Project 2 Notebook

Introduction

Scientific Question

When examining dog breeds (Canis lupus familiars), will breeds of a similar size (e.g. Cocker Spaniel, English Cocker Spaniel) have more related genes (IGF1, IGSF1, LCORL, and SMAD2) and SNPs surrounding longevity than breeds of a different size (e.g.Doberman Pinscher,Miniature Pinscher)? Will these differences results in expression changes between breeds of different sizes?

Note: I selected 4 genes closely associated with life span (IGF1, IGSF1, LCORL,and SMAD2). There are more genes involved than this. Breed size will be determined by the American Kennel Club (AKC). You can filter by all AKC recognized dog breeds by size on their website: https://www.akc.org/dog-breeds/.

Species of interest: Canis lupus familiaris

Canis lupus familiars or the domestic dog, is the most phenotypically variable mammal on earth. This phenotypic variation is due to the intense selection that humans have imposed on dogs for the thousands of years we have been coexisting. But this wide phenotypic variation does more than make dogs cuter or more usable for humans. Variation in dog size in particular has a huge effect on the longevity of the dog.

Gene of interest: LCORL

LCORL or Ligand Dependent Nuclear Receptor Corepressor Like. This gene encodes a transcription factor which functions in spermatogenesis. Polymorphisms of this gene are associated with changes in skeletal frame size and height. (https://www.genecards.org/cgi-bin/carddisp.pl?gene=LCORL)

Gene of interest: IGF1

IGF1 or Insulin Like Growth Factor 1. This gene encodes a protein similar to insulin (functionally and structurally) and is involved in mediating growth and development. (https://www.genecards.org/cgi-bin/carddisp.pl?gene=IGF1&keywords=IGF1)

Gene of interest: IGSF1

IGSF1 or Immunoglobulin Super family Member 1. This gene encodes a member of the immunoglobulin-like domain-containing super family. It contains various immunoglobulin-like domains thought to regulate interactions between cells. (https://www.genecards.org/cgi-bin/carddisp.pl?gene=IGSF1&keywords=IGsF1)

Gene of interest: SMAD2

SMAD2 or SMAD Family Member 2. This gene encodes a protein that mediates signaling pathways. This includes the signal of the transforming growth factor (TGF)-beta. So SMAD2 indirectly regulates multiple cellular processes, such as cell proliferation, apoptosis, and differentiation. (https://www.genecards.org/cgi-bin/carddisp.pl?gene=SMAD2&keywords=SMAD2)

Scientific Hypothesis

If you examine canine breeds, then breeds of a similar size (e.g.Cocker Spaniel,English Cocker Spaniel) they will have more related SNPs and/or fragments of genes surrounding longevity (e.g., IGF1, IGSF1, LCORL, and SMAD2) than breeds of a different size (e.g., Doberman Pinscher,Miniature Pinscher). Changes in these SNPs and gene fragments will results in expression changes between breeds.

Analysis Performed:

SNP analysis

- Exploratory Data Analysis (EDA): Pie charts were created to show surface-level nucleotide distributions before size is taken into account.
- Multiple Sequence Alignment: multiple sequence alignment (MSA) was performed on the SNPs and bordering sequences (+/- 5 bp). The results of these alignments was displayed with msaPrettyPrint()
- Clustering of MSA results, which were visualized as dendrograms

Gene analysis:

- Multiple Sequence Alignment–>Clustering: multiple sequence alignment (MSA) was performed on the genes. These alignments were converted into a data format feasible to calculate a distance matrix from. This distance matrix was used to cluster the data which was then visualized as a dendrogram.

Gene expression data:

- EDA: Scatter plots with regression lines were utilized see if expression, not changes in SNPs are the reason behind differences in longevity based on dog size.
- Clustering of expression data, for genes studied in the prior two analysis steps. This data was kept on one excel sheet so the clustering of expression data by dog size could be represented in one dendrogram.

Data Sourcing

- Dog breed information (visualization and sizing): American Kennel Club (https://www.akc.org/)
- Data downloads:
- SNP and gene positions for MSA and clustering: The ideal positions to perform our MSA and clustering at were determined by table 2 in Plassais' 2019 paper on whole genome sequence analysis on canines (https://www.nature.com/articles/s41467-019-09373-w/tables/2#ref-CR35). this table identifies the regions for each gene which account for the most variation. In the cases where the noted region was larger than the gene itself, I instead used the gene coordinates given by NCBI. We now have coordinated to perform our MSA alignment on, but no sequences. How to obtain these sequences is explained below.
- NCBI dog reference genomes for MSA and clustering: First search up your gene of interest + "dog" (Example search for IGSF1 https://www.ncbi.nlm.nih.gov/gene/?term=IGSF1+dog). Select the relevant result. Under "Genomic regions, transcripts, and products" will be the 6 possible reference genomes. Select the reference genome of interest from the drop down menu and then select "FASTA" from the "Go to nucleotide" menu. Once you are viewing the reference sequence, change "Change region shown" to "Selected region" and enter your coordinates of choice. "Update view" must then be selected for this

change to take place. Once the proper region is visualized, you can download this sequence to your local machine with "Send to", "complete record", and your file extension of choice (I used .fasta).

- Gene expression data for clustering: This gene expression data was extracted from the "Source Data" file from Plassais' 2019 paper on whole genome sequence analysis on canines (https://www.nature.com/articles/s41467-019-09373-w/tables/2#ref-CR35). The expression is on the S4-S6 sheet. I removed the genes that I did not examine during prior steps. I then shifted the organization of the to better work for clustering (eg made the dog breed the row names and added expression results as column data making a data frame with fewer rows and more columns), altering no numbers in the process. Since the breed names were listed without size data, I manually added in each breeds AKC size designation. This labeling makes interpreting our dendrograms much simpler down the line.

Package definitions

Before running our code numerous packages must be loaded in, below is a short summary of their general purpose and their purpose in the scope of my Project 2

1) BiocManager: A tool to install and update Bioconductor packages. For the purposes of my project, BiocManager is used to install msa and DECIPHER.
2) readxl: A package to import excel files into R. For the purposes of my project, readxl is used to read in excel data with only one sheet. Installation: install.packages("readxl")
3) xlsx: A package to import excel files into R. For the purposes of my project, xlsx is used to read in multi-sheet excel data with row-names specifications. Installation: install.packages("xlsx")
4) msa: A package containing an interface for three multiple sequence alignment algorithms (CLUSTALW, CLUSTALOmega, and MUSCLE). For the purposes of my project, msa is used to perform multiple sequence alignment on our genes and SNPs of interest and visualization of these resulting alignments. Installation (must also have BiocManager installed to do this):

   if (!requireNamespace("BiocManager", quietly=TRUE)) install.packages("BiocManager") BiocManager::install("msa")
5) tinytex: A custom La TeX distribution for R. For the purposes of my project, tinytex is used for my msaprettyprint output. Installation: install.packages('tinytex')
6) ggplot2: A system to create graphics declarative or piece-by-piece, this allows for a great amount of customization. For the purposes of my project, ggplot is used to visualize a lot of my exploratory data analysis (breakdown of dog populations and expression data). Installation: install.packages('ggplot2')
7) DECIPHER:. For the purposes of my project, DECIPHER is used to cluster DNA sequences and form their resulting dendrograms. Installation (must also have BiocManager installed to do this):

   if (!requireNamespace("BiocManager", quietly = TRUE)) install.packages("BiocManager") BiocManager::install("DECIPHER")
8) dendextend: A package which extends the functionality of dendrogram objects in R through tree vs. tree comparisons and extensive graphical customization of dendrograms. For the purposes of my project dendextend is used to fine tune my dendrograms (color by cluster group, prevent breed names being cut off, control branch height, adding titles). Installation: install.packages('dendextend')
9) seqinr: A package which performs EDA and data visualization for biological sequence data. For the purposes of my project, I will use seqinr to convert msa alignments to seqinr alignments, allowing me to create distance matrices from them. Installation: install.packages('seqinr')
10) ape: provides functions for reading, writing, manipulating, analyzing, and simulating phylogenetic trees and DNA sequences. For the purposes of my project ape will help convert msa alignments to seqinr alignments, allowing me to create distance matrices from them. Installation: install.packages('ape')

```r
#for reading in fasta files
library("BiocManager")
#for reading in single sheet excel files
library("readxl")
#for reading in multi-sheet excel files
library("xlsx")
#for multiple sequence alignment
library("msa")
```

```
## Loading required package: Biostrings
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min
```

```
## Loading required package: S4Vectors
```

```
## Loading required package: stats4
```

```
##
## Attaching package: 'S4Vectors'
```

```
## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname
```

```
## Loading required package: IRanges
```

```
## Loading required package: XVector
```

```
## Loading required package: GenomeInfoDb
```

```
##
## Attaching package: 'Biostrings'
```

```
## The following object is masked from 'package:base':
##
##     strsplit
```

```
##
## Attaching package: 'msa'
```

```
## The following object is masked from 'package:BiocManager':
##
##     version
```

```r
#for msa pretty print
library("tinytex")
#visualization of results
library("ggplot2")
#for clustering of DNA sequences
library("DECIPHER")
```

```
## Loading required package: RSQLite
```

```
## Loading required package: parallel
```

```r
#for cleaning up dendrograms
library('dendextend')
```

```
##
## ---------------------
## Welcome to dendextend version 1.15.2
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attaching package: 'dendextend'
```

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

```
## The following object is masked from 'package:stats':
##
##     cutree
```

```r
#for converting msa alignments to seqinr-msa alignments
library("seqinr")
```

```
##
## Attaching package: 'seqinr'
```

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

```r
#for converting msa alignments to seqinr-msa alignments
library("ape")
```

```
##
## Attaching package: 'ape'
```

```
## The following objects are masked from 'package:seqinr':
##
##     as.alignment, consensus
## The following objects are masked from 'package:dendextend':
##
##     ladderize, rotate
```

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

First we will be performing MSA and clustering with various sequence data (SNPS and gene fragments). Unfortunately, the scope of this data is limited to the 5 breeds shown below, and heavily weighs our analysis towards larger dogs.

