Chapter 14 Phylogenetics

Angelo LaCommare-Soto

2025-05-07

Evolutionary Divergence in *Leptasterias* with Sequence Data

Research Question 1: Does sequence data show evidence of divergence within *Leptasterias*, and are divergence patterns consistent with geographic structure?

Section 1 - Load and Align Sequences

```
# set working directory for all chunks in this file (default working directory is wherever Rmd file is)
getwd()
```

```
## [1] "C:/Users/Angelo L/Documents/GitHub/BIOL710/RCode710/RCode/working_directory"
```

```
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.3
```

```
## Warning: package 'purrr' was built under R version 4.4.3
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(Biostrings)
```

```
## Loading required package: BiocGenerics
##
## Attaching package: 'BiocGenerics'
##
## The following objects are masked from 'package:lubridate':
##
##     intersect, setdiff, union
```

```
##
## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union
##
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
##
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, 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, saveRDS, setdiff,
##     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:lubridate':
##
##     second, second<-
##
## The following objects are masked from 'package:dplyr':
##
##     first, rename
##
## The following object is masked from 'package:tidyr':
##
##     expand
##
## The following object is masked from 'package:utils':
##
##     findMatches
##
## 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:lubridate':
##
##     %within%
##
## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice
```

```
##
## The following object is masked from 'package:purrr':
##
##     reduce
##
## The following object is masked from 'package:grDevices':
##
##     windows
##
## Loading required package: XVector
##
## Attaching package: 'XVector'
##
## The following object is masked from 'package:purrr':
##
##     compact
##
## Loading required package: GenomeInfoDb
##
## Attaching package: 'Biostrings'
##
## The following object is masked from 'package:base':
##
##     strsplit
```

```
library(msa)
library(phangorn)
```

```
## Warning: package 'phangorn' was built under R version 4.4.3

## Loading required package: ape

## Warning: package 'ape' was built under R version 4.4.3

##
## Attaching package: 'ape'
##
## The following object is masked from 'package:Biostrings':
##
##     complement
##
## The following object is masked from 'package:dplyr':
##
##     where
```

```
library(ggtree)
```

```
## ggtree v3.14.0 Learn more at https://yulab-smu.top/contribution-tree-data/
##
## Please cite:
##
## Guangchuang Yu, Tommy Tsan-Yuk Lam, Huachen Zhu, Yi Guan. Two methods
```

```
## for mapping and visualizing associated data on phylogeny using ggtree.
## Molecular Biology and Evolution. 2018, 35(12):3041-3043.
## doi:10.1093/molbev/msy194
##
## 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
##
## The following object is masked from 'package:tidyr':
##
##     expand
```

```
library(ape)
library(maps)
```

```
## Warning: package 'maps' was built under R version 4.4.3
```

```
##
## Attaching package: 'maps'
##
## The following object is masked from 'package:purrr':
##
##     map
```

```
library(ggspatial)
```

```
## Warning: package 'ggspatial' was built under R version 4.4.3
```

```
library(tidytree)
```

```
## Warning: package 'tidytree' was built under R version 4.4.3
```

```
## If you use the ggtree package suite in published research, please cite
## the appropriate paper(s):
##
## LG Wang, TTY Lam, S Xu, Z Dai, L Zhou, T Feng, P Guo, CW Dunn, BR
## Jones, T Bradley, H Zhu, Y Guan, Y Jiang, G Yu. treeio: an R package
```

```
## for phylogenetic tree input and output with richly annotated and
## associated data. Molecular Biology and Evolution. 2020, 37(2):599-603.
## doi: 10.1093/molbev/msz240
##
## Guangchuang Yu.  Data Integration, Manipulation and Visualization of
## Phylogenetic Trees (1st edition). Chapman and Hall/CRC. 2022,
## doi:10.1201/9781003279242
##
## Attaching package: 'tidytree'
##
## The following objects are masked from 'package:ape':
##
##     drop.tip, keep.tip
##
## The following object is masked from 'package:S4Vectors':
##
##     rename
##
## The following object is masked from 'package:stats':
##
##     filter
```

```
# Load sequences
dna <- readDNAStringSet("coi_nucleotide.fasta")

# Align sequences
dna_aln <- msa(dna, method = "ClustalW")
```

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

Question Answers

a. There are 57 COI DNA sequences are in the dataset.
b. The metadata includes information about the locations from which each sea star was sampled and whether that sample is historical or contemporary.
c. The process of genetic sequence alignment assumes that the sequences from two different individual organisms are evolutionarily and/or functionally similar.
d. Some sequences are more challenging to align if they exhibit large mutations, such as multi nucleotide deletions or insertions. This can create many different scenarios involving which nucleotides best align with each other.

