Phylogenetic Community Structure of *Carabidae* Species in France

To access this assignment (including the associated RMD file and data) online, please visit https://github.com/jacquelineliz/BINF6210_Assignment5

Introduction

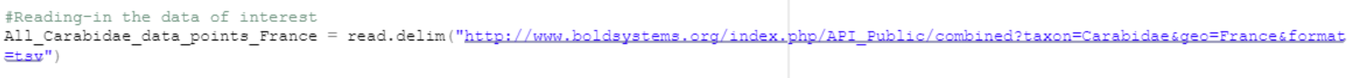
Carabidae is a globally distributed family of insects (Ball, 1978; BOLD, 2021; CABI, 2021). Individuals from the *Carabidae* family are commonly referred to as ground beetles (CABI, 2021). By 1978, approximately 40000 ground beetle species had been described, but detailed studies existed for fewer than 100 of them (Ball, 1978). The estimated number of ground beetle species has not changed much since then, but there is a much better understanding of their biology (Britannica, 2019). Ground beetles are holometabolous (Ball, 1978). Holometaboly is positively associated with species richness and evidence suggests a recent decrease in extinction in holometabolous insects (Nicholson et al., 2014, as cited in Ferns & Jervis, 2016; Ferns & Jervis, 2016). Carabids exist in various climates, including different humidities, temperatures and altitudes (Baust & Miller, 1970; Kirichenko-Babko et al., 2020; Ariza et al., 2021; BOLD, 2021; Ijala et al., 2021). Interestingly, ground beetles thrive after certain natural disasters and are quick to colonize freshly deglaciated sites (Moret et al., 2020; Riley Peterson et al., 2021). Their resilience may be attributed to a number of factors. For one, there exist carnivorous, herbivorous and omnivorous ground beetles (Thiele, 1977, as cited in Riley Peterson et al., 2021; Lövei & Sunderland, 1996, as cited in Riley Peterson et al., 2021). Furthermore, over 250 chemical compounds have been identified in relation to *Carabidae* defense (Rork & Renner, 2018).

Though, ground beetles live at various altitudes, they are not unaffected by it (Maveety et al., 2011; Staunton et al., 2016; Ijala et al., 2021). In Uganda, study by Ijala et al. found that altitude significantly influenced the abundance of insects from various *Carabidae* genera (2021). In Peru, a study by Maveety et al. showed that the distribution of species within the *Carabidae* family is restricted by altitude (2011). Specifically, the study sampled 5 altitudinal site and found that approximately 70% of the *Carabidae* species observed only existed at one of the sites (Maveety et al., 2011). In Spain and France, a 2012 study by Faille et al. was the first globally to identify *Trechus bruckoides* and *Trechus bouilloni*, two "species of high altitude".

Generally, graphing species richness in relation to altitude results in a hump-shaped distribution (Bertuzzo et al., 2016; Staunton et al., 2016). However, species richness may also display a negative correlation with altitude, as seen in a study by Maveety et al. (2011). The following assignment assesses the correlation between species distribution and altitude in France.

Data Set

Data was obtained from BOLD using the site's API.



```
#Reading-in the data of interest
All_Carabidae_data_points_France = read.delim("http://www.boldsystems.org/index.php/API_Public/combined?taxon=Carabidae&geo=France&format
=tsv")
```

Figure 1: Code to read-in the French *Carabidae* data.

The geographic region to examine was narrowed down by viewing the BOLD taxonomy browser and finalized by exploring species richness by country in R. Data points were filtered to retain those for the COI-5P gene. Filtering according to the presence of a specific gene was necessary to ensure that the species examined could be genetically compared to one another. The COI gene was chosen based on familiarity and data availability. For the dendrogram, data was further filtered to use sequences between 600 and 700 nucleotides long whenever possible. These limits were selected because they are approximately +/- 50 nucleotides from the length of the COI gene.

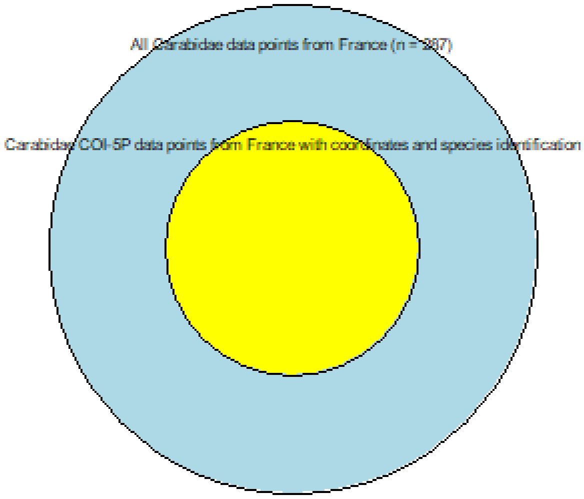

All Carabidae data points from France (n = 287)
Carabidae COI-5P data points from France with coordinates and species identification

Figure 2: Venn diagram of data filtration. Venn diagram showing the sample sizes relative to each other. The step yielding all *Carabidae* COI-5P data points from France with coordinates was skipped as it was visually indistinguishable from the sample of *Carabidae* COI-5P data points from France with coordinates and species identification.

<u>Software Tools</u>

To construct the dendrogram, the R packages msa, ape and DECIPHER were used (either directly or indirectly). msa was used to align the DNA sequences; ape was used to determine the distance between the sequences; and DECIPHER was directly used to construct the dendrogram through the function IdClusters. The distances between the sequences were obtained using the default method of ape::dist.dna (i.e. Kimura's 2-parameters distance). For the dendrogram, the neighbour joining method was used because of its speed, ability to handle large amounts of data, compatibility with Kimura's 2-parameters distance and general superiority to UPGMA (DECIPHER::IdClusters's default method).