Basenji
(S)

Labrador
Retriever
(L)

Boxer
(L)

Ger
Shep
(

Great Dane
(XL)

Function definitions

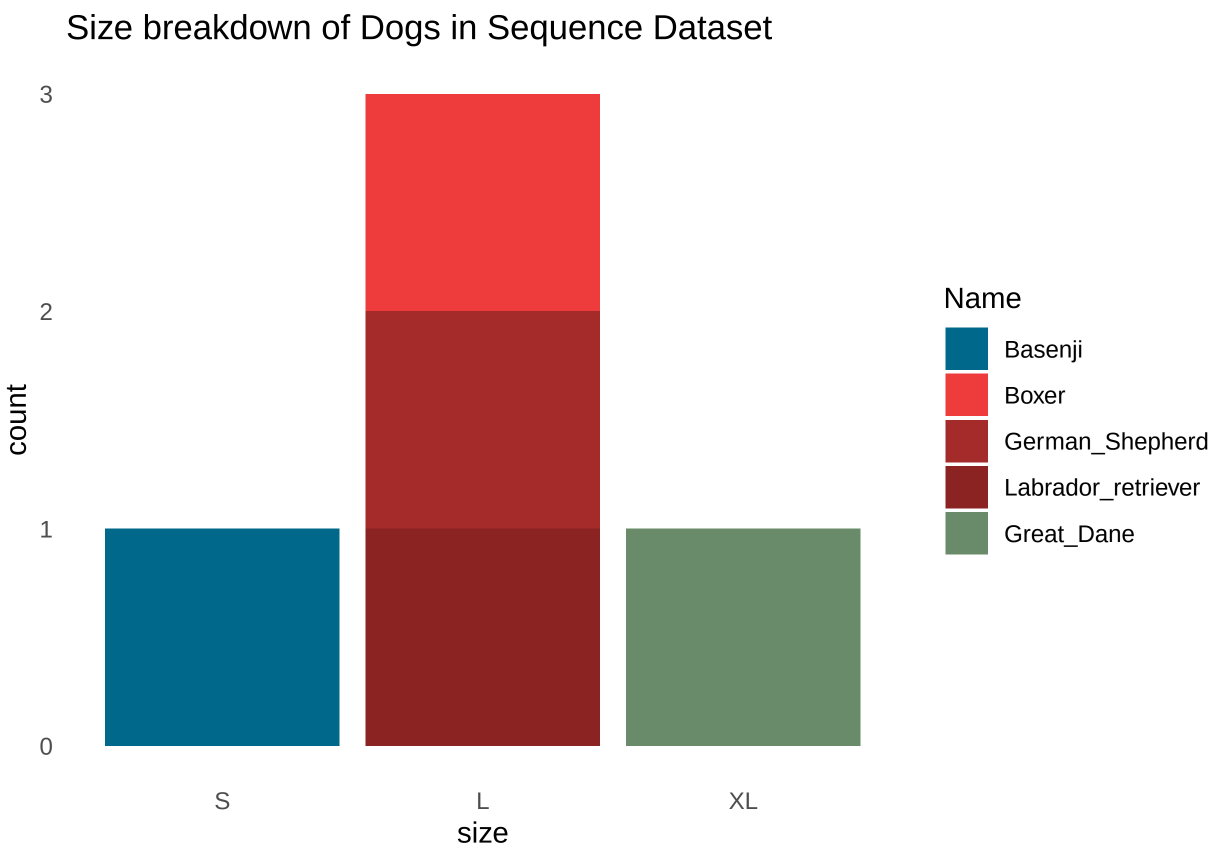
```
#global variable, string used to read in msa alignment PDFs
alignment_name<<-""
#global variable, vector of color names (strings) used to color my future ggplots
myColors <- c("antiquewhite4", " coral3", "darkgoldenrod4","darkcyan", "darkorchid4")

#mult_alignments: align fasta depending on inputs, returns msaprettyprint alignment or plain text align
#file_name: fasta file to be read in
#fasta_names: names to display for prettyprint alignment, "names" in file_name files are entire paragra
#big_aln: determines if msaprettyprint is run (True=run and return pdf name for display by knitr, False
#DNA_set: determines how the fasta data is read in (True=DNA, False=Amino Acid) this speeds up alignmen
mult_alignments<-function(file_name,fasta_names,name,big_aln=FALSE,dna_set=TRUE){

  #read in fasta for all dogs
  #use DNA for small files
  if(dna_set=="TRUE"){
  #string_set: local variable that holds our fasta data as DNAStringSets.
  #The fasta data is the sequence data for the 6 reference sequence breeds
  # in our gene or SNP of interest
  string_set<-readDNAStringSet(file=file_name,use.names=FALSE)
  }
  #AA for large files
  else{
      #string_set: local variable that holds our fasta data as AAStringSets
      string_set<-readAAStringSet(file=file_name,use.names=FALSE)

  }

  #update names for pretty print display from  list which contains our shortened
  #sequences names read in from fasta_names, a .text file with no headers and
  #variables separated by new lines
  names(string_set)<-read.table(fasta_names, header = FALSE, sep = "\n")[["V1"]]

  #alignment: local variable which stores the msa of our aligned, named sequences,
  #we want to maintain our input order, which is ordered by breed size.
  alignment<-msa(string_set,order="input")
  #if seq cant be displayed with msa pretty print, return
  if(big_aln==TRUE){
    return(alignment)
  }
  #update global variable so multiple pretty print runs don't overrun each
  #other. Combine the strings of the name of the gene inputted with the file
  #type, ".pdf". The gsub call ensures there's no unwanted spaces between our
  #gene name and the file extension
  alignment_name<<-gsub(" ", "", paste(name,".pdf"), fixed = TRUE)

#return pretty alignment, as a pdf. file saves our results to the desired,
#named pdf. output="pdf": determines our file extension, showNames="right":
#places our sequence names (dog breed + size) to the right of the aligned
#sequences, showLogo="top": places a DNA logo on the top of our alignment,
#askForOverwrite=False overwrites old files with the same name without warnings,
#showNumbering =none removes unnecessary numerbing from alignment. paperWidth
# and paperHeight control output sizing
  msaPrettyPrint(alignment, file=alignment_name,output="pdf",
                 showNames="right",showLogo="top",askForOverwrite=FALSE,
                 showNumbering="none",paperWidth=6,paperHeight=3)
#return location of pdf
return(alignment_name)
}
#create figure with white background, no grid line and only ticks along x and y
#axis. Designed for bar plot displays but should work with many other forms of
#ggplots. fig: ggplot to be altered, is directly returned
tune_figure<-function(fig){
  #builds changes based off of theme_minimal().
  return(fig+theme_minimal()+theme(
    #removes background color
    plot.background = element_blank(),
    #removes gridding on plot
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    #removes x and y axis lines, keeping the ticks in place.
    panel.border = element_blank()))
}
#create dendrogram based on fasta files, names of items clustered in fasta_names
# plots the dendrogram and returns the object in case further manipulation is
# desired
#fasta_path: path to fasta file
#fasta_names: path to names to display on dendrogram
create_dendrogram<-function(fasta_path, fasta_names, fig_title){
  #grab DNA info from collated file
  #DNA is a local variable that contains the read in fasta file as a DNAStringSet.
  #The names are not read in due to their excessive length
  dna <- string_set<-readDNAStringSet(file=fasta_path,use.names=FALSE)

  #update DNAStringSet sequences to have the shorter names read in from the
  #file, fasta_names, read in as a list
  names(dna)=read.table(fasta_names, header = FALSE, sep = "\n")[["V1"]]

  #create local variable d1 which forms a distance matrix for clustering
  #based off our named DNA sequence data
  d1 <- DistanceMatrix(dna, type="dist")

  #form local dendrogram object, dendrogram with IdClusters the DNA specific
  #clustering package
  dendogram<-IdClusters(d1, method="complete", cutoff=0.05, showPlot=FALSE,
                        type="dendrogram")
  #local variable used in plot call to fix names being cut-off
  nodePar <- list(lab.cex = 0.6, pch = c(NA, 19),
                cex = 0.7, col = "black")
#plot results utilizing passed figure title
plot(as.dendrogram(dendogram), ylab = "Height", nodePar =nodePar,main=fig_title)

#return as.dendrogram
  return(as.dendrogram(dendogram))
}
```

```
#plot_expression(): plots the expression data from excel_name, on sheet_name as
# a ggplot scatter plot. Sets the x-axis and y-axis data with x and y.
# Colors points according to col (column in data frame). Labels the x-axis and
# y-axis with xlab and ylab. Sets the figure title to be title
plot_expression<-function(excel_name,sheet_name,x,y,col, xlab, ylab,title){

  #read in the sheet_name expression data as the local dataframe expression
  expression<-read_excel(excel_name,sheet=sheet_name)

  #plot the scatterplot as the local variable p
  #ggplot(): sets our data to be expression, our y data to be y (the expression of our gene of choice),
  #geom_point(): customizes our point sizes and opacity
  #labs(): sets the x and y axis labels to be xlab and ylab
  #ggtitle(): sets the figure title with title
  #scale_color_manual(): sets the color of the points to global variable myColors
  #geom_smooth(): adds regression line to scatter plot
  p<-ggplot(data = expression, mapping = aes_string(y=y,x=x ,col= col))+geom_point(size=4,alpha=0.6)+ la
    ,aes(group=1),color="black",alpha=0.6,se=TRUE)
#return ggplot object (has not been displayed yet)
return(p)
}
```

