wiki/Toothless_(Franchise)))

Spike (Figure 3), from My Little Pony, is a purple and green dragon. He is one of Twilight Sparkle's friend and assistant.



Spike from My Little Pony. Credit to: [https://dragons.fandom.com/wiki/Spike_\(My_Little_Pony\)](https://dragons.fandom.com/wiki/Spike_(My_Little_Pony))
([https://dragons.fandom.com/wiki/Spike_\(My_Little_Pony\)](https://dragons.fandom.com/wiki/Spike_(My_Little_Pony)))

Dragonite (Figure 4), from Pokemon Go, is a orange dragon. He is the final form of Dratini.



Dragonite in Pokemon Go. Credit to: <https://fantendo.fandom.com/wiki/Dragonite>
(<https://fantendo.fandom.com/wiki/Dragonite>)

According to Figure 1, Toothless and Spike are closely related and have the same ancestor.