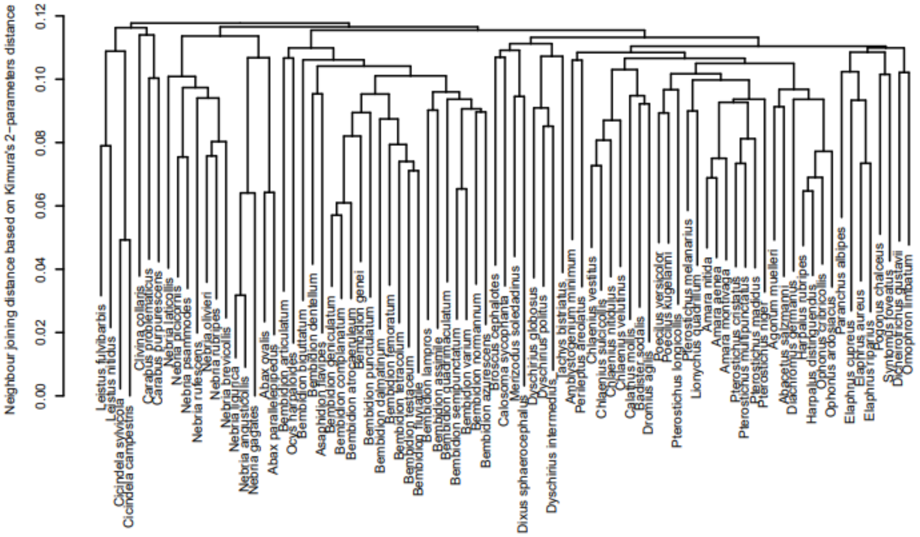


Figure 3: Dendrogram of Carabidae spp. in France. The dendrogram is constructed using the neighbour joining method with no cutoffs for sequences of the same cluster.

To assess diversity by altitude, a map of each community was created using the package maps and compared to a topological map from topological-map.com (2021). The package geosphere was also used in a function I wrote. The function separates the data points into communities.

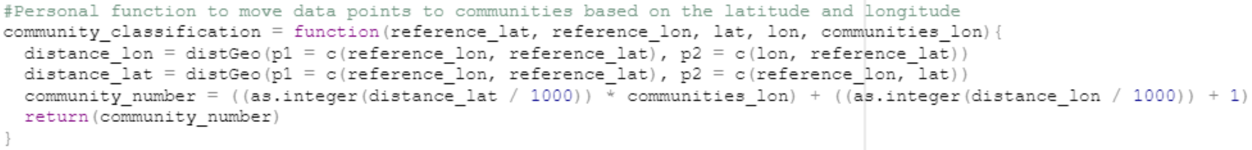

```
#Personal function to move data points to communities based on the latitude and longitude
community_classification = function(reference_lat, reference_lon, lat, lon, communities_lon){
  distance_lon = distGeo(p1 = c(reference_lon, reference_lat), p2 = c(lon, reference_lat))
  distance_lat = distGeo(p1 = c(reference_lon, reference_lat), p2 = c(reference_lon, lat))
  community_number = ((as.integer(distance_lat / 1000)) * communities_lon) + ((as.integer(distance_lon / 1000)) + 1)
  return(community_number)
}
```

Figure 4: Code for function community_classification. The arguments reference_lat and reference_lon indicate the latitude North of the country and the longitude farthest West of the country, respectively. The arguments lat and lon indicate the latitude and longitude of the data point being classified. The argument communities_lon indicates how many communities exist along the longitude of the country at any given latitude within the country.

Results and Discussion

By comparing the maps in Figure 5, we can see that the community where the two most common species are the most genetically similar is located in the region of France with the highest altitudes. By contrast the community where the two most common species are the most genetically different is located in a coastal region of northern France where the altitudes are low.

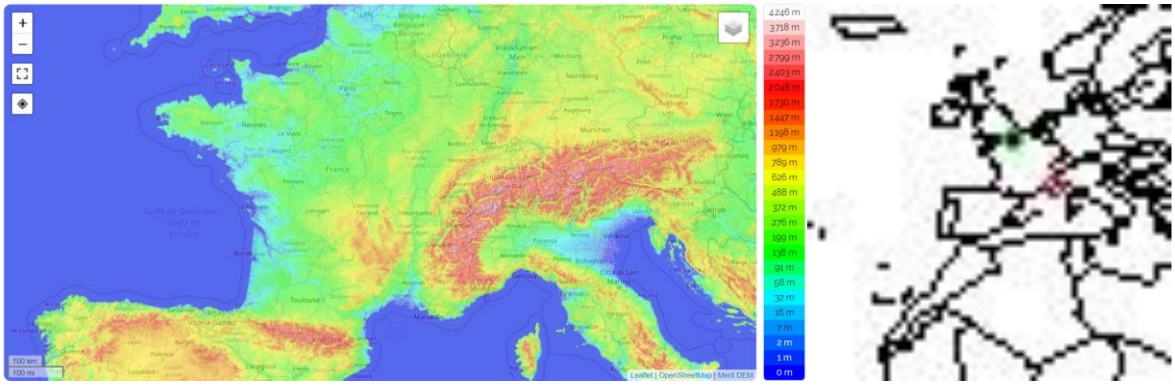
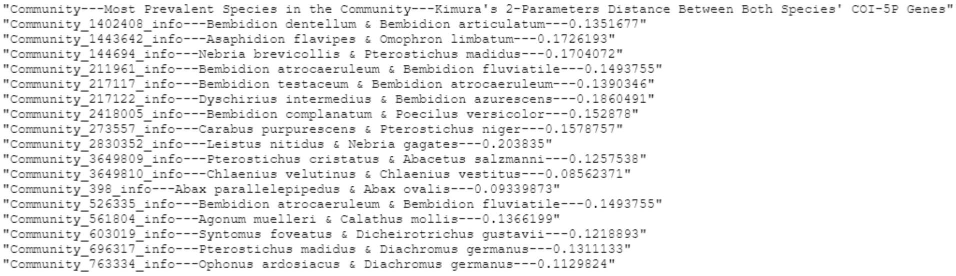


Figure 5: Maps of France. Left) Topographical map of France (topographic-map.com, 2021). Right) Map showing the location of the community with the most genetic difference between the two most common species within the community and the community with the least genetic difference between the two most common species within the community in green and red respectively.