EDA of Sequence data

Before performing Bioinformatics method on our sequencing data, we can perform some Exploratory Data Analysis (EDA) too see if any preliminary tends are visible, first we will examine the breeds which will be studied through this sequence data.

```
#visualize size breakdown of dogs (snps and p are both global variables)

#read in dog snps.xlsx, an excel sheet created by me which contain the demographic breakdowns of this s
snps<-read_excel("dog snps.xlsx")

#Fix ordering of legend to follow dogs ordered from left to right, top to bottom
#instead of alphabetically
snps$Name <- factor(snps$Name, levels = c("Basenji", "Boxer", "German_Shepherd","Labrador_retriever","G

#form barplot which  depicts demographics breakdown of files
#ggplot() sets snps as the dataframe and size as the x-axis
#scale_x_discrete Set numeric limits of size as size-strings for visualization purposes
#geom_bar sets Name as the y-axis data
#scale_fill_manual: sets color of individual dog breeds in bar plot
#ggtitle: adds title to figure
p<-ggplot(data = snps, aes(size))+scale_x_discrete(limits = c("S","L","XL"))+geom_bar(aes(fill = Name))
#runs tune_figure to standardize presentation
tune_figure(p)
```



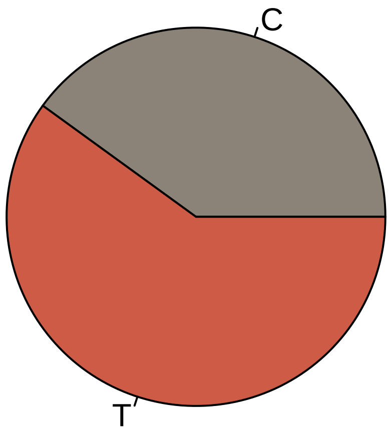
Our sequence data is heavily biased towards larger breeds. 4 of the 5 breeds examined were large or extra large breeds. There was only 1 breed of small dog, and no extra small or medium dogs examined.

LCORL Sequence Analysis

Moving into the analysis of the first gene, we wish to study, lets look at the nucleotide breakdown for LCORL. Note that although we are aligning the LCOR snp +/- 5 bp, for this visualization we will only examine the SNP itself

```
#read in dog snps.xlsx sheet pie, an excel sheet created by me which contains
#the nucleotide breakdowns of this SNP as the global dataframe pies
pies<-read_excel("dog snps.xlsx",sheet="pie")

#display prepared dataframe data for lcorl. Adding the table(pies$lcorl) removes
#the error "'x' values must be positive." col=MyColors sets out categorical
#data to a predetermined color vector, this allows consistency of presentation
#across our EDA graphs. main sets the title
pie(table(pies$lcorl),col=myColors,main="LCORL SNP Breakdown")
```

LCORL SNP Breakdown



In our dog population, only the C and T allele are seen with the T allele having a slightly higher frequency.

Now lets move to multiple sequence alignment of our LCORL SNP, here we will use the previously explained function mult_alignments to perform MSA and save our results to a pdf. We will print the global variable alignment to show how it was modified through this function.

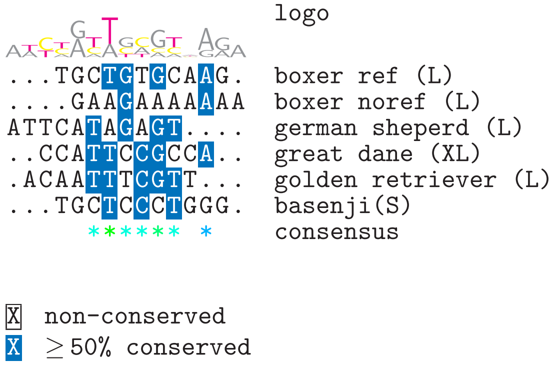
```
#LCORL call multiple sequence alignment helper function. Since LCORL_file.txt is
#a smaller file, do save a msaprettyprint output."LCORL_file.txt": contains our
#fasta data for each dog breeds SNP +/- 5 bp. "fasta/names.txt": contains the
#names of our sequence alignments to be used during msaprettyprint. "LCORL"
#determines the name of the saved prettyprint call to be LCORL.pdf
alignment<-mult_alignments("fasta/LCORL_file.txt","fasta/names.txt","LCORL")
```

```
## use default substitution matrix
```

```
#show updated global variable
print(alignment_name)
```

```
## [1] "LCORL.pdf"
```

Ok, now that we have performed MSA, lets read back in our sequence results for visualization purposes with knitr::include_graphics. We will use the updated global variable alignment_name to accomplish this



```
                  logo
...TGCTGTGCAAG.   boxer ref (L)
....GAAGAAAAAAA   boxer noref (L)
ATTCATAGAGT....   german sheperd (L)
..CCATTCCGCCA..   great dane (XL)
.ACAATTTCGTT...   golden retriever (L)
...TGCTCCCTGGG.   basenji(S)
     ****** *     consensus
```

☒ non-conserved

🟦 ≥ 50% conserved

These alignments appear to be largely unrealated, it looks like msa introduces 4 indels into each sequence to get the center pieces to align. So what is being aligned in the middle of the sequences is often not the consensus sequence. There are several possible reasons for these results:

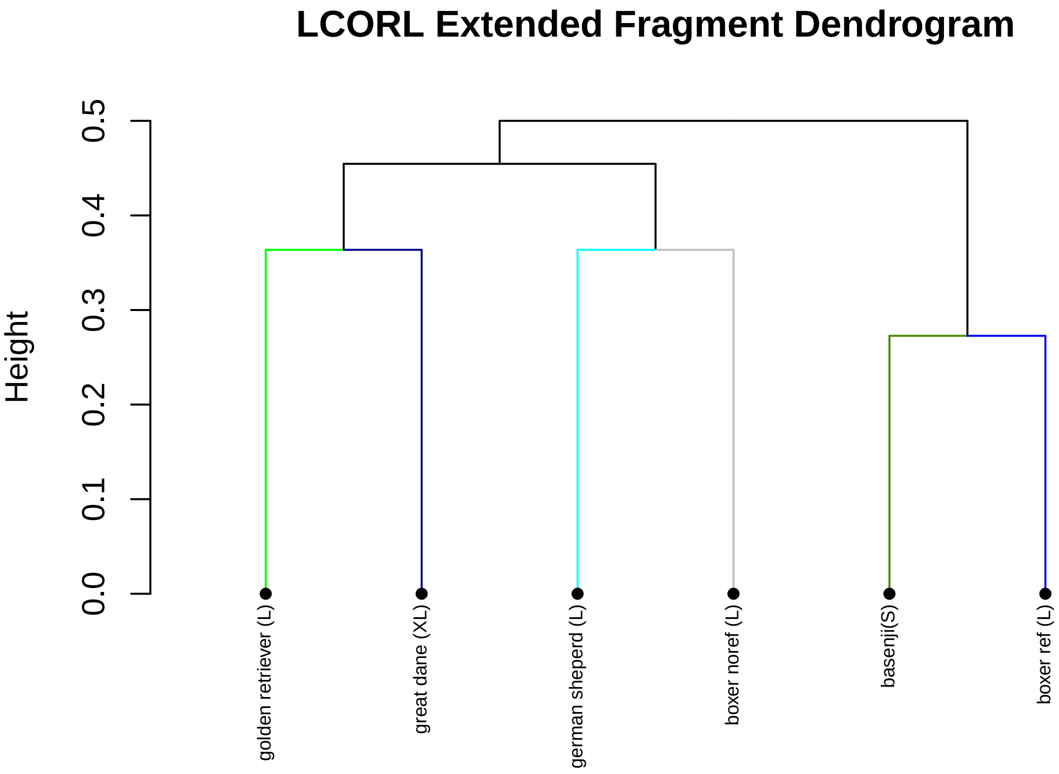
- Even though this region was found to have the most variation among dog breeds of different sizes, this SNP, by nature, is not large enough to make an impact on clustering data. This can especially be by how msa shifts the SNPs off to the left of right to align together more of the trailing basepairs.
- The fasta data was misindexed. Since the provided positions do not notate the reference dog genome, I indexed the reference genomes on NCBI with the same position throughout. Thus the regions I chose to align and cluster may not be the correct region for all 6 breeds.