Section 2 - Building and Comparing Trees

```
# Convert aligned sequences to phyDat format
dna_phy <- as.phyDat(msaConvert(dna_aln, type = "seqinr"), type = "DNA")

# View structure in the console
dna_phy
```

```
## 57 sequences with 536 character and 91 different site patterns.
## The states are a c g t
```

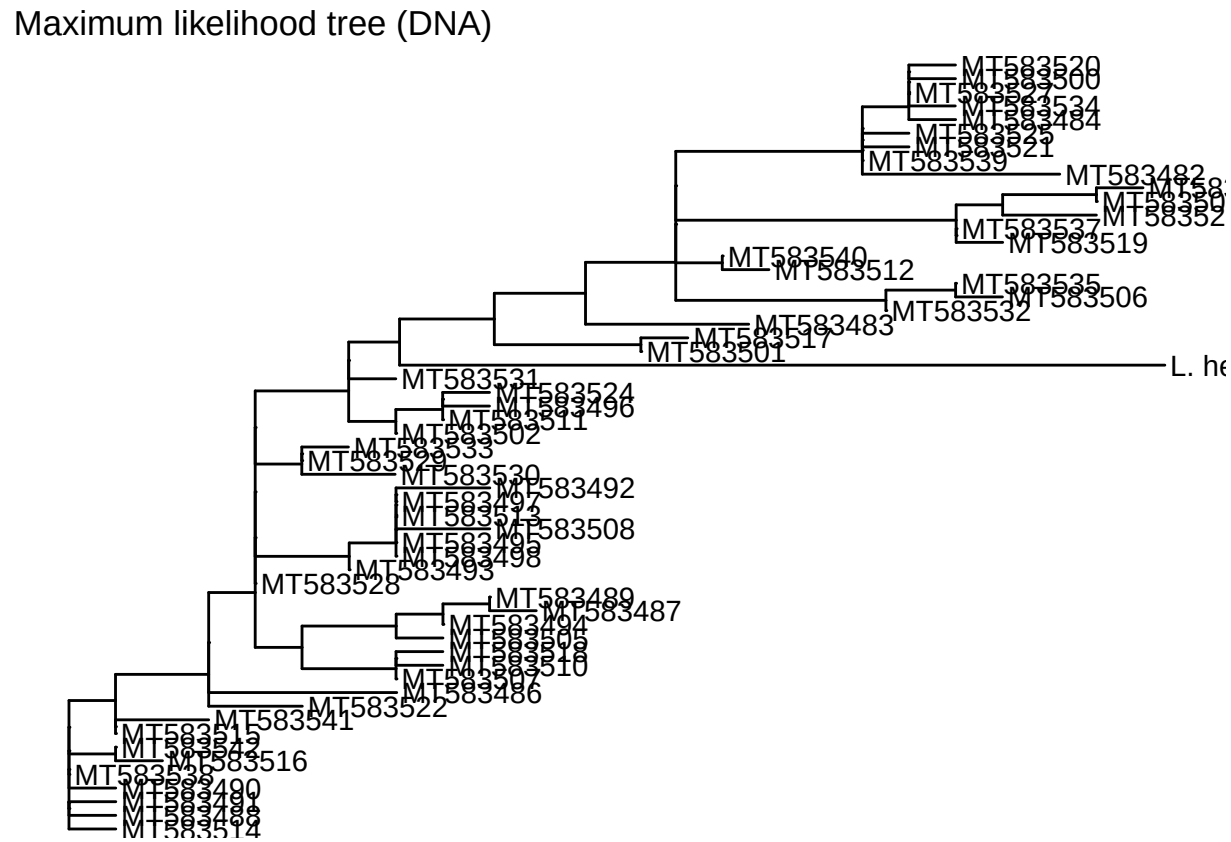
```
#Estimate pairwise distances and construct a starting tree using neighbor joining
# Create initial tree
tree_dna <- NJ(dist.ml(dna_phy))
```

We then apply a model of sequence evolution to calculate the maximum likelihood tree. This model accounts for how different types of nucleotide substitutions occur over time. For DNA data, we use the GTR model (General Time Reversible), a widely used and flexible model that allows different substitution rates for each pair of nucleotides. This helps improve the accuracy of tree estimation, especially when sequences have evolved under complex patterns of change.

```
# Maximum likelihood optimization
fit_dna <- optim.pml(pml(tree_dna, data = dna_phy), model = "GTR")
```

```
## optimize edge weights:  -1546.619 --> -1534.379
## optimize rate matrix:  -1534.379 --> -1468.487
## optimize edge weights:  -1468.487 --> -1467.784
## optimize rate matrix:  -1467.784 --> -1467.784
## optimize edge weights:  -1467.784 --> -1467.784
```

```
# Visualize tree
ggtree(fit_dna$tree) +
geom_tiplab() +
ggtitle("Maximum likelihood tree (DNA)")
```



Section 3 - Adding Bootstrap Support

```
# # Add bootstrap values to tree object
# bs_dna <- bootstrap.pml(fit_dna, bs = 100, optNni = TRUE)
# tree_bs <- plotBS(fit_dna$tree, bs_dna, type = "none")
#
# # Visualize in ggtree
# ggtree(tree_bs) +
#   geom_tiplab() +
#   geom_text2(aes(label = label), hjust = -0.3) +
#   ggtitle("ML tree with bootstrap support (DNA)")
```

Question Answers

a. The resulting phylogeny seems very convoluted. The lack of discernible tips and tip labels makes it difficult to interpret. There are several paraphyletic groups, indicating either high uncertainty in genetic relatedness between individuals or near complete genetic similarity among individuals at the tips.