This suggests that a negative correlation between Carabidae species diversity and altitude and/or Carabidae species richness and altitude holds true in France. Unfortunately, I did not examine enough communities or species within the communities to fully understand the correlation between Carabidae species diversity and altitude or Carabidae species richness and altitude. Given more time, I would expand my analysis to address this limitation. Figure 6 shows data available for communities in France.



```
"Community---Most Prevalent Species in the Community---Kimura's 2-Parameters Distance Between Both Species' COI-5P Genes"
"Community_1402408_info---Bembidion dentellum & Bembidion articulatum---0.1351677"
"Community_1443642_info---Asaphidion flavipes & Omophron limbatum---0.1726193"
"Community_144694_info---Nebria brevicollis & Pterostichus madidus---0.1704072"
"Community_211961_info---Bembidion atrocaeruleum & Bembidion fluviatile---0.1493755"
"Community_217117_info---Bembidion testaceum & Bembidion atrocaeruleum---0.1390346"
"Community_217122_info---Dyschirius intermedius & Bembidion azurescens---0.1860491"
"Community_2418005_info---Bembidion complanatum & Poecilus versicolor---0.152878"
"Community_273557_info---Carabus purpurescens & Pterostichus niger---0.1578757"
"Community_2830352_info---Leistus nitidus & Nebria gagates---0.203835"
"Community_3649809_info---Pterostichus cristatus & Abacetus salzmanni---0.1257538"
"Community_3649810_info---Chlaenius velutinus & Chlaenius vestitus---0.08562371"
"Community_398_info---Abax parallelepipedus & Abax ovalis---0.09339873"
"Community_526335_info---Bembidion atrocaeruleum & Bembidion fluviatile---0.1493755"
"Community_561804_info---Agonum muelleri & Calathus mollis---0.1366199"
"Community_603019_info---Syntomus foveatus & Dicheirotrichus gustavii---0.1218893"
"Community_696317_info---Pterostichus madidus & Diachromus germanus---0.1311133"
"Community_763334_info---Ophonus ardosiacus & Diachromus germanus---0.1129824"
```

Figure 6: Two most prevalent species by communities and the genetic distance between their COI genes.

I would also like to make my assessment of altitude more quantitative. This would be an important next step because quantitative approaches are typically less prone to error than qualitative ones, and we can see from Figure 5 that there is are quick changes in altitude in the South-East coastal regions of France further discouraging the use of judgement.
Although informative, I do not think that the figures I generated during my analysis are aesthetically pleasing or of the best quality. I would have liked to change the focus of the map (Figure 5, Right) (i.e. restrict the map to France as opposed to a world map). Furthermore, red and green were chosen for the map because I felt green has a positive connotation and red has a negative one. However, in retrospect, this choice of colours was not great given the prevalence of

red-green colour blindness. I also feel it necessary to better understand the functions involved in the graphics of dendrogram objects created with DECIPHER. I found that the visuals, including colours, resolution and figure margins changed a lot based on arguments that should be unrelated.

Acknowledgements

I would like to acknowledge consultation of my BINF*6210 assignment 1 in creating the Venn diagram for the current assignment. I would like to acknowledge consultation of my BINF*6210 assignment 2 and of the edits provided by Raamkumar Sivakumaar to said assignment. Raam's edits were helpful in adding a label to my dendrogram. I would like to acknowledge that I learnt how to use some of the functions (specifically those in the maps package) from Melanie Goens during BINF*6210 assignment 3. I would like to acknowledge consultation of BINF*6210 October 21st Introduction to Clustering and Phylogenetics slides in selecting the neighbour joining method for my dendrogram. Lastly, I would like to acknowledge that the length of the COI gene was learnt from the BINF*6210 October 19th prelecture video.

References

Ariza, G. M., Jácome, J., Esquivel, H. E., & Kotze, D. J. (2021). Early successional dynamics of ground beetles (Coleoptera, Carabidae) in the tropical dry forest ecosystem in Colombia. *ZooKeys*, *1044*, 877–906. https://doi.org/10.3897/zookeys.1044.59475

Ball. (1978). Ecology of the Carabidae: Carabid Beetles in Their Environments . A Study on Habitat Selection by Adaptations in Physiology and Behaviour. Hans-Ulrich Thiele. Translated from the German. Springer-Verlag, New York, 1977. xviii, 372 pp., illus. $44.20. Zoophysiology and Ecology, vol. 10. *Science (American Association for the Advancement of Science)*, *201*(4357), 704–705. https://doi.org/10.1126/science.201.4357.704

Barcode of Life Data System (BOLD) (accessed December 2021). Carabidae Taxonomy Browser. https://www.boldsystems.org/index.php/Taxbrowser_Taxonpage?taxon=carabidae&searchTax=Search+Taxonomy

Baust, J. G., & Miller, L. K. (1970). Variations in glycerol content and its influence on cold hardiness in the Alaskan carabid beetle, Pterostichus brevicornis. *Journal of Insect Physiology*, *16*(5), 979–990. https://doi.org/10.1016/0022-1910(70)90227-1

Becker, R. A., Wilks, A. R., Brownrigg, R., Minka, T. P., & Deckmyn, A. (2021). Package 'maps': Draw Geographical Maps (3.4.0). n.d.

Bertuzzo, E., Carrara, F., Mari, L., Altermatt, F., Rodriguez-Iturbe, I., & Rinaldo, A. (2016). Geomorphic controls on elevational gradients of species richness. *Proceedings of the National Academy of Sciences of the United States of America*, *113*(7), 1737–1742. https://doi.org/10.1073/pnas.1518922113

Bodenhofer, U., Bonatesta, E., Horejs-Kainrath, C., & Hochreiter S. (2015). msa: an R package for multiple sequence alignment. *Bioinformatics 31*(24), 3997–3999. DOI: bioinformatics/btv494.

Britannica, T. Editors of Encyclopaedia (2019, November 28). *ground beetle*. *Encyclopedia Britannica*. https://www.britannica.com/animal/ground-beetle

CABI (accessed December 2021). Carabidae (Ground Beetles). https://www.cabi.org/isc/datasheet/614

Faille, A., Bourdeau, C., & Fresneda, J. (2012). Molecular phylogeny of the Trechus brucki group, with description of two new species from the Pyreneo-Cantabrian area (France, Spain) (Coleoptera, Carabidae, Trechinae). *ZooKeys*, (217), 11–51. https://doi.org/10.3897/zookeys.217.3136

Ferns, P., & Jervis, M. (2016). Ordinal species richness in insects-a preliminary study of the influence of morphology, life history, and ecology. *Entomologia Experimentalis et Applicata*, *159*(2), 270–284. https://doi.org/10.1111/eea.12417

Hijmans, R. J. (2021). Package 'geosphere': Spherical Trigonometry (1.5-14). n.d.