LCORL Clustering

Now we can cluster our LCORL data with the previously explained create_dendrogram. The leaves in this case will be the dog breed name alongside their AKC designated size

```
#Cluster LCORL extended fragment
#call helper function create_dendrogram. LCORL_file.txt is the location of our
#fasta file containing the LCORL SNP +/- 5 bp for our our 6 dogs. names.txt
#contains the names of these fasta sequences to be used for the visualization
#of the dendrogram. "LCORL Extended Fragment Dendrogram" is the figure title
create_dendrogram("fasta/LCORL_file.txt", "fasta/names.txt", "LCORL Extended Fragment Dendrogram")
```

```
## ================================================================================
##
## Time difference of 0 secs
##
## ================================================================================
##
## Time difference of 0.01 secs
```



```
## 'dendrogram' with 2 branches and 6 members total, at height 0.5
```

The clustering by dog breed in this case appears to be inconclusive While the Golden Retriever (L), Great Dane (XL), German Shepherd (L), and boxer no ref (large) are all more closely related to each other than the Basenji (S) this does not explain the Basenji's close relationship with the Boxer ref sequence. There are several possible explanations for these results:

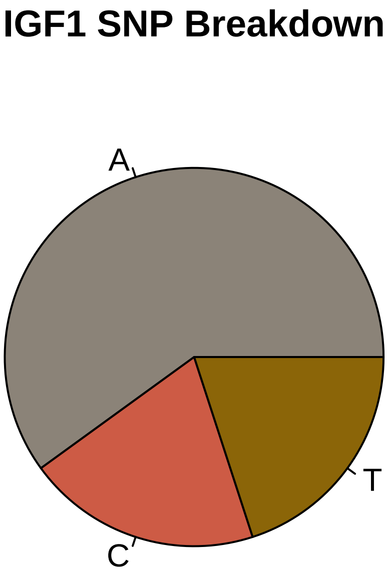
- Even though this region was found to have the most variation among dog breeds of different sizes, this SNP, by nature, is not large enough to make an impact on clustering data.
- The population we clustered is relatively homogeneous made up of primarily large dogs, this is further emphasized by the occurrence of only two LCORL alleles, as depicted in the LCORL pie chart. If the 5 bp regions around this SNP are similar across breeds then this double branching configuration makes more sense.
- The fasta data was misindexed. Since the provided positions do not notate the reference dog genome, I indexed the reference genomes on NCBI with the same position throughout. Thus the regions I chose to align and cluster may not be the correct region for all 6 breeds.

IGF1 Sequence Analysis

EDA: Moving into the analysis of the next gene we wish to study, IGF1. lets look at the nucleotide breakdown for IGF1 Note that although we are aligning the IGF1 snp +/- 5 bp, for this visualization we will only examine the SNP itself

```
#read in dog snps.xlsx sheet pie, an excel sheet created by me which contains
#the nucleotide breakdowns of this SNP as the global dataframe pies
pies<-read_excel("dog snps.xlsx",sheet="pie")

#display prepared dataframe data for IGF1 Adding the table(pies$igf1) removes
#the error "'x' values must be positive." col=MyColors sets out categorical
```

```
#data to a predetermined color vector, this allows consistency of presentation
#across our EDA graphs
pie(table(pies$igf1),col=myColors,main="IGF1 SNP Breakdown")
```



There is more variation in the IGF1 SNP data than the LCORL data. The majority of individuals had the A allele, but there was also a C and T sub population.

IGF1 Sequence Alignment

Now lets move to multiple sequence alignment of our IGF1 SNP, here we will use the previously explained function mult_alignments to perform MSA and save our results to a pdf.

```
#igf1 call multiple sequence alignment helper function. Since igf1.fasta is a
# smaller file, do save a msaprettyprint output. "LCORL_file.txt": contains our
#fasta data for each dog breeds SNP +/- 5 bp. "fasta/igf1_names.txt": contains
#the names of our sequence alignments to be used during msaprettyprint. "igf1"
#determines the name of the saved prettyprint call to be igf1.pdf
alignment<-mult_alignments("fasta/igf1.fasta","fasta/igf1_names.txt","igf1")
```

```
## use default substitution matrix
```

Ok, now that we have performed MSA, lets read back in our sequence results for visualization purposes with knitr::include_graphics

```
                  logo
TTGCTTTTGTA....   Boxer no ref (L)
.GGGCCCGGCTG...   Boxer ref (L)
....TCTGAAGAGTA   German shepherd (L)
..GGCATTCCCCT..   Basenji (S)
AGGTCATGACT....   Great dane (XL)
..AATTCAGTGAA..   Labrador retriever (L)
  *   ****        consensus
```

X non-conserved

X ≥ 50% conserved

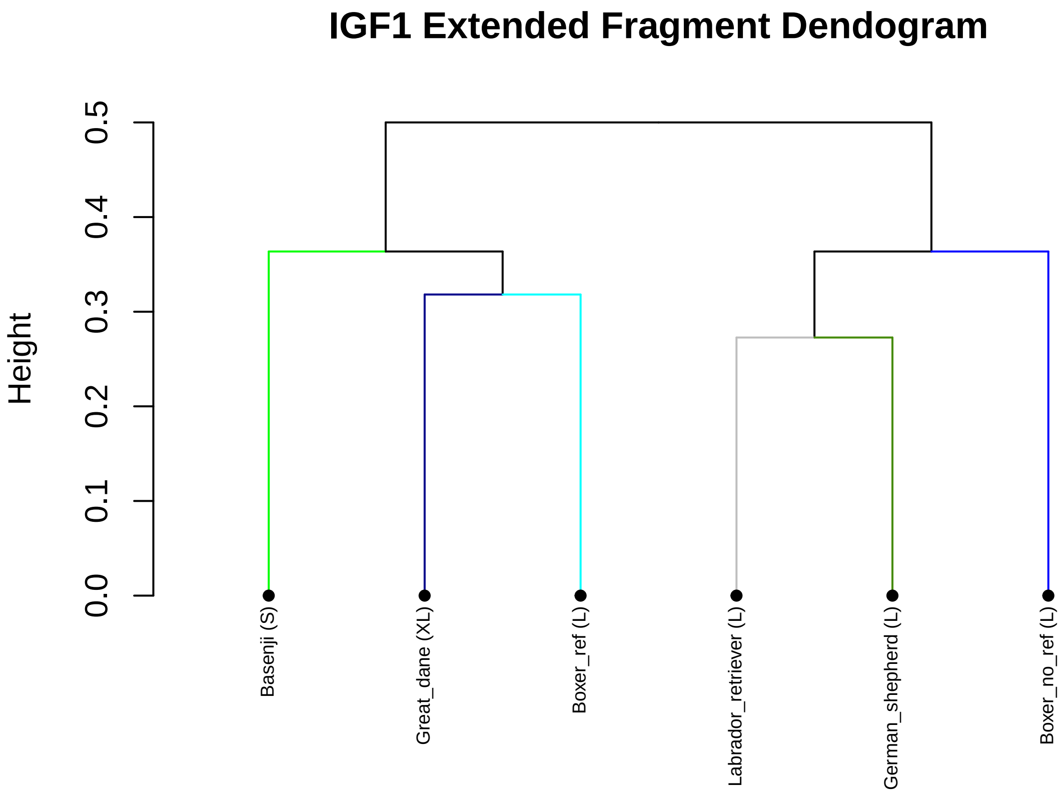
The resulting MSAs appear to be largely unrelated. Just as they were in the case of LCORL, msa introduces 4 indels into each sequence to get the center pieces to align. So what is being aligned in the middle of the sequences is often not the consensus sequence. There are several possible reasons for these results are the same as the reasons listed under the LCORL alignment, as the issue here with the data is mechanical, not related to gene function or dog sizing in particular.

IGF1 Clustering

Now we can cluster our IGF1 data with the previously explained create_dendrogram. The leaves in this case will be the dog breed name alongside their AKC designated size

```r
#Cluster IGF1 extended fragment
#call create_dendrogram. igf1.fasta is the location of our fasta file containing
#the LCORL snp +/- 5 bp for our our 6 dogs. igf1_names.txt contains the names of
#these fasta sequences to be used for the visualization of the dendrogram.
#"IGF1 Extended Fragment Dendrogram" is the figure title
create_dendrogram("fasta/igf1.fasta", "fasta/igf1_names.txt", "IGF1 Extended Fragment Dendogram")
```

```
## ================================================================================
##
## Time difference of 0 secs
##
## ================================================================================
##
## Time difference of 0 secs
```



```
## 'dendrogram' with 2 branches and 6 members total, at height 0.5
```

The clustering by dog breed in this case appears to be inconclusive While the Labrador Retriever (L), German Shepherd (L), and Boxer_no_ref (L) are all more closely related to each other than the Basenji (S) this does not explain the Basenji's closer relationship with the Great Dane (XL) and Boxer ref (L). It is also odd that the two boxer references cluster so far from one another. There are several possible explanations for these results:

- Even though this region was found to have the most variation among dog breeds of different sizes, this SNP, by nature, is not large enough to make an impact on clustering data.
- The Great Dane (XL) and Boxer_ref (L) are not homozygous for the IGF1 SNP. This could allow for them to cluster with the Basenji on the reference data but still have the large-size phenotype from the other member of their haplotype.
- The fasta data was misindexed. Since the provided positions do not notate the reference dog genome, I indexed the reference genomes on NCBI with the same position throughout. Thus the regions I chose to align and cluster may not be the correct region for all 6 breeds.