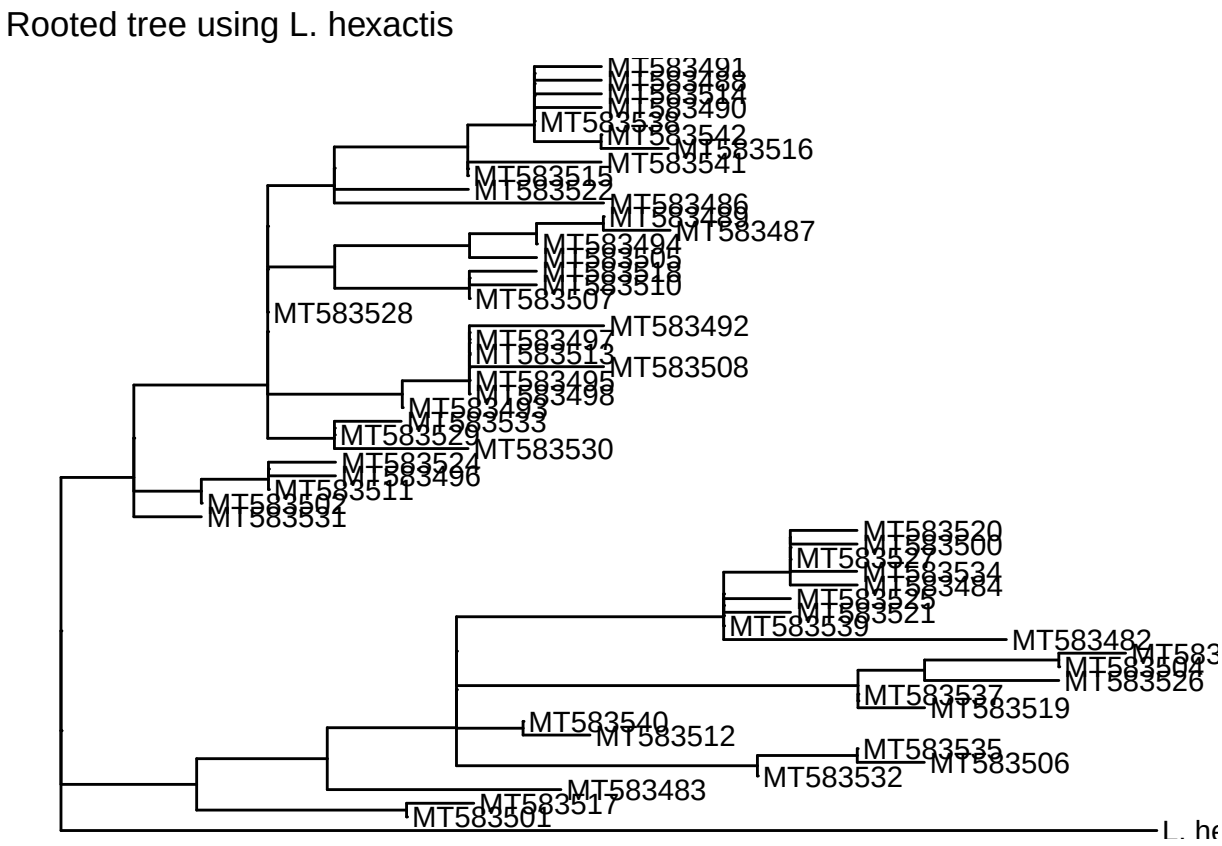
Section 4 - Rooting the Tree Using an Outgroup

```
# View tip labels and identify L. hexactis samples
fit_dna$tree$tip.label
```

```
##  [1] "MT583500"             "MT583520"             "MT583527"
##  [4] "MT583534"             "MT583484"             "MT583521"
##  [7] "MT583525"             "MT583539"             "MT583482"
## [10] "MT583504"             "MT583509"             "MT583526"
## [13] "MT583519"             "MT583537"             "MT583512"
## [16] "MT583540"             "MT583506"             "MT583535"
## [19] "MT583532"             "MT583483"             "MT583531"
## [22] "MT583501"             "MT583517"             "MT583488"
## [25] "MT583491"             "MT583514"             "MT583490"
## [28] "MT583538"             "MT583516"             "MT583542"
## [31] "MT583515"             "MT583541"             "MT583522"
## [34] "MT583486"             "MT583487"             "MT583489"
## [37] "MT583494"             "MT583505"             "MT583510"
## [40] "MT583518"             "MT583507"             "MT583497"
## [43] "MT583513"             "MT583492"             "MT583508"
## [46] "MT583495"             "MT583498"             "MT583493"
## [49] "MT583528"             "MT583529"             "MT583533"
## [52] "MT583530"             "MT583496"             "MT583524"
## [55] "MT583511"             "MT583502"             "L. hexactis AF162095"
```

```
# Re-root the tree on the most recent common ancestor of hexactis tips
rooted_dna <- root(fit_dna$tree, outgroup = "L. hexactis AF162095", resolve.root = TRUE)

# Plot the rooted tree
ggtree(rooted_dna) +
```

```
  geom_tiplab() +
  ggtitle("Rooted tree using L. hexactis")
```