Ijala, A. R., Kyamanywa, S., Cherukut, S., Sebatta, C., Hilger, T., & Karungi, J. (2021). Can Occurrence and Distribution of Ground Beetles (Carabidae) Be Influenced by the Coffee Farming System in the Mount Elgon Region of Uganda?. *Neotropical entomology*, *50*(4), 562–570. https://doi.org/10.1007/s13744-021-00872-4

Kirichenko-Babko, M., Danko, Y., Musz-Pomorksa, A., Widomski, M. K., & Babko, R. (2020). The Impact of Climate Variations on the Structure of Ground Beetle (Coleoptera: Carabidae) Assemblage in Forests and Wetlands. *Forests*, *11*(10), 1074–. https://doi.org/10.3390/f11101074

Maveety, S. A., Browne, R. A., & Erwin, T. L. (2011). Carabidae diversity along an altitudinal gradient in a Peruvian cloud forest (Coleoptera). *ZooKeys*, (147), 651–666. https://doi.org/10.3897/zookeys.147.2047

Moret, P., Barragán, Á., Moreno, E., Cauvy-Fraunié, S., & Gobbi, M. (2020). When the Ice Has Gone: Colonisation of Equatorial Glacier Forelands by Ground Beetles (Coleoptera: Carabidae). *Neotropical Entomology*, *49*(2), 213–226. https://doi.org/10.1007/s13744-019-00753-x

Paradis, E., Blomberg, S., Bolker, B., Brown, J., Claramunt, S., Claude, J., Cuong, H. S., Desper, R., Didier, G., Durand, B., Dutheil, J., Ewing, R. J., Gascuel, O., Guillerme, T., Heibl, C., Ives, A., Jones, B., Krah, F., Lawson, D., … de Vienne, D. (2021). Package 'ape': Analyses of Phylogenetics and Evolution (5.5). n.d.

Riley Peterson, K., Browne, R. A., & Erwin, T. L. (2021). Carabid beetle (Coleoptera, Carabidae) richness, diversity, and community structure in the understory of temporarily flooded and non-flooded Amazonian forests of Ecuador. *ZooKeys*, *1044*(3), 831–876. https://doi.org/10.3897/zookeys.1044.62340

Rork, A. M., & Renner, T. (2018). Carabidae Semiochemistry: Current and Future Directions. *Journal of chemical ecology*, *44*(12), 1069–1083. https://doi.org/10.1007/s10886-018-1011-8

R Core Team. (2021). *R: A language and environment for statistical computing* (4.1.0). R Foundation for Statistical Computing.

Staunton, K. M., Nakamura, A., Burwell, C. J., Robson, S. K., & Williams, S. E. (2016). Elevational Distribution of Flightless Ground Beetles in the Tropical Rainforests of North-Eastern Australia. *PloS one*, *11*(5), e0155826. https://doi.org/10.1371/journal.pone.0155826

topographic-map.com (accessed December 2021). Topographic map of France. https://fr-fr.topographic-map.com/maps/6/France/

Venables, W. N., Smith, D. N., & R Core Team. (2021). *An introduction to R* (4.1.0). R Core Team.

Wright, E. S. (2021). Package 'DECIPHER': Tools for curating, analyzing, and manipulating biological sequences (2.22.0). n.d.

Appendix A: RMarkdown PDF Output

BINF6210_Assignment5

Jacqueline Wu

17/12/2021

I consulted https://bookdown.org/yihui/rmarkdown/pdf-document.html#latex-options to set the margin size.

```
#tidy.opts and tidy argument settings based on https://stackoverflow.com/questions/33481271/how-to-wrap
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE, echo = TRUE)
```

Please Note: There appears to be a problem exporting the figure created using the function maps::map. The points on the map exported in the pdf are consistently different from those obtained in R.

#Supplementary R Code

The following code demonstrates how I chose the region (country) for assignment 5.

```
# Reading-in the data
Carabidae_data = read.delim("http://www.boldsystems.org/index.php/API_Public/combined?taxon=Carabidae&f

# For time sake, I narrowed down the countries by looking
# at BOLD Country Australia Austria Bolivia South Africa
# France Pakistan Indonesia Finland Norway Netherlands
# Kenya Argentina China Ecuador Italy Mexico Spain Peru
# Number of data points 814 713 466 424 403 401 400 360 352
# 337 273 229 168 142 131 120 112 105

Potential_Countries = c("Australia", "Austria", "Bolivia", "South Africa",
    "France", "Pakistan", "Indonesia", "Finland", "Norway", "Netherlands",
    "Kenya", "Argentina", "China", "Ecuador", "Italy", "Mexico",
    "Spain", "Peru")

for (i in 1:length(Potential_Countries)) {
    assign((sprintf("%s_Data_Points", Potential_Countries[i])),
        Carabidae_data[Carabidae_data$country == Potential_Countries[i],
            ])

    # The data points need to have known coordinates to
    # separate them spatially. They also need a common
    # gene. (for loop extended to check for both
    # conditions)
    temporary_data_frame = Carabidae_data[Carabidae_data$country ==
        Potential_Countries[i], ]
    second_temporary_data_frame = temporary_data_frame[(is.na(temporary_data_frame$lat) ==
        FALSE & is.na(temporary_data_frame$lon) == FALSE & temporary_data_frame$markercode ==
        "COI-5P"), ]
    assign((sprintf("%s_Data_Points_with_lat_and_lon", Potential_Countries[i])),
        second_temporary_data_frame)

    if (nrow(second_temporary_data_frame) > 50) {
        # Need to add 1 to 50 to account for NAs
        if (length(unique(second_temporary_data_frame$species_name)) >
            51) {
            print(paste(Potential_Countries[i], " has enough species for the analysis.",
                collapse = "", sep = ""))
        }
    }
}
```

```
## [1] "Austria has enough species for the analysis."
## [1] "France has enough species for the analysis."
## [1] "Finland has enough species for the analysis."
## [1] "Norway has enough species for the analysis."
## [1] "Italy has enough species for the analysis."
```

```
# The output informed me that I could choose Austria,
# France, Finland, Norway or Italy. Based on the output and
# quick manual exploration, I chose France.
```

#Assignment 5 Code

##Part 1/6 (Reading-in data and general organization)