IGSF1 Sequence Alignment

Now we will move to analyzing IGSF1, where we will be utilizing the whole gene sequence in our analysis. This changes our data pipeline in comparison to IGF1 and LCORL.

First utilize the mult_alignments function, but this time we set big_aln=TRUE which returns the alignment result without visualizing it (as the alignment exceeds msaprettyprint's character limit). We will also set dna_set=FALSE for then our data will align as an AA sequence. This greatly shortens the alignment run time.

```
#igsf1 call multiple sequence alignment helper function. Since IGSF1.fasta is a
#larger file, do not save a msaprettyprint output. "IGSF1.fasta": contains our
```

```
#fasta data for each dog breeds SNP +/- 5 bp. "fasta/IGSF1_names.txt": contains
#the names of our sequence alignments to be used during msaprettyprint. "IGSF1"
#determines the name of the saved prettyprint call to be IGSF1.pdf. big_aln=TRUE
#returns the alignment result without visualizing it and dna_set=FALSE for then
#our data will align as an AA sequence. This greatly shortens run time.
igsf1<-mult_alignments("fasta/IGSF1.fasta","fasta/IGSF1_names.txt","IGSF1",TRUE,FALSE)
```

```
## use default substitution matrix
```

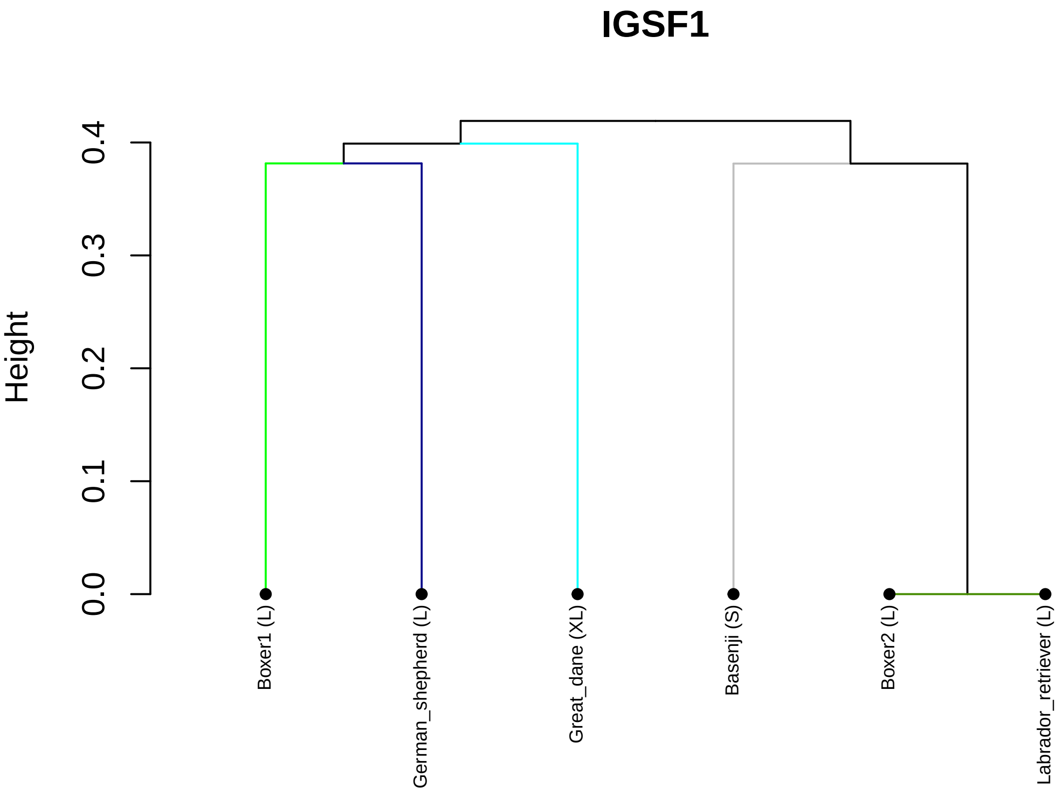
IGSF1 Clustering

Now that we have a multiple alignment we can use msaconvert() to shift this msa alignment into a seqinr alignment. This seqinr alignment will allow us to form a distance matrix of our data which we can then utilize IdClusters on, just like our previous SNP clusters did.

```
#unlike the dendrogram formation for the snps, igsf1_aln, d, dendogram, and,
#nodePar are all global variables

#convert our msa alignment to a seqinr::alignment (of type chr "alignment")
igsf1_aln <- msaConvert(igsf1, type="seqinr::alignment")

#form a distance matrix from this new information
d <- dist.alignment(igsf1_aln, "identity")

#create a dendrogram from the same function used for SNP clustering with the same defaults
dendogram<-IdClusters(d, method="complete", cutoff=0.05, showPlot=FALSE,type="dendrogram")
```

```
## ================================================================================
##
## Time difference of 0 secs
```

```
  # list with plot specifications that fix names being cut-off in dendrogram
  nodePar <- list(lab.cex = 0.6, pch = c(NA, 19),
                cex = 0.7, col = "black")
#plot results
plot(as.dendrogram(dendogram), ylab = "Height", nodePar =nodePar,main="IGSF1")
```



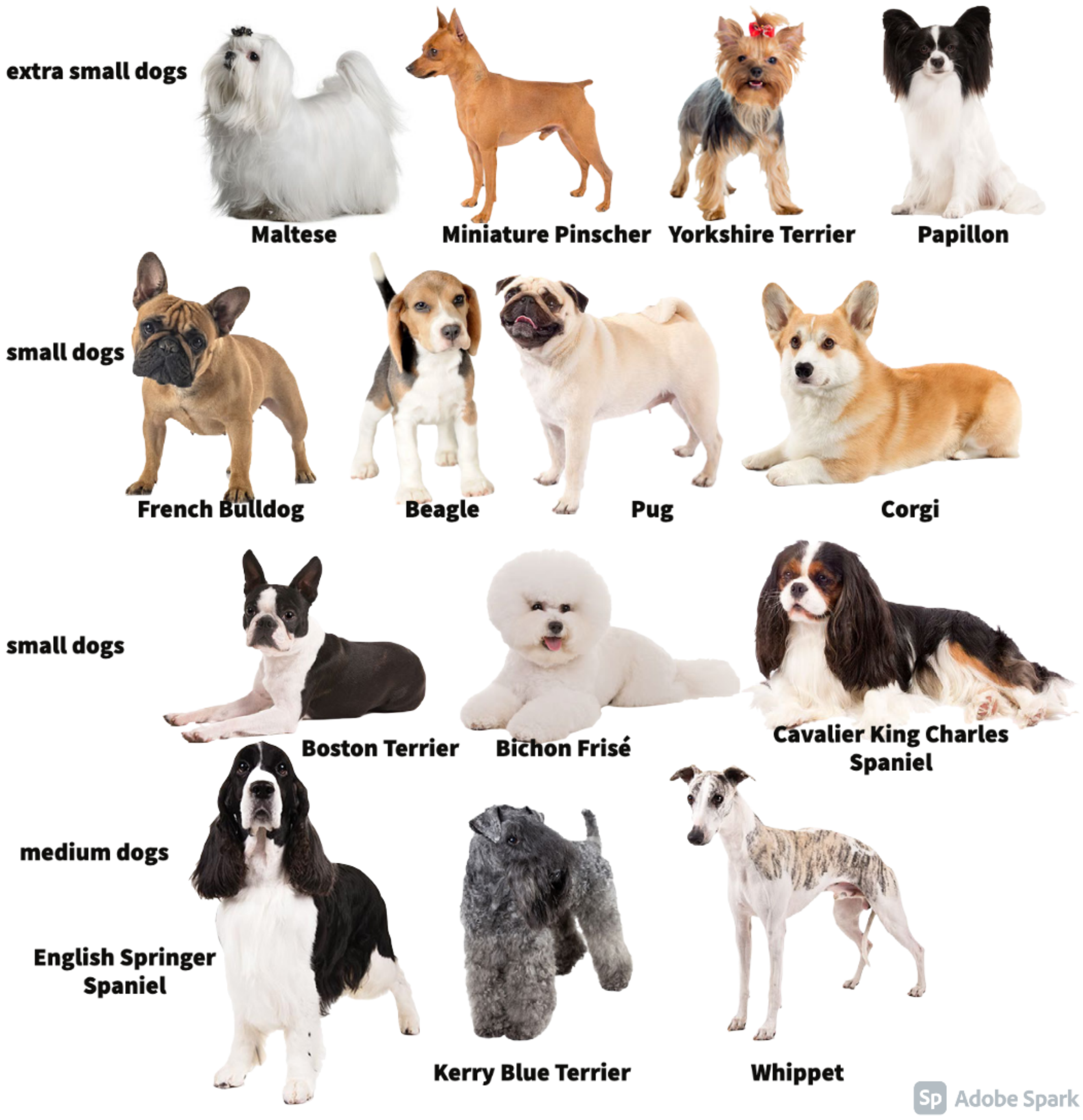
This is our most promising dendrogram. There are two branches with two large dogs as their leaves (Boxer1, German Shepherd and Boxer2, Labrador Retriever). The dogs that have a different size classification (Great Dane, Basenji) are more distant ancestors to this large dog branches. The only inexplicable aspect of this data is how distantly related the two large dog branches are denoted as. A Possibility for this is that there are two distinct homozygous sequences of IGF1 that both produce large dogs. In the case of heterozygotes, the dog could instead be smaller or larger than the homozygous configuration.