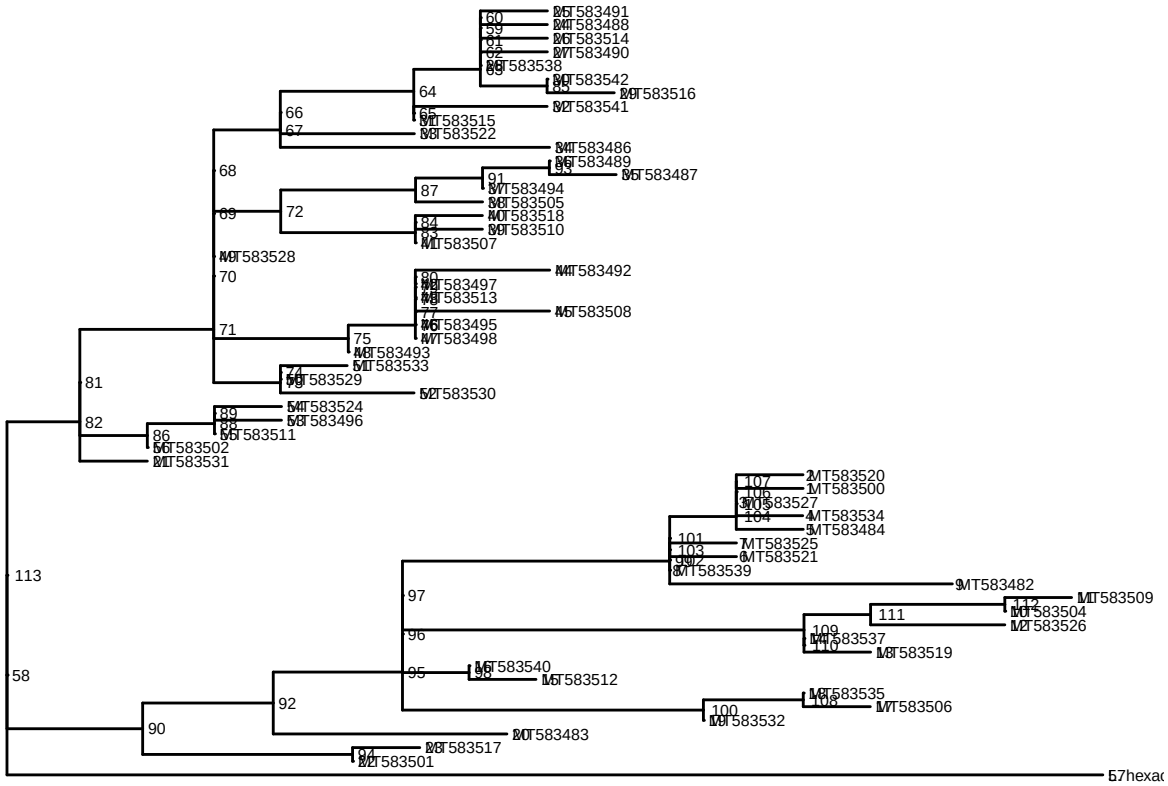
Section 5 - Labeling Clades and Visualizing Structure

Once your tree is rooted, the next step is to explore its structure and annotate according to your research question. Each internal node in the tree represents a hypothetical common ancestor. To label clades, you must first identify internal nodes of interest and determine which tips descend from them.

Start by displaying node numbers on your tree:

```
ggtree(rooted_dna) +
  geom_tiplab(size = 2) +
  geom_text2(aes(label = node), size = 2, hjust = -0.3) +
  ggtitle("Rooted tree with node numbers")
```

