Exemplars

@rvosa

2024-04-02

Introduction

In this analysis we assess which method of exemplar picking produces branch lengths that hew closest to those estimated freely under the same model (GTRGAMMA). The methods of exemplar picking are:

- *tallest*: here, the exemplars are the tallest tips on either side of the root of the subtree.
- *shortest*: the exemplars are the shallowest tips on either side.
- *median*: the tips closest to the median root-to-tip path length on either side.

To assess which method works best, we compare the root-to-tip path lengths on the grafted tree with freely estimated branch lengths on the same topology. Hence, we have three pairs of trees. In the following code block, we have a function that takes such a pair of trees, computes the path lengths, and does a linear regression. Hence, the criterion becomes: which method maximizes the R^2?

```
library(ape)
library(adephylo)
```

```
## Loading required package: ade4
```

```
library(ggplot2)

regress_grafted_estimated <- function(grafted, estimated, title) {

  # Calculate all root-to-tip distances with adephylo
  dists_grafted <- distRoot(grafted)
  dists_estimated <- distRoot(estimated)

  # Remove quotes in tip labels, if there are any
  names(dists_grafted) <- gsub("'", "", names(dists_grafted))
  names(dists_estimated) <- gsub("'", "", names(dists_estimated))

  # Merge into a single df
  df1 <- data.frame(Name = names(dists_grafted), Grafted = dists_grafted)
  df2 <- data.frame(Name = names(dists_estimated), Estimated = dists_estimated)
  combined_df <- merge(df1, df2, by = "Name", all = TRUE)

  # Remove misaligned values. These are in here because when re-estimating
  # the branch lengths we didn't run the revcom check that the pipeline does,
  # so some tips have erroneous values. It'd be better if these weren't there
  # at all but for now we remove them after the fact.
  df_cleaned <- subset(combined_df, Estimated <= 0.7)

  # Assign to x/y for simplicity in the rest of the code.
  x <- df_cleaned$Grafted
```

```
  y <- df_cleaned$Estimated

  # Do a linear regression to get the R^2
  model <- lm(x ~ y)
  r_squared <- summary(model)$r.squared

  # Create the plot
  ggplot() +
    geom_point(aes(x = x, y = y), color = "blue") + # Scatter plot
    geom_smooth(aes(x = x, y = y), method = "lm", color = "red", se = FALSE) + # Regression line
    annotate("text", x = max(x), y = min(y), label = paste("R^2 =", round(r_squared, 2)), hjust = 1, vj
    labs(title = title,
         x = "Grafted",
         y = "Estimated") +
    theme_minimal()
}
```