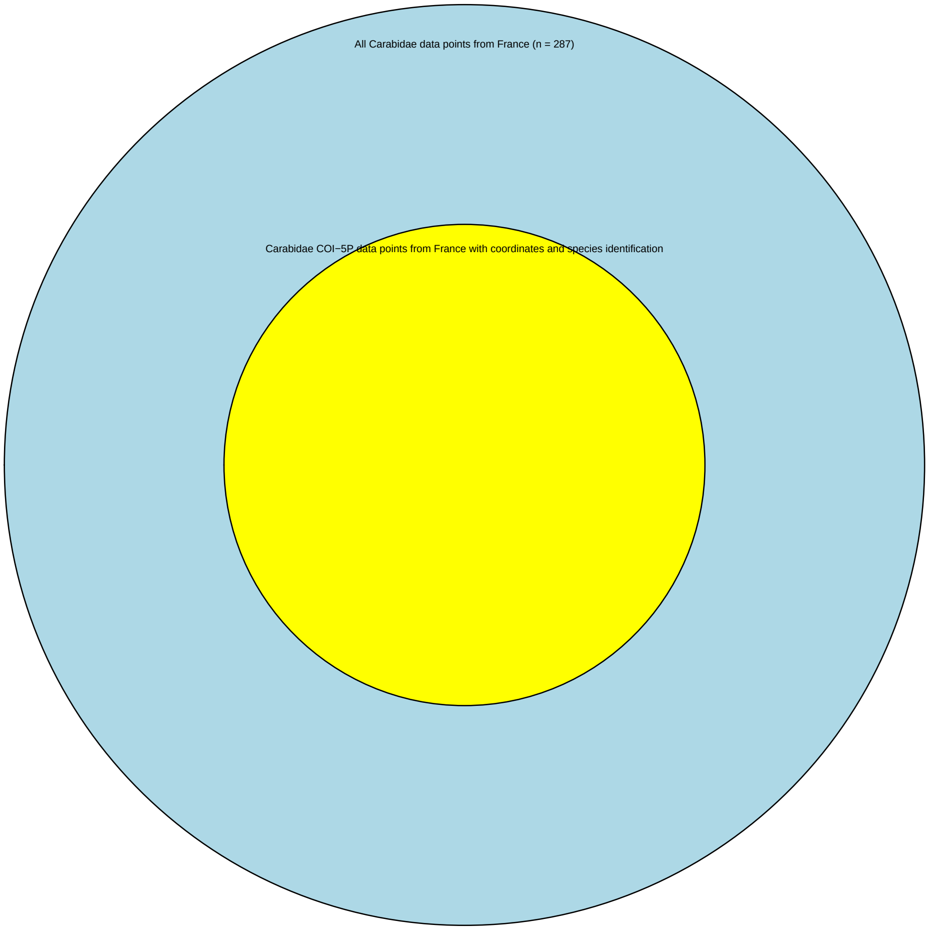
```
# The output from the following chuck of code would
# normally be place in a separate file, hence why a removed
# the output from the PDF. link consulted:
# https://stackoverflow.com/questions/47710427/how-to-show-code-but-hide-output-in-rmarkdown

# Reading-in the data of interest
All_Carabidae_data_points_France = read.delim("http://www.boldsystems.org/index.php/API_Public/combined

# Export the data to a separate file for reproducibility
# (static copy of the data set)
sink("French_Carabidae_Static_Copy.txt")
for (i in 1:nrow(All_Carabidae_data_points_France)) {
    for (e in 1:ncol(All_Carabidae_data_points_France)) {
        cat(All_Carabidae_data_points_France[i, e])
        if (e < ncol(All_Carabidae_data_points_France)) {
            cat(";")
        }
    }
    if (i < nrow(All_Carabidae_data_points_France)) {
        cat("\n")
    }
}
sink()

# Some filtration
Carabidae_COI5P_data_points_with_lat_and_lon_France = All_Carabidae_data_points_France[(is.na(All_Carab
    FALSE & is.na(All_Carabidae_data_points_France$lon) == FALSE &
    All_Carabidae_data_points_France$markercode == "COI-5P"),
    ]
Carabidae_COI5P_data_points_with_species_identification_and_coordinates_France = Carabidae_COI5P_data_po
    "", ]

# Organizing the data
sink("sequences_assignment5.txt")
for (i in 1:nrow(Carabidae_COI5P_data_points_with_species_identification_and_coordinates_France)) {
    sink("sequences_assignment5.txt", append = TRUE)
    cat(paste(Carabidae_COI5P_data_points_with_species_identification_and_coordinates_France$species_nam
        ";", Carabidae_COI5P_data_points_with_species_identification_and_coordinates_France$nucleotides
        collapse = "", sep = ""))
    sink()
    if (i < nrow(Carabidae_COI5P_data_points_with_species_identification_and_coordinates_France)) {
        sink("sequences_assignment5.txt", append = TRUE)
        cat("\n")
        sink()
    }
}
sink()

species_sequences_assignment5 = read.csv("sequences_assignment5.txt",
    header = FALSE, sep = ";")
```

##Part 2/6 (Visual representation of data filtration)

The following code was written to create a Venn diagram showing sample size at each filtration step.

```
labels_venn_diagram = c(paste("All Carabidae data points from France (n = ",
    nrow(All_Carabidae_data_points_France), ")", collapse = "",
    sep = ""))

symbols(x = 0, y = 0, xlim = c(-55, 55), ylim = c(-55, 55), col.axis = "transparent",
    col.lab = "transparent", bty = "n", xaxt = "n", yaxt = "n",
    circles = 50, bg = "light blue", inches = FALSE)

# Placement is 1 less than the upper limit of the circle
text(x = 0, y = 49, labels = labels_venn_diagram, cex = 0.5)

new_circle_diameter = (nrow(Carabidae_COI5P_data_points_with_lat_and_lon_France)/nrow(All_Carabidae_data
    100

# The lines below were removed for clarity symbols(x = 0, y
# = 0, circles = (new_circle_diameter / 2), bg = 'green',
# inches = FALSE, add = TRUE)

# text(x = 0, y = ((new_circle_diameter / 2) - 1), labels =
# 'Carabidae COI-5P data points with coordinates from
# France', cex = 0.5)

new_circle_diameter_2 = (nrow(Carabidae_COI5P_data_points_with_species_identification_and_coordinates_Fr
    new_circle_diameter
```

```
symbols(x = 0, y = 0, circles = (new_circle_diameter_2/2), bg = "yellow",
    inches = FALSE, add = TRUE)

text(x = 0, y = ((new_circle_diameter_2/2) - 1), labels = "Carabidae COI-5P data points from France wit
    cex = 0.5)
```



##Part 3/6 (General phylogeny of Carabidae)