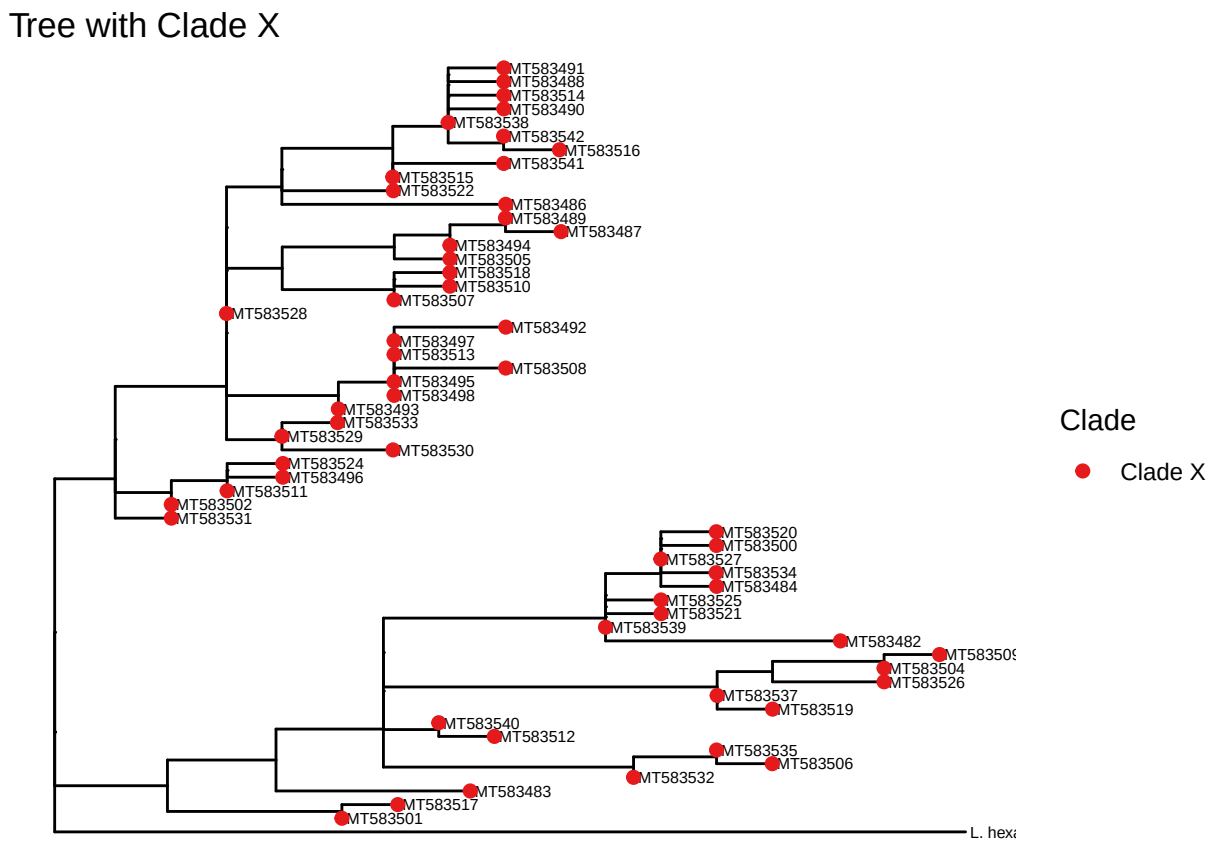
Rooted tree with node numbers



Use these node numbers to define clades. For example, to label all tips descending from node 113 as "Clade X":

```
# Extract tips from node 113
tips_113 <- tree_subset(rooted_dna, node = 113, levels_back = 0)$tip.label

# Create a data frame that assigns clade to each tip
clade_df <- tibble(tip = tips_113, Clade = "Clade X")

# Plot with color
ggtree(rooted_dna) %<+% clade_df +
  geom_tiplab(size = 2) +
  geom_tippoint(aes(color = Clade), size = 2) +
  scale_color_brewer(palette = "Set1", na.translate = FALSE) +
  ggtitle("Tree with Clade X")
```

```
## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_point_g_gtree()`).
```



Challenge 1: Identify the 3 major clades in the tree. Give them each a name (e.g. Clade A, B, C) and annotate them on the tree. Are the groups well-supported?