Tallest

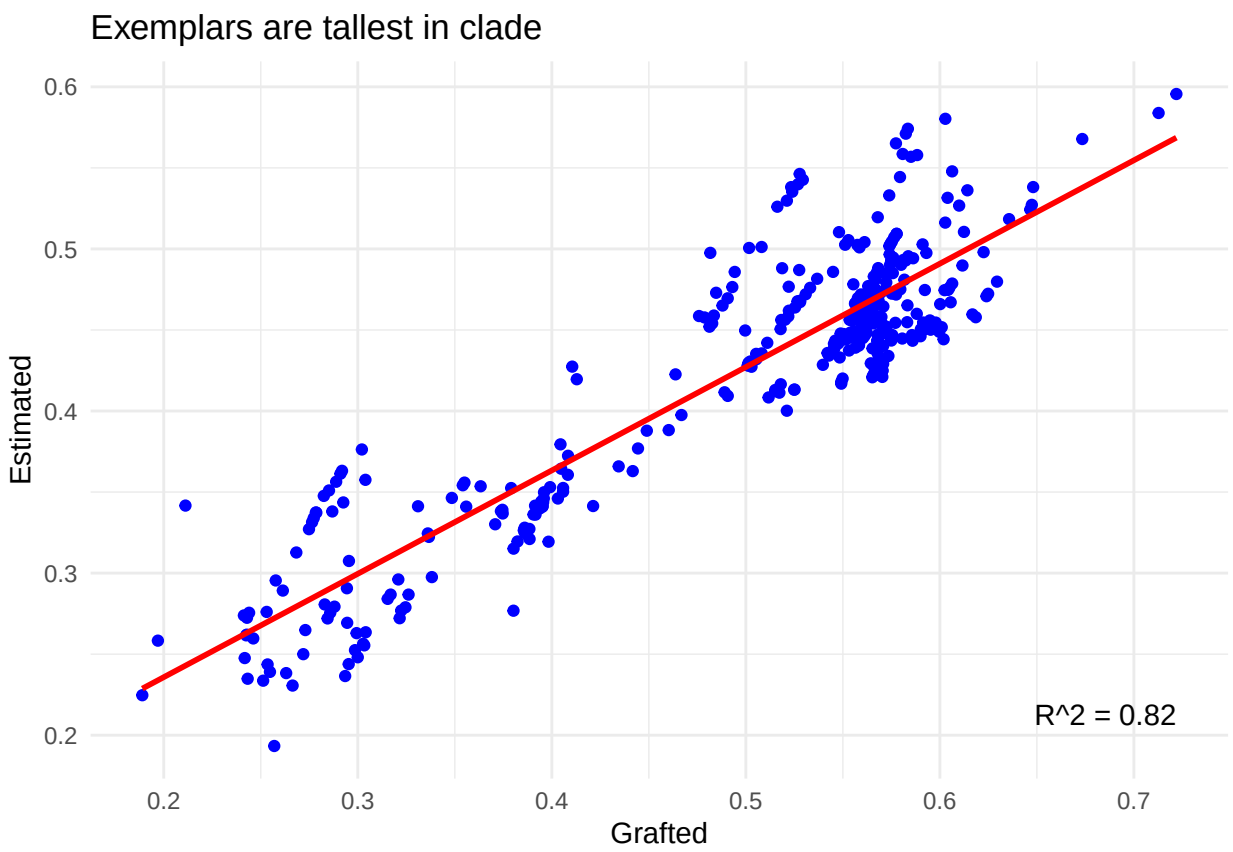
Here we do the calculation for the case where the exemplars are the tallest in the subtree. Spoiler alert: this is the best method, though not by that much. This is not surprising (I expected it) because the shorter the exemplars, the more they are affected by only a few columns in the alignment, which gives a kind of stochasticity that propagates through the rest of the subtree as it is rescaled. Or, conversely, the tallest exemplars get closest to the Central Limit Theorem or whatever. I mean, ask a statistician how to phrase this correctly but it makes intuitive sense. (Also, I wrote a book chapter about a related issue ages ago where constraining the longest path gave the least variance in subtree branch lengths.)

```
# Read trees
tallest_grafted <- read.tree(file="grafted-tallest.tre")
tallest_estimated <- read.tree(file="RAxML_result.grafted-tallest")
regress_grafted_estimated(tallest_grafted, tallest_estimated, 'Exemplars are tallest in clade')
```

```
## Warning in checkTree(object): Labels are not unique.

## Warning in checkTree(object): Labels are not unique.

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



Shallowest

This method does the poorest. See under Tallest why that is.

```
# Read trees
tallest_grafted <- read.tree(file="grafted-shortest.tre")
tallest_estimated <- read.tree(file="RAxML_result.grafted-shortest")
regress_grafted_estimated(tallest_grafted, tallest_estimated, 'Exemplars are shortest in clade')
```

```
## Warning in checkTree(object): Labels are not unique.