Note on SMAD2: SMAD2 was too large to perform the same analysis pipeline I used with the other larger sequence, IGSF1 (it consistently crashed R during alignment). It will still be examined in our gene expression analysis.

Gene Expression Analysis

Note on IGSF1: IGSF is not present in the data set that I based this analyis pipeline off of, so it will not be included in any of the analysis in this section

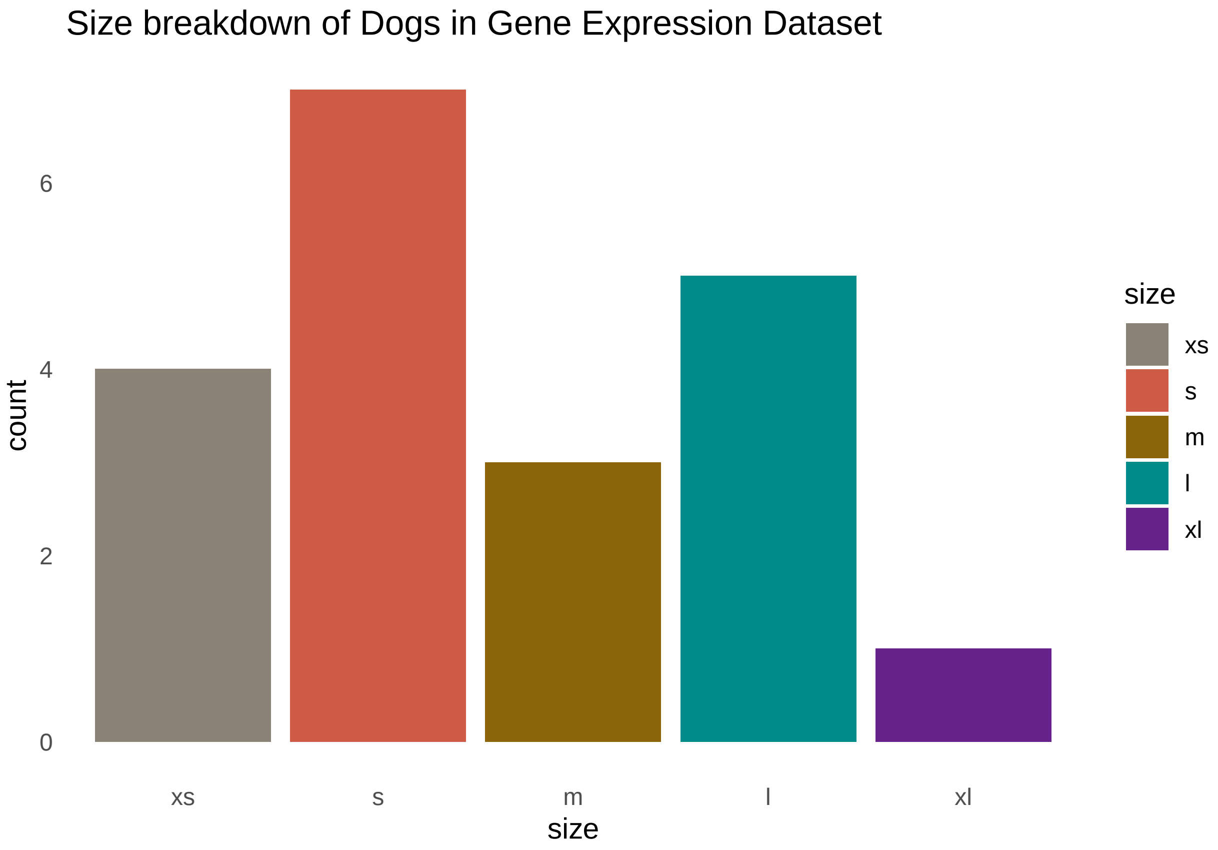
Through both alignment and clustering, differences between dogs sequence-wise seem to be insignificant when examining such a small subsection of dogs. Thankfully there is also expression data for the genes studied which covers more dog-breeds, depicted below.


extra small dogs
Maltese
Miniature Pinscher
Yorkshire Terrier
Papillon
small dogs
French Bulldog
Beagle
Pug
Corgi
small dogs
Boston Terrier
Bichon Frisé
Cavalier King Charles Spaniel
medium dogs
English Springer Spaniel
Kerry Blue Terrier
Whippet
Sp Adobe Spark

EDA of Dataset

Before performing clustering on our gene expression data, we can perform some Exploratory Data Analysis (EDA) to see if any preliminary tends are visible, first we will examine the breeds studied through this sequence data.

```
#visualize size breakdown of dogs, IGF1 contains all of the breeds'
#average IGF1, but we will only focused on the breeds examined here
#Is a global dataframe which contains the dog breed information relevant for our EDA
expression<-read_excel("dog snps.xlsx",sheet="IGF1")

#fix ordering of bars from being ordered alphabetically to being ordered by size
expression$size <- factor(expression$size, levels = c("xs","s","m","l","xl"))

#create a bar plot with this freshly ordered dog breed data to the global
#variable, p
#ggplot() sets dataframe as expression and size as the x-axis
#geom_bar() sets counts of size data to be the y-value
#scale_fill_manual: sets color of size categories in barplot
p<-ggplot(data = expression, aes(size))+geom_bar(aes(fill = size))+scale_fill_manual(values =myColors)+
#previously explained function, cleans up barplot, add title
tune_figure(p)
```



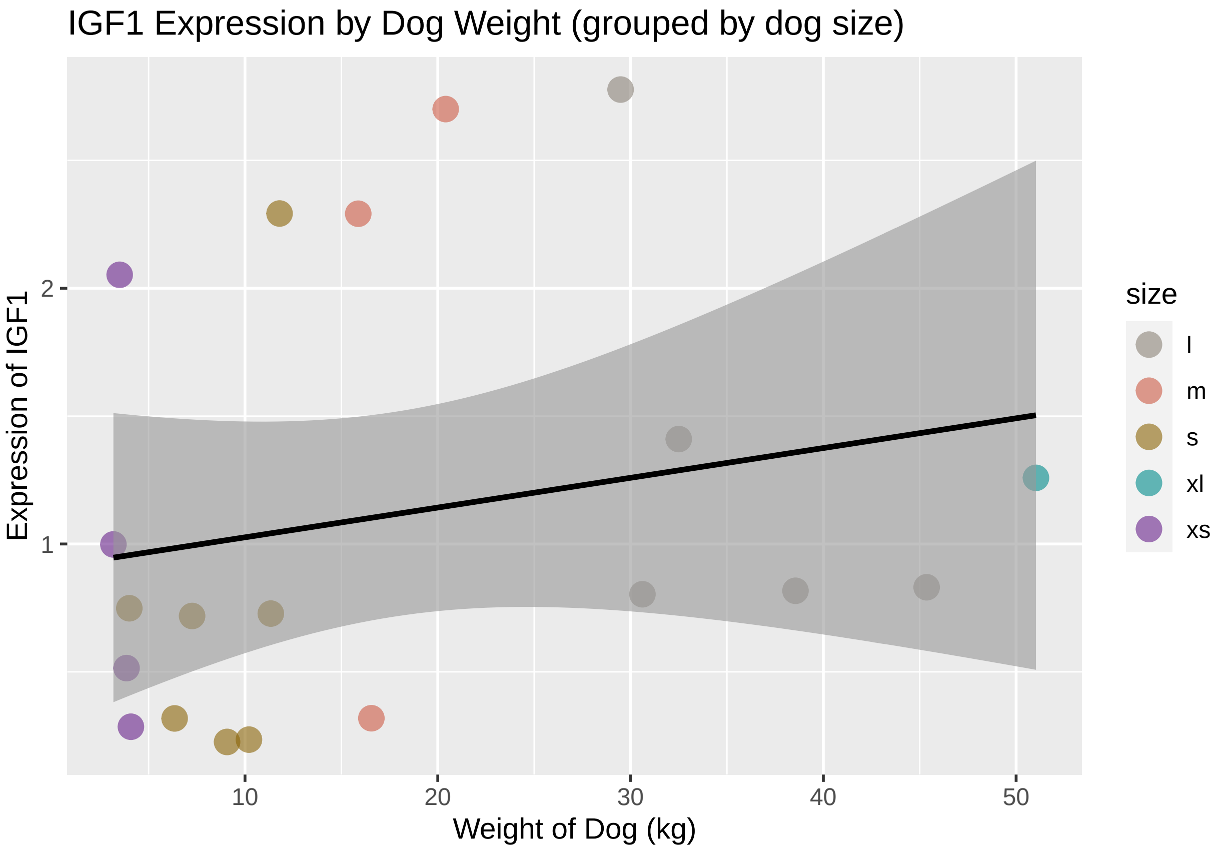
This data is much more balanced than our sequence data but the dataset favors smaller dogs