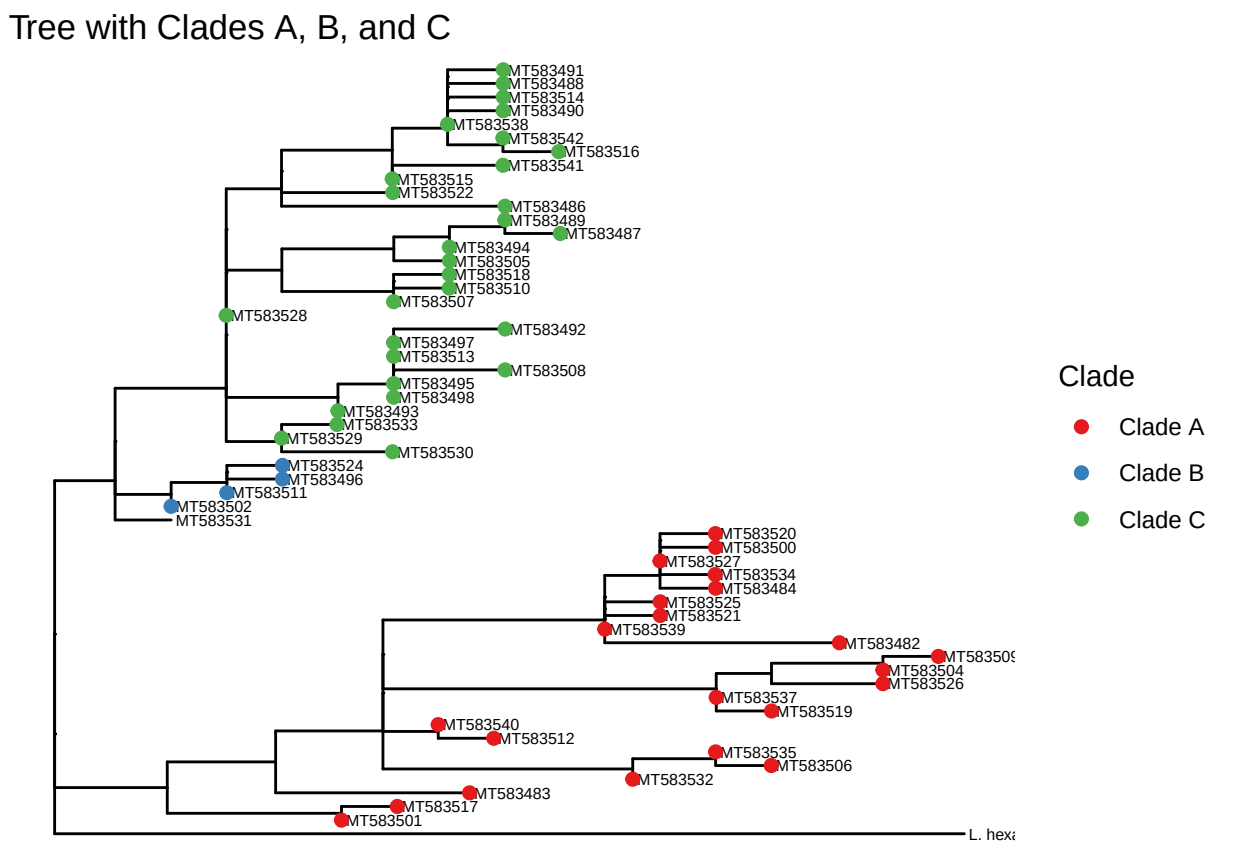
```
# Extract tips from node 58
tips_90 <- tree_subset(rooted_dna, node = 90, levels_back = 0)$tip.label
tips_86 <- tree_subset(rooted_dna, node = 86, levels_back = 0)$tip.label
tips_71 <- tree_subset(rooted_dna, node = 71, levels_back = 0)$tip.label

# Create a data frame that assigns clade to each tip
clade_df_A<- tibble(tip = tips_90, Clade = "Clade A")
clade_df_B<- tibble(tip = tips_86, Clade = "Clade B")
clade_df_C<- tibble(tip = tips_71, Clade = "Clade C")

clade_df_ABC<- rbind(clade_df_A, clade_df_B, clade_df_C)
str(clade_df_ABC)
```

```
## tibble [55 x 2] (S3: tbl_df/tbl/data.frame)
##  $ tip  : chr [1:55] "MT583500" "MT583520" "MT583527" "MT583534" ...
##  $ Clade: chr [1:55] "Clade A" "Clade A" "Clade A" "Clade A" ...
```

```
# Plot with color
ggtree(rooted_dna) %<+% clade_df_ABC +
  geom_tiplab(size = 2) +
```

```
  geom_tippoint(aes(color = Clade), size = 2) +
  scale_color_brewer(palette = "Set1", na.translate = FALSE) +
  ggtitle("Tree with Clades A, B, and C")
```

```
## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point_g_gtree()`).
```



The groups are relatively well supported because they include a common ancestors and all descendants in a clear arrangement, as a clade should by definition.

Challenge 2: Does the protein sequence data show the same pattern? Hint: Repeat steps above to build a rooted, labeled, bootstrap-annotated tree with protein sequence data and compare tree topology.

```
# Load sequences
aa <- readAAStringSet("coi_protein.fasta")

# Align sequences
aa_aln <- msa(aa, method = "ClustalW")
```

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

```r
# Convert aligned sequences to phyDat format
aa_phy <- as.phyDat(msaConvert(aa_aln, type = "seqinr"), type = "AA")

# View structure in the console
aa_phy
```

```
## 51 sequences with 155 character and 57 different site patterns.
## The states are A R N D C Q E G H I L K M F P S T W Y V
```

```r
#Estimate pairwise distances and construct a starting tree using neighbor joining
# Create initial tree
tree_aa <- NJ(dist.ml(aa_phy))
```