## Warning in checkTree(object): Labels are not unique.

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

Median

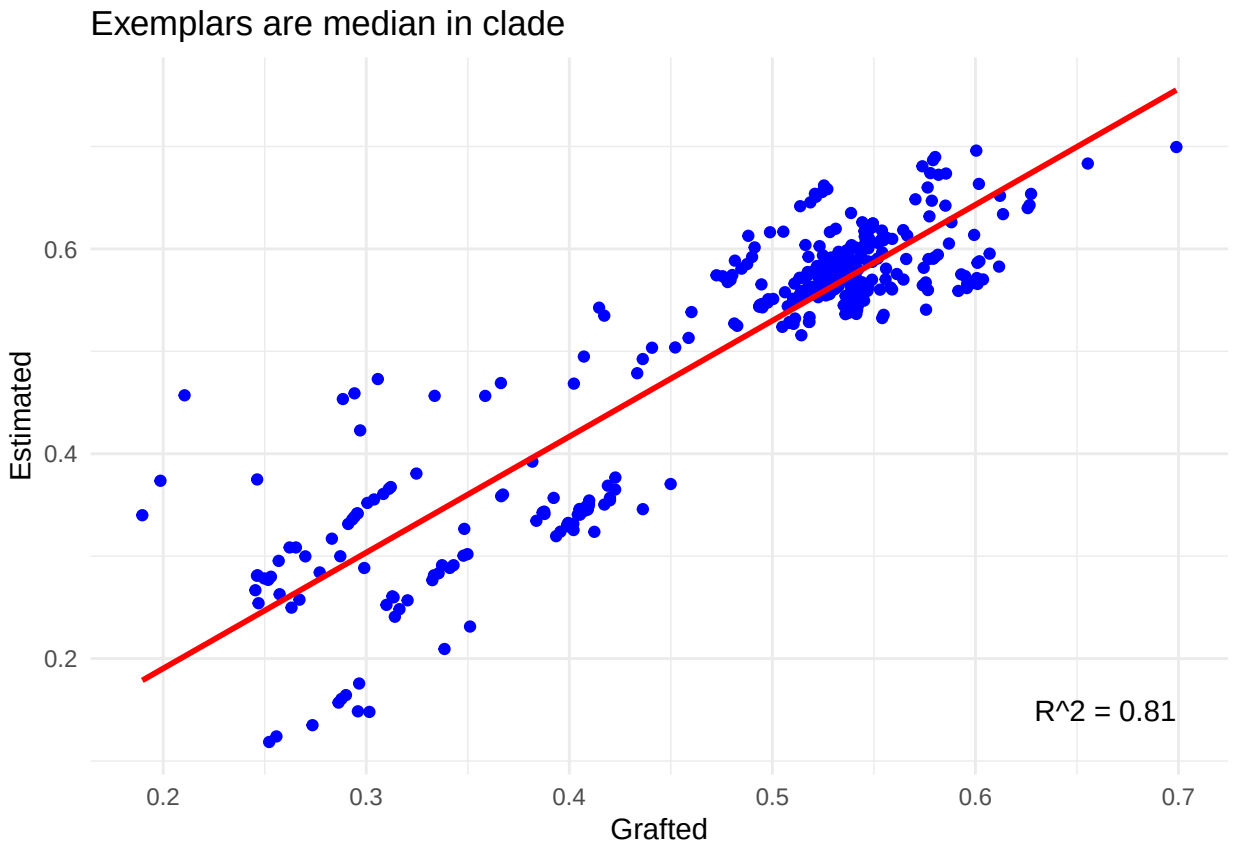
This one is in the middle. If you get the other two, you get this one.

```
# Read trees
tallest_grafted <- read.tree(file="grafted-median.tre")
tallest_estimated <- read.tree(file="RAxML_result.grafted-median")
regress_grafted_estimated(tallest_grafted, tallest_estimated, 'Exemplars are median in clade')
```

```
## Warning in checkTree(object): Labels are not unique.

## Warning in checkTree(object): Labels are not unique.

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


Exemplars are median in clade
Estimated
0.6
0.4
0.2
0.2
0.3
0.4
0.5
0.6
0.7
Grafted
R^2 = 0.81