EDA of IGF1

Now we will begin analysis on our first gene of interest, IGF1. Since the expression data is numeric we can plot the expression data vs. weight of the dog breed measured. We will then color the points according to the breeds size grouping

```
#runs the previously explained plot_expression to receive a scatter plot with a
#regression line of IGF1 expression data by dog size, returns the ggplot object
#to be displayed to the global variable, w
w<-plot_expression(excel_name="dog snps.xlsx",sheet_name = "IGF1",x="weight_kg",y="norm_exp",col="size"
w
```

```
## `geom_smooth()` using formula 'y ~ x'
```



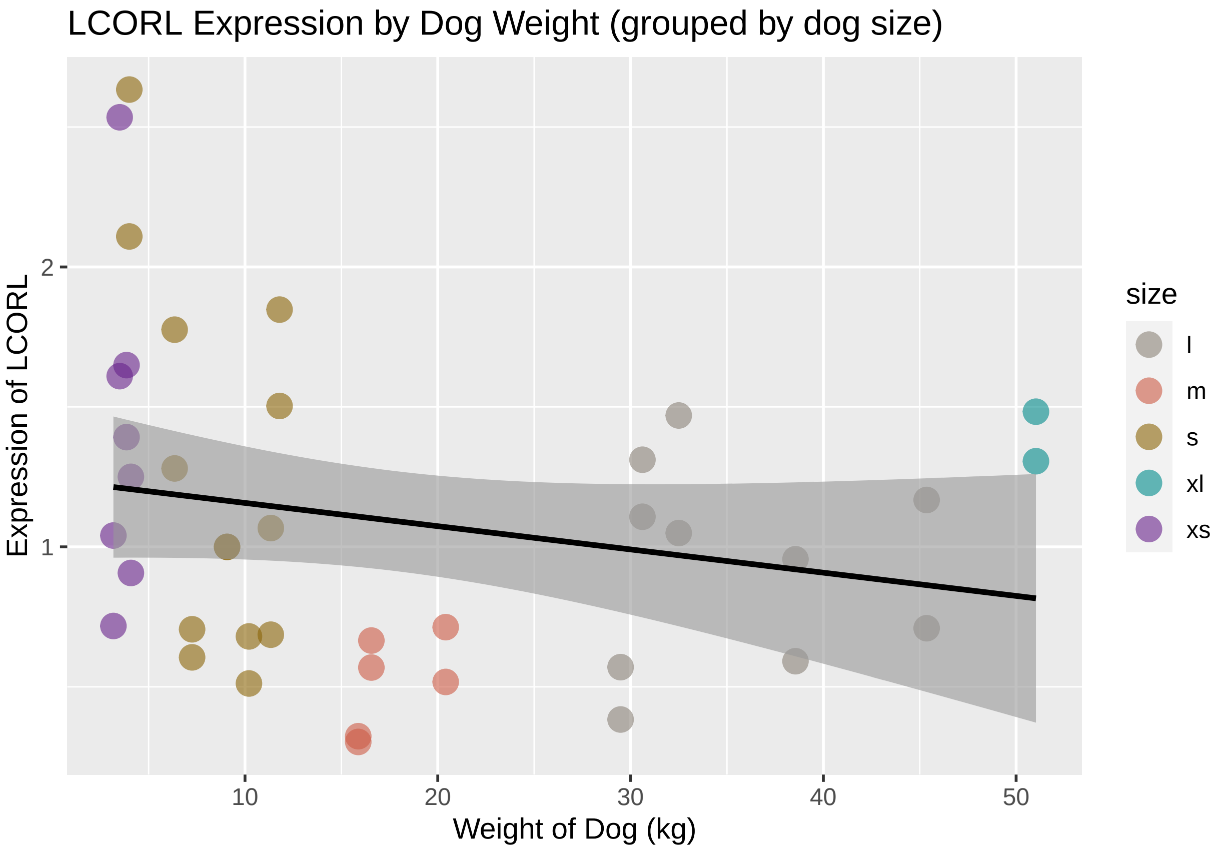
The expression of IGF1 seems have a loose positively correlation to weight, though there are outliers in the xs, s,m which have higher expression of IGF1 than even the largest dog in our dataset, the Irish Wolfhound. It is likely these dogs have lower expression of others factors associated with growth to account for this.

EDA of LCORL

Next we will begin analysis on LCORL Since the expression data is numeric we can plot the expression data vs. weight of the dog breed. We will then color the points according to the breeds' AKC size grouping

```
#runs the previously explained plot_expression to receive a scatter plot with a
#regression line of LCORL expression data by dog size, returns the ggplot object
#to be displayed to the global variable, w
w<-plot_expression(excel_name="dog snps.xlsx",sheet_name = "LCORL",x="weight_kg",y="norm_exp",col="size
w
```

```
## `geom_smooth()` using formula 'y ~ x'
```



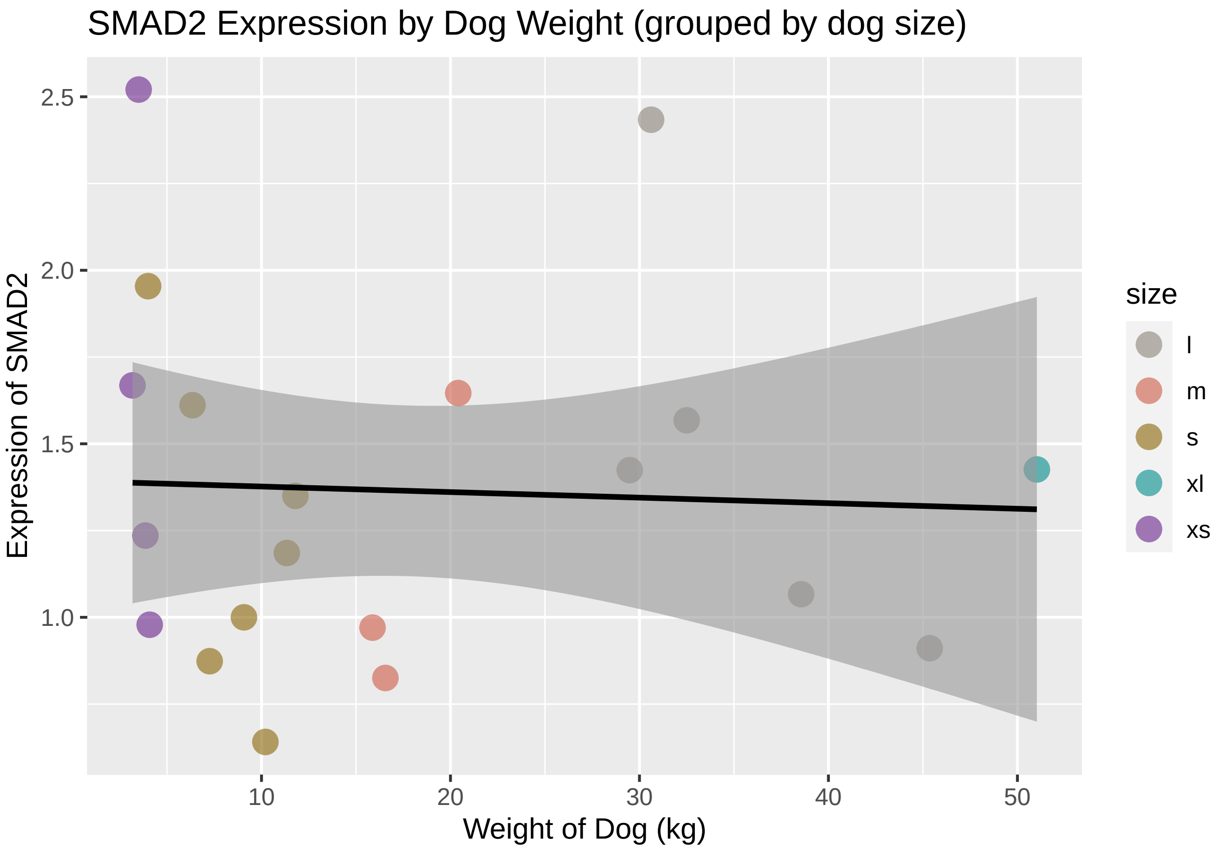
The expression of LCORL appears to be negatively correlated to weight. In most breeds this is a slight decrease in expression as size/weight increases, but for many dogs classified as xs and s, they have drastically higher expression than their m, l, and xl counterparts.