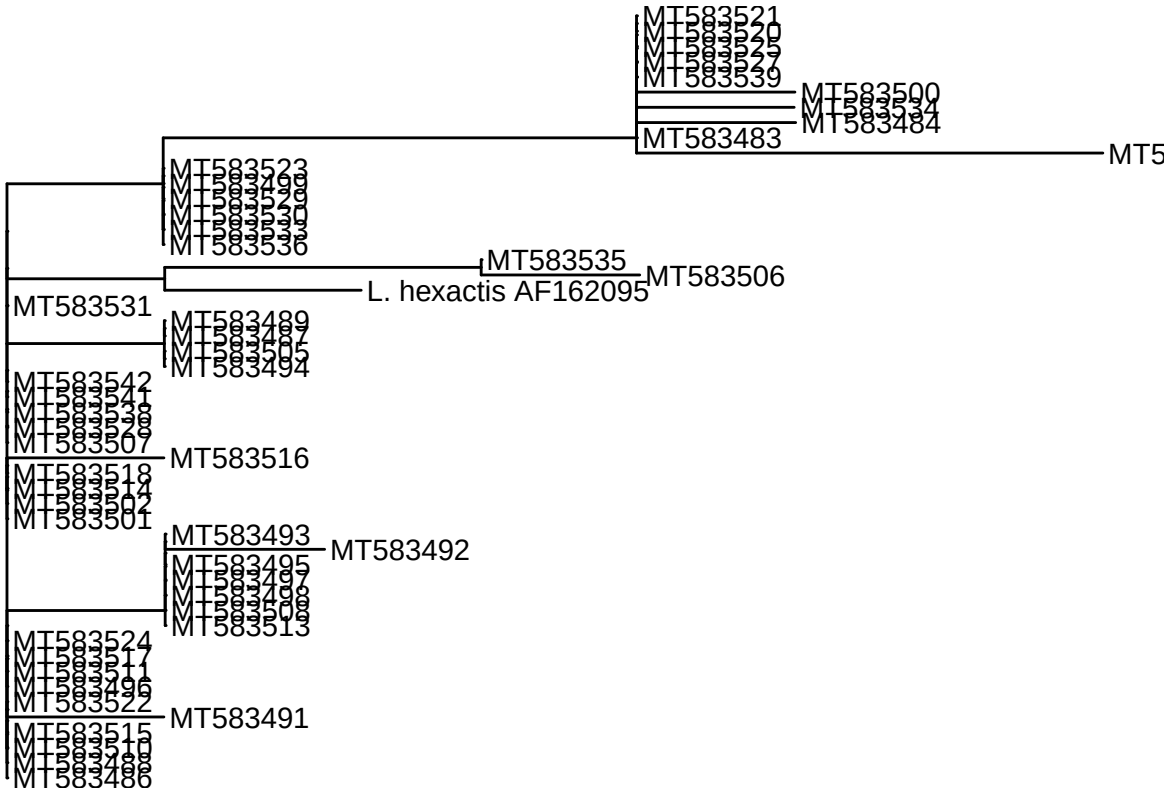
We then apply a model of sequence evolution to calculate the maximum likelihood tree. This model accounts for how different types of nucleotide substitutions occur over time. For DNA data, we use the GTR model (General Time Reversible), a widely used and flexible model that allows different substitution rates for each pair of nucleotides. This helps improve the accuracy of tree estimation, especially when sequences have evolved under complex patterns of change.

```r
# Maximum likelihood optimization
fit_aa <- optim.pml(pml(tree_aa, data = aa_phy), model = "WAG")
```

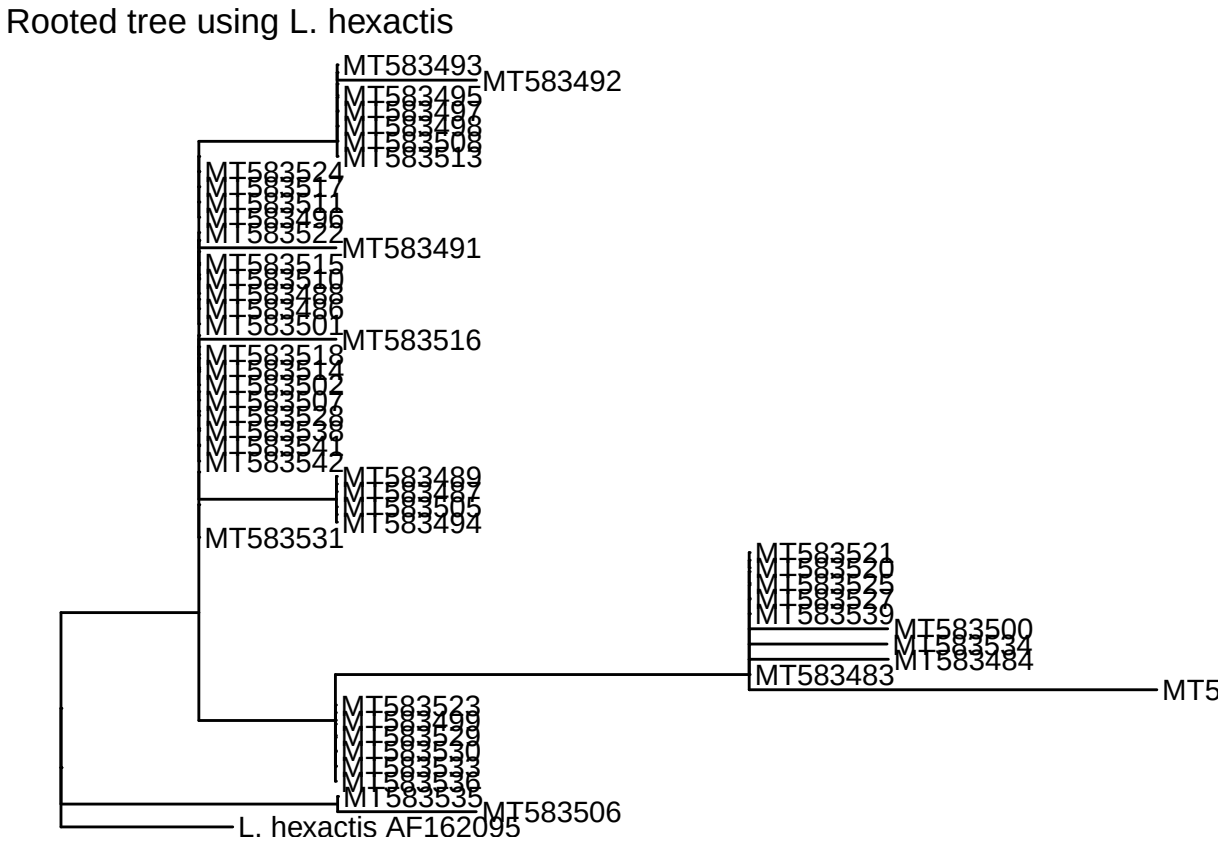
```
## optimize edge weights:  -602.3613 --> -599.8747
## optimize edge weights:  -599.8747 --> -599.8747
```

```r
# Visualize tree
ggtree(fit_aa$tree) +
geom_tiplab() +
ggtitle("Maximum likelihood tree (AA)")
```