The following code shows how I found the phylogenetic relation between species in the filtered data set and how I created to dendrogram to visually represent it.

```
# Filtering the sequences to keep only one for each species
# (for the dendrogram)
species_to_include = unique(species_sequences_assignment5$V1)
```

```r
sink("filtered_data_frame.txt")
cat("")
sink()

for (i in 1:length(species_to_include)) {
    assign(paste(species_to_include[i], "_data_frame"), species_sequences_assignment5[species_sequences_
        species_to_include[i], ])
    third_temporary_data_frame = species_sequences_assignment5[species_sequences_assignment5$V1 ==
        species_to_include[i], ]
    data_indices = c()
    if (nrow(third_temporary_data_frame) > 1) {
        for (q in 1:nrow(third_temporary_data_frame)) {
            if (nchar(third_temporary_data_frame$V2[q]) < 700 &
                nchar(third_temporary_data_frame$V2[q]) > 500) {
                data_indices = c(data_indices, q)
            }
        }
        if (is.null(data_indices)) {
            sink("filtered_data_frame.txt", append = TRUE)
            cat(paste(third_temporary_data_frame$V1[1], ";",
                third_temporary_data_frame$V2[1], collapse = "",
                sep = ""))
            if (i < length(species_to_include)) {
                cat("\n")
            }
            sink()
        } else {
            a = sample(x = data_indices, size = 1)
            sink("filtered_data_frame.txt", append = TRUE)
            cat(paste(third_temporary_data_frame$V1[a], ";",
                third_temporary_data_frame$V2[a], collapse = "",
                sep = ""))
            if (i < length(species_to_include)) {
                cat("\n")
            }
            sink()
        }
    } else {
        sink("filtered_data_frame.txt", append = TRUE)
        cat(paste(third_temporary_data_frame$V1, ";", third_temporary_data_frame$V2,
            collapse = "", sep = ""))
        if (i < length(species_to_include)) {
            cat("\n")
        }
        sink()
    }
}

filtered_data_frame_France_Carabidae = read.csv("filtered_data_frame.txt",
    sep = ";", header = FALSE)

# Determining the phylogeny
# Converting sequence data to a fasta file for
# compatibility with the msa readDNAStringSet function.
sink("filtered_data_frame_France_Carabidae_fasta_version.fasta")
cat("")
sink()
for (i in 1:nrow(filtered_data_frame_France_Carabidae)) {
    sink("filtered_data_frame_France_Carabidae_fasta_version.fasta",
        append = TRUE)
    cat(">")
    cat(filtered_data_frame_France_Carabidae$V1[i])
    cat("\n")
    cat(filtered_data_frame_France_Carabidae$V2[i])
    # If statement to avoid an empty line at the end of the
    # file
    if (i < nrow(filtered_data_frame_France_Carabidae)) {
        cat("\n")
    }
    sink()
}

# The libraries were read-in directly before their use to
# minimize any conflicts between functions. Attaching msa
# library
library(msa)
```

```
## Loading required package: Biostrings

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: ’BiocGenerics’

## The following objects are masked from ’package:parallel’:
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## 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

## Warning: package ’S4Vectors’ was built under R version 4.1.1

## Loading required package: stats4

##
## Attaching package: ’S4Vectors’

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

## Loading required package: IRanges

##
## Attaching package: ’IRanges’

## The following object is masked from ’package:grDevices’:
##
##     windows

## Loading required package: XVector

## Loading required package: GenomeInfoDb

## Warning: package ’GenomeInfoDb’ was built under R version 4.1.1

##
## Attaching package: ’Biostrings’

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

```
Carabidae_DNA_string_set = readDNAStringSet("filtered_data_frame_France_Carabidae_fasta_version.fasta")
Carabidae_DNA_string_set_alignment = msa(Carabidae_DNA_string_set,
    method = "Muscle")

# Attaching ape library
library(ape)
```

```
## Warning: package ’ape’ was built under R version 4.1.1

##
## Attaching package: ’ape’

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

```
Carabidae_DNA_string_set_distance = dist.dna(as.DNAbin(Carabidae_DNA_string_set_alignment),
    as.matrix = TRUE, pairwise.deletion = TRUE)

# Attaching DECIPHER library
library(DECIPHER)
```

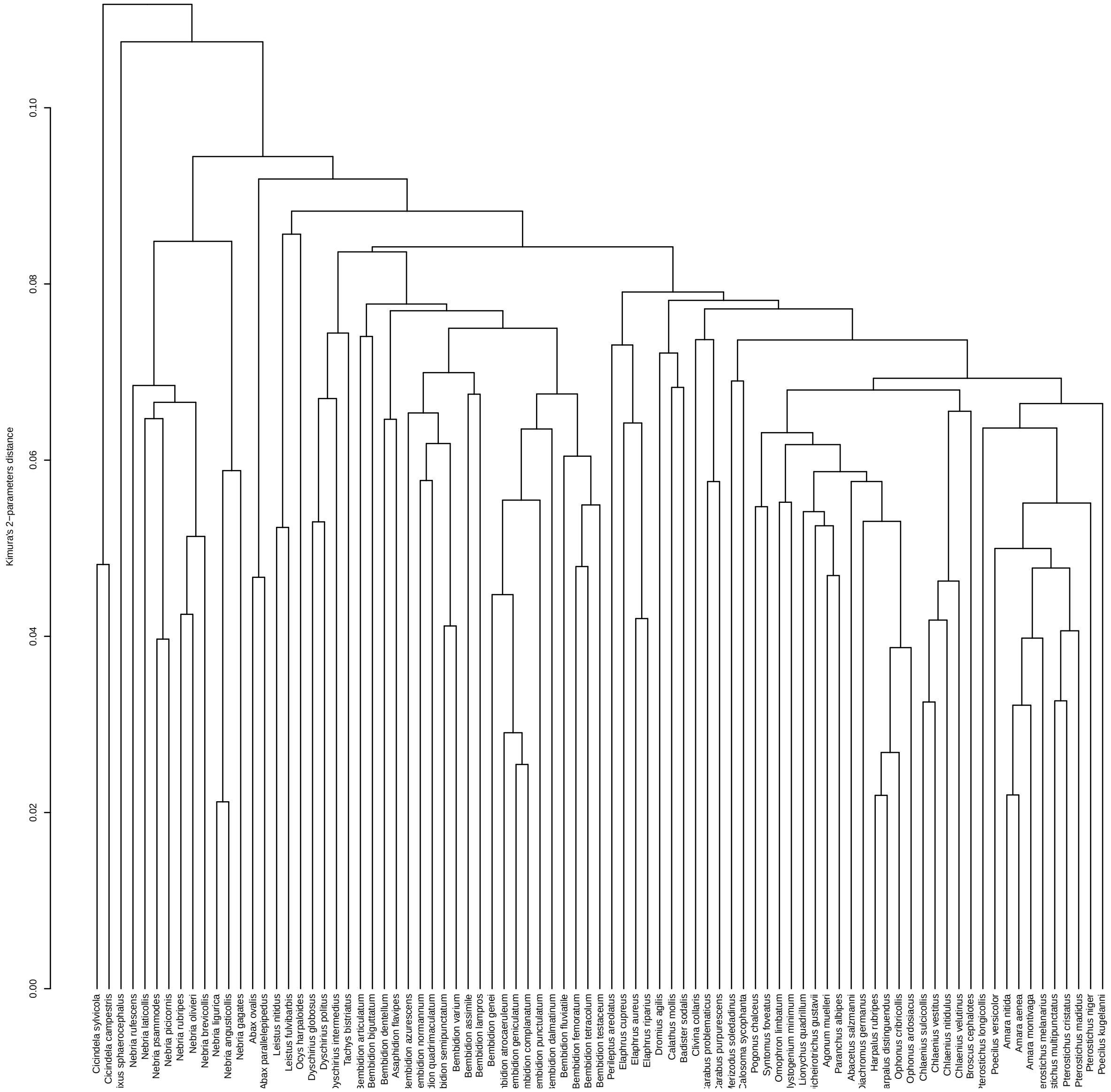
```
## Loading required package: RSQLite
```

```
## Warning: package 'RSQLite' was built under R version 4.1.1
```

```
Carabidae_phylogeny = IdClusters(myDistMatrix = Carabidae_DNA_string_set_distance,
    type = "dendrogram")
```

```
## ================================================================================
##
## Time difference of 0.02 secs
```

```
par(cex = 0.5)
plot(Carabidae_phylogeny, ylab = "Kimura's 2-parameters distance")
```