EDA of SMAD2

Since the expression data is numeric we can plot the expression data vs. weight of the dog breed measured. We will then color the points according to the breeds' AKC size grouping

```
#runs the previously explained plot_expression to receive a scatter plot with a
#regression line of SMAD2 expression data by dog size, returns the ggplot
#object to be displayed as the global variable, w
w<-plot_expression(excel_name="dog snps.xlsx",sheet_name = "SMAD2",x="weight_kg",y="norm_exp",col="size
w
```

```
## `geom_smooth()` using formula 'y ~ x'
```



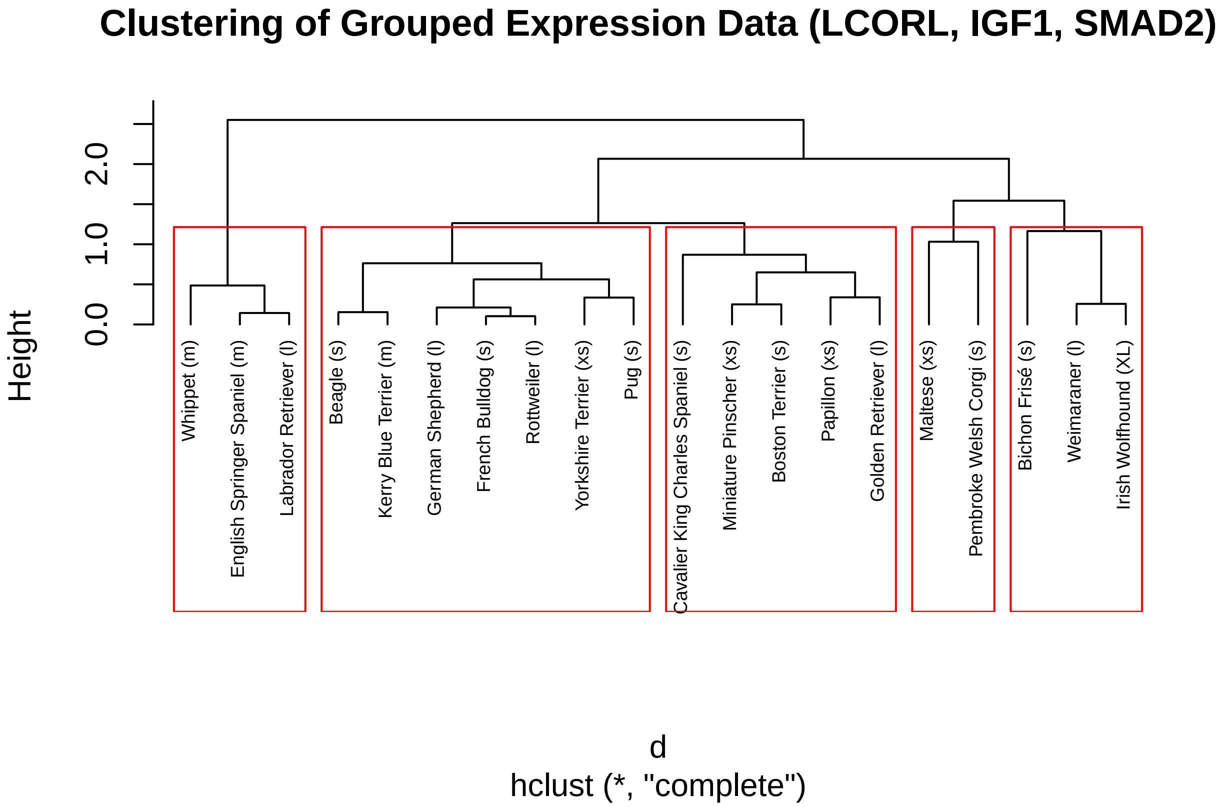
SMAD2 expression appears to be completely uncorrelated to either dog weight or dog size. Perhaps this lack of correlation is caused by the sample's bias towards smaller dogs.

Clustering

Now that we have an idea of what sort of data we are working with, we can move to clustering our expression data.

```
#clusters, d, fit, and groups are all outside of function definitions and are
#global variables

#read in excel data containing the expression data for all genes who had their
#scatter plot displayed (LCORL, IGF1, SMAD2) as a dataframe
clusters<-read.xlsx2("dog snps.xlsx",row.names=1,sheetName="clustering")

#form distance matrix with our dataframe cluster, using the "maximum" method to
#calculate distance
d <- dist(clusters, method = "maximum")

#now that we have a distance matrix, perform hierarchical clustering using the
#"complete" method to calculate cluster difference
fit <- hclust(d, method="complete")

#determine how many clusters we want our data to display, I chose 5 since that
#is the number of AKC size categories. You could also reasonably cluster with 4
#clusters, since the XL size classification only has one member in this dataset,
#the Irish Wolfhound
```

```r
groups <- cutree(fit, k=5)

#visualize the dendrogram
plot(fit, cex = 0.6, hang = -1, main="Clustering of Grouped Expression Data (LCORL, IGF1, SMAD2)")

#explicitly draw boxes surrounding the 5 clusters
rect.hclust(fit, k=5, border="red")
```



This is probably the most promising result in the whole notebook. We will now examine the the clusters from left to right

Note: All weights are in kg

- Cluster #1: Whippet (m, weight=15.9), English Springer Spaniel (m, weight=20.4), and Labrador retriever (l, weight=29.5). The English Springer Spaniel, and Whippet make sense to cluster together because of their similar weight, but the Labrador retriever seems to be an odd choice being both larger and heavier than the English Springer Spaniel and Whippet. The connecting factor between these three is life-span; In a study run by the American Kennel Club the Labrador retriever's median life span was found to be 135.5 months only 2.5 months shorter than the English Springer Spaniel's 135. The Whippet was identified as an outlier breed by the study for it's short lifespan of 118 months, especially considering its size (Lewis, T. W., 2018).
- Cluster #2: Is much more of a toss-up than cluster one. The grouping of the Yorkshire terrier (xs, weight=3.2) and the pug (s, weight=7.3) makes sense. The grouping of the German Shepherd (L, weight=38.6), French Bulldog (m, weight=11.3), and Rottweiler (l, weight=45.4) when considering past work on muscled dog breeds (like the French Bulldog) which have been observed to cluster more closely with larger, more muscled dog breeds than their same-size counterparts (Plassais, 2017). It is the combination of these two sensible branches into a cluster that is nonsensical, especially considering the inclusion of the unrelated Beagle and Kerry Blue Terrier branch, which has so reasonable explanation.

- Cluster #3: This is likely the most accurate cluster in the dendrogram. It's a grouping of small and extra small dogs with the inclusion of the Golden Retriever (l, weight=30.6). This result makes sense in the context of prior studies where Golden Retriever were found to an outlier regarding their longevity in canines not just their longevity in comparison to other canines (Lewis, T. W., 2018). Since we are examining genes associated with longevity it then makes sense for the long-lived Golden Retriever to cluster with smaller dogs known to also live longer.
- Cluster #4: The clustering of the Maltese (xs, weight=3.5), and the Pembroke Welsh Corgi (s, weight=11.8) is not an incompletely unrelated clustering since they belong to adjacent size groups.
- Cluster #5: The clustering of the Irish Wolfhound (xl, weight=51.0) and the Weimaraner (l, weight=32.5), especially then considering that the Irish Wolfhound is the only xl dog in the data set. Of course the inclusion of the Bichon Frisé (s, weight=4) is inexplicable since the Bichon Frisé is by no means a muscled dog.

Conclusion

In regards to sequence composition, the hypothesis has been disproved. The genotypic changes that cause huge phenotypic changes (both in morphology and life span) are too small to be detected through alignment, and clustering of these aligned sequences. Most of these changes are caused by SNPs which do not cause large, absolute changes.

In regards to gene expression, the hypothesis would need to be reexamined in more detail. When viewing the expression of singular longevity/size genes the trend between expression and size is relatively weak. But if you produce a data set which has multiple of these consistent smaller changes in expression based on dog size (e.g., clustering based off of the expression of multiple genes with similar functions) this pattern can begin to be elucidated with clustering. Perhaps in the future a more balanced expression data set in regards to dog size could produce more concrete results in this field.

Sources

Plassais, Jocelyn. "Analysis of large versus small dogs reveals three genes on the canine X chromosome associated with body weight, muscling, and backfat thickness." PLOS Genetics 2017. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5357063/

Metzger, Julia. "Expression Levels of LCORL Are Associated with Body Size in Horses." PLoS One 2019. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3572084/

Creevy, Kate. "The Companion Dog as a Model for the Longevity Dividend." Cold Springs Harbor Perspectives in Medicine 2016.https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4691800/

Lewis, T. W." Longevity and mortality in Kennel Club registered dog breeds in the UK in 2014" Canine genetics and epidemiology 2018. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6191922/

Plassais, J. "Whole-genome sequencing of canids reveals genomic regions under selection and variants influencing the morphology" Nature 2019. https://www.nature.com/articles/s41467-019-09373-w

Dextend usage: http://www.sthda.com/english/wiki/beautiful-dendrogram-visualizations-in-r-5-must-known-methods-unsupervised-machine-learning#plot.dendrogram-function