Maximum likelihood tree (AA)

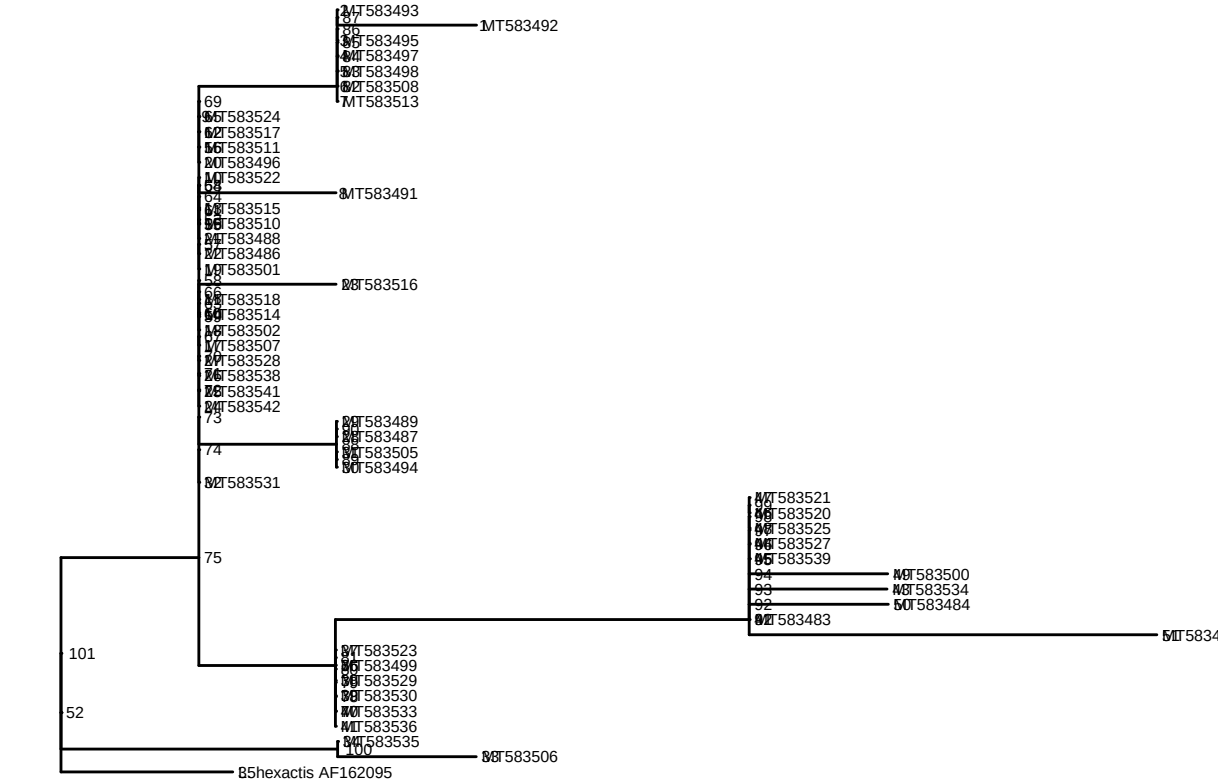



```
# Re-root the tree on the most recent common ancestor of hexactis tips
rooted_aa <- root(fit_aa$tree, outgroup = "L. hexactis AF162095", resolve.root = TRUE)

# Plot the rooted tree
ggtree(rooted_aa) +
  geom_tiplab() +
  ggtitle("Rooted tree using L. hexactis")
```




```
# # Add bootstrap values to tree object
# bs_aa <- bootstrap.pml(fit_aa, bs = 100, optNni = TRUE)
# tree_bsaa <- plotBS(fit_aa$tree, bs_aa, type = "none")
#
# # Visualize in ggtree
# ggtree(tree_bsaa) +
#   geom_tiplab() +
#   geom_text2(aes(label = label), hjust = -0.3) +
#   ggtitle("ML tree with bootstrap support (AA)")
```

```
ggtree(rooted_aa) +
  geom_tiplab(size = 2) +
  geom_text2(aes(label = node), size = 2, hjust = -0.3) +
  ggtitle("Rooted tree with node numbers")
```



The protein sequences do not show the same patterns. There are much more unresolved lineages in comparison to the DNA tree, which indicates similarity among individuals regarding their encoded proteins despite differences in nucleotide sequences.

Discussion Question Answers

c. To test whether genetic structure reflects geographic isolation, I would create a matrix pairing genetic distance with geographic distance and perform a Mantel test.

d. This analysis does give some information about the potential for cryptic species in the *Leptasterias* genus, especially given the functional similarity of the encoded protein. However, much more information (behavioral, ecological, ontogenetic, etc.) that could potentially point to differences in species is necessary to make a call on the presence of cryptic species.