##Part 4/6 (Sorting data points into communities)

Communities were sorted using a grid system as detailed in the code below.

```
upper_lat = max(Carabidae_COI5P_data_points_with_species_identification_and_coordinates_France$lat)
lower_lat = min(Carabidae_COI5P_data_points_with_species_identification_and_coordinates_France$lat)

upper_lon = max(Carabidae_COI5P_data_points_with_species_identification_and_coordinates_France$lon)
lower_lon = min(Carabidae_COI5P_data_points_with_species_identification_and_coordinates_France$lon)

# 1 degree latitude by 1 degree longitude does not give a
# consistent area (more area towards the equator).

upper_lon_lower_lat = c(upper_lon, lower_lat)
upper_lon_upper_lat = c(upper_lon, upper_lat)
```

```r
lower_lon_lower_lat = c(lower_lon, lower_lat)
lower_lon_upper_lat = c(lower_lon, upper_lat)

# Attaching geosphere
library(geosphere)
```

```
## Warning: package 'geosphere' was built under R version 4.1.2
```

```r
distance_lon = distGeo(p1 = upper_lon_lower_lat, p2 = lower_lon_lower_lat)
distance_lat = distGeo(p1 = upper_lon_upper_lat, p2 = upper_lon_lower_lat)
distance_lon_km = ((as.integer(distance_lon))/1000) + 1
distance_lat_km = ((as.integer(distance_lat))/1000) + 1

# Figuring out how many communities along the latitude and
# along the longitude given the decision that each km^2
# will be a separate community
communities_dimensions = c(as.integer(distance_lon_km) + 1, as.integer(distance_lat_km) +
    1)

# Personal function to move data points to communities
# based on the latitude and longitude
community_classification = function(reference_lat, reference_lon,
    lat, lon, communities_lon) {
    distance_lon = distGeo(p1 = c(reference_lon, reference_lat),
        p2 = c(lon, reference_lat))
    distance_lat = distGeo(p1 = c(reference_lon, reference_lat),
        p2 = c(reference_lon, lat))
    community_number = ((as.integer(distance_lat/1000)) * communities_lon) +
        ((as.integer(distance_lon/1000)) + 1)
    return(community_number)
}

communities_numbers = c()
for (i in 1:nrow(Carabidae_COI5P_data_points_with_species_identification_and_coordinates_France)) {
    communities_numbers = c(communities_numbers, community_classification(reference_lat = upper_lat,
        reference_lon = lower_lon, lat = Carabidae_COI5P_data_points_with_species_identification_and_coo
        lon = Carabidae_COI5P_data_points_with_species_identification_and_coordinates_France$lon[i],
        communities_lon = communities_dimensions[1]))
}

France_Carabidae_with_communities = base::cbind(Carabidae_COI5P_data_points_with_species_identification_
    communities_numbers)

communities_observed = unique(communities_numbers)

community_info_vector = c()
for (i in 1:length(communities_observed)) {
    assign(paste("Community_", communities_observed[i], "_info",
        collapse = "", sep = ""), France_Carabidae_with_communities[France_Carabidae_with_communities$co
        communities_observed[i], ])
    community_info_vector = c(community_info_vector, paste("Community_",
        communities_observed[i], "_info", collapse = "", sep = ""))
}
```

##Part 5/6 (Getting information on common species within communities)

```
sink("common_species_pairs_by_community.txt")
cat("")
sink()

for (i in 1:length(community_info_vector)) {
    community_examine = get(community_info_vector[i])
    community_examine_species = unique(community_examine$species_name)
    if (length(community_examine_species) > 1) {
        if (length(community_examine_species) == 2) {
            sink("common_species_pairs_by_community.txt", append = TRUE)
            cat(community_info_vector[i])
            cat(";")
            cat(community_examine_species[1])
            cat(";")
            cat(community_examine_species[2])
            sink()
            if (i < length(community_info_vector)) {
                sink("common_species_pairs_by_community.txt",
                  append = TRUE)
                cat("\n")
                sink()
            }
        }
        if (length(community_examine_species) > 2) {
            sorting_by_prevalence_intermediate = community_examine_species
            sorting_by_prevalence = sorting_by_prevalence_intermediate
            for (q in 1:length(sorting_by_prevalence)) {
                sorting_by_prevalence[q] = length(community_examine$species_name[community_examine$speci
                  sorting_by_prevalence_intermediate[q]])
            }
            max_occurrence = max(sorting_by_prevalence)
            index_of_max_occurrence = c()
            for (z in 1:length(sorting_by_prevalence)) {
                if (sorting_by_prevalence[z] == max_occurrence) {
                  index_of_max_occurrence = c(index_of_max_occurrence,
                    z)
                }
            }
            if (length(index_of_max_occurrence) > 1) {
                sink("common_species_pairs_by_community.txt",
                  append = TRUE)
                cat(community_info_vector[i])
                cat(";")
                cat(sorting_by_prevalence_intermediate[index_of_max_occurrence[1]])
                cat(";")
                cat(sorting_by_prevalence_intermediate[index_of_max_occurrence[2]])
                if (i < length(community_info_vector)) {
                  cat("\n")
                }
                sink()
            }
            if (length(index_of_max_occurrence) == 1) {
                most_prevalent_species = sorting_by_prevalence_intermediate[index_of_max_occurrence[1]]
                community_examine_without_most_prevalent = community_examine[community_examine$species_n
                  most_prevalent_species, ]
                looking_for_second_most_prevalent = unique(community_examine_without_most_prevalent$spec
                looking_for_second_most_prevalent_ind = looking_for_second_most_prevalent
                for (u in 1:length(looking_for_second_most_prevalent_ind)) {
                  looking_for_second_most_prevalent_ind[u] = length(community_examine_without_most_preva
                    looking_for_second_most_prevalent[u]])
                }
                second_max_occurrence = max(looking_for_second_most_prevalent_ind)
                index_of_second_max_occurrence = c()
                for (g in 1:length(looking_for_second_most_prevalent)) {
                  if (looking_for_second_most_prevalent_ind[g] ==
                    second_max_occurrence) {
                    index_of_second_max_occurrence = c(index_of_second_max_occurrence,
                      g)
                  }
                }
                sink("common_species_pairs_by_community.txt",
                  append = TRUE)
                cat(community_info_vector[i])
                cat(";")
                cat(most_prevalent_species)
                cat(";")
                cat(looking_for_second_most_prevalent[index_of_second_max_occurrence[1]])
                if (i < length(community_info_vector)) {
                  cat("\n")
                }
                sink()
            }
        }
    }
}

Common_Species_by_Community = read.csv("common_species_pairs_by_community.txt",
    sep = ";", header = FALSE)

# Distance between species sequences
sink("Community_Species_Figure.txt")
cat("Community---Most Prevalent Species in the Community---Kimura's 2-Parameters Distance Between Both S
```

```
## Community---Most Prevalent Species in the Community---Kimura's 2-Parameters Distance Between Both Spe
```

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

```
sink()

for (v in 1:length(Common_Species_by_Community$V1)) {
    sink("Community_Species_Figure.txt", append = TRUE)
    cat(Common_Species_by_Community$V1[v])
    cat("---")
    cat(Common_Species_by_Community$V2[v])
    cat(" & ")
    cat(Common_Species_by_Community$V3[v])
    cat("---")
    sink()

    for (b in 1:length(colnames(Carabidae_DNA_string_set_distance))) {
        if (identical(colnames(Carabidae_DNA_string_set_distance)[b],
            Common_Species_by_Community$V2[v])) {
            index_colnames_one = b
        }
        if (identical(colnames(Carabidae_DNA_string_set_distance)[b],
            Common_Species_by_Community$V3[v])) {
            index_colnames_two = b
        }
    }
    sink("Community_Species_Figure.txt", append = TRUE)
    cat(Carabidae_DNA_string_set_distance[index_colnames_one,
        index_colnames_two])
    cat("\n")
    sink()
}
sink()

Common_Species_Figure_intermedate = readLines("Community_Species_Figure.txt")

Common_Species_Figure = sort(Common_Species_Figure_intermedate)
print(Common_Species_Figure)
```

```
##  [1] "Community_1402408_info---Bembidion dentellum & Bembidion articulatum---0.136886"
##  [2] "Community_1443642_info---Asaphidion flavipes & Omophron limbatum---0.1739249"
##  [3] "Community_144694_info---Nebria brevicollis & Pterostichus madidus---0.1745258"
##  [4] "Community_211961_info---Bembidion atrocaeruleum & Bembidion fluviatile---0.1493306"
##  [5] "Community_217117_info---Bembidion testaceum & Bembidion atrocaeruleum---0.1390346"
##  [6] "Community_217122_info---Dyschirius intermedius & Bembidion azurescens---0.1835364"
##  [7] "Community_2418005_info---Bembidion complanatum & Poecilus versicolor---0.1601697"
##  [8] "Community_273557_info---Carabus purpurescens & Pterostichus niger---0.1578757"
##  [9] "Community_2830352_info---Leistus nitidus & Nebria gagates---0.203835"
## [10] "Community_3649809_info---Pterostichus cristatus & Abacetus salzmanni---0.127592"
## [11] "Community_3649810_info---Chlaenius velutinus & Chlaenius vestitus---0.08562371"
## [12] "Community_398_info---Abax parallelepipedus & Abax ovalis---0.09339873"
## [13] "Community_526335_info---Bembidion atrocaeruleum & Bembidion fluviatile---0.1493306"
## [14] "Community_561804_info---Agonum muelleri & Calathus mollis---0.1366199"
## [15] "Community_603019_info---Syntomus foveatus & Dicheirotrichus gustavii---0.1218893"
## [16] "Community_696317_info---Pterostichus madidus & Diachromus germanus---0.13471"
## [17] "Community_763334_info---Ophonus ardosiacus & Diachromus germanus---0.1129824"
```

##Part 6/6 (Mapping the communities)

```
# Attaching library maps
library(maps)
```

```
## Warning: package 'maps' was built under R version 4.1.1
```

```
sink("Kimura_distance.txt")
cat("Kimuras_2Parameters_Distance_Between_Both_Species_COI5P_Genes")
```

```
## Kimuras_2Parameters_Distance_Between_Both_Species_COI5P_Genes
```

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

```
sink()

for (v in 1:length(Common_Species_by_Community$V1)) {
    for (b in 1:length(colnames(Carabidae_DNA_string_set_distance))) {
        if (identical(colnames(Carabidae_DNA_string_set_distance)[b],
            Common_Species_by_Community$V2[v])) {
            index_colnames_one = b
        }
        if (identical(colnames(Carabidae_DNA_string_set_distance)[b],
            Common_Species_by_Community$V3[v])) {
            index_colnames_two = b
        }
    }
    sink("Kimura_distance.txt", append = TRUE)
    cat(Carabidae_DNA_string_set_distance[index_colnames_one,
        index_colnames_two])
    cat("\n")
    sink()
}
sink()

kimura_distance = readLines("Kimura_distance.txt")
max_community_distance = max(kimura_distance)
min_community_distance = min(kimura_distance)

for (i in 1:length(kimura_distance)) {
    if (kimura_distance[i] == max_community_distance) {
        max_kimura_index = i
    }
    if (kimura_distance[i] == min_community_distance) {
        min_kimura_index = i
    }
}

max_kimura_community = communities_observed[max_kimura_index]
min_kimura_community = communities_observed[min_kimura_index]

max_kimura_df = France_Carabidae_with_communities[France_Carabidae_with_communities$communities_numbers
    max_kimura_community, ]
min_kimura_df = France_Carabidae_with_communities[France_Carabidae_with_communities$communities_numbers
    min_kimura_community, ]

map()
points(x = max_kimura_df$lon, y = max_kimura_df$lat, col = "green")
points(x = min_kimura_df$lon, y = min_kimura_df$lat, col = "red")
```