

plot bootstraps

laura dijkhuizen

04/08/2020

This is a late afternoon inquiry to see if plotting distributions of support values in trees is somewhat insightful perhaps.

create simple tab files with bootstrap values

```
for t in analyses/*_trees/*/*.treefile
do out=$(echo $t | sed 's/.treefile/.bootstraptab/')
  if [ ! -f $out ]
  then egrep -o '[0-9./]+' $t | tr -d ')' | tr '/' '\t' > $out
  fi
done
```

optionally, inspect your files for troubleshooting etc.

```
#for t in analyses/*_trees/*/*.bootstraptab
#do head $t
#done
```

Read all tab files

now simplify these names a bit

Note that theecho = FALSE' parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```
metatabs <- strsplit(x = files,split = '/')
rm(combitab)
```

```
## Warning in rm(combitab): object 'combitab' not found
```

```
combitab <- data.table()
for (i in 1:length(tabs)) {
  temp <- tabs[[i]]
  if (dim(temp)[2] == 1) {
    names(temp) <- 'nonparametricBootstrap'
    temp$SHaLRT <- NA
    temp$UFBootstrap <- NA
  }
  if (length(names(temp)) == 2) {
    names(temp) <- c('SHaLRT','UFBootstrap')
    temp$nonparametricBootstrap <- NA
  }
}
```

```

temp$dataset <- factor(metatabs[[i]][2])
temp$alignment <- as.character(metatabs[[i]][3])
temp$iqtree <- factor(metatabs[[i]][4])

combitab <- rbind(combitab,temp,fill=T)
rm(temp)
}
combitab$nonparametricBootstrap <- as.numeric(combitab$nonparametricBootstrap)
combitab$SHaLRT <- as.numeric(combitab$SHaLRT)
combitab$UFBootstrap <- as.numeric(combitab$UFBootstrap)
summary(combitab)

```

```

## nonparametricBootstrap      SHaLRT      UFBootstrap
## Min. : 4.00      Min. : 0.00      Min. : 11.00
## 1st Qu.: 40.00      1st Qu.: 71.25      1st Qu.: 73.00
## Median : 67.00      Median : 86.40      Median : 93.00
## Mean : 63.70      Mean : 77.81      Mean : 84.15
## 3rd Qu.: 90.75      3rd Qu.: 94.65      3rd Qu.: 99.00
## Max. :100.00      Max. :100.00      Max. :100.00
## NA's :599      NA's :374      NA's :374
##
##                                dataset
## combi_sequences_linear_trees      : 88
## combi-I-to-VIII-Azfi_sequences_linear_trees      :437
## combi-I-to-VIII-Azfi-Arabidopsis_sequences_linear_trees:396
## combi-VI-VII-Azfi_suspects_trees      : 52
##
##
## alignment
## Length:973
## Class :character
## Mode :character
##
##
##
## combi-I-to-VIII-Azfi-Arabidopsis_sequences_linear_aligned-mafft-einsi_trim-gt4_iqtree-b200.bootstraps
## combi-I-to-VIII-Azfi-Arabidopsis_sequences_linear_aligned-mafft-einsi_trim-gt4_iqtree-bb2000-alrt2000
## combi-I-to-VIII-Azfi-Arabidopsis_sequences_linear_aligned-mafft-einsi_trim-gt5_iqtree-b200.bootstraps
## combi-I-to-VIII-Azfi-Arabidopsis_sequences_linear_aligned-mafft-einsi_trim-gt5_iqtree-bb2000-alrt2000
## combi-I-to-VIII-Azfi-Arabidopsis_sequences_linear_aligned-mafft-einsi_trim-gt6_iqtree-b200.bootstraps
## combi-I-to-VIII-Azfi-Arabidopsis_sequences_linear_aligned-mafft-einsi_trim-gt6_iqtree-bb2000-alrt2000
## (Other)

```

```

rm(metatab,tabs,i,files
)

```

```
## Warning in rm(metatab, tabs, i, files): object 'metatab' not found
```

melt dataframe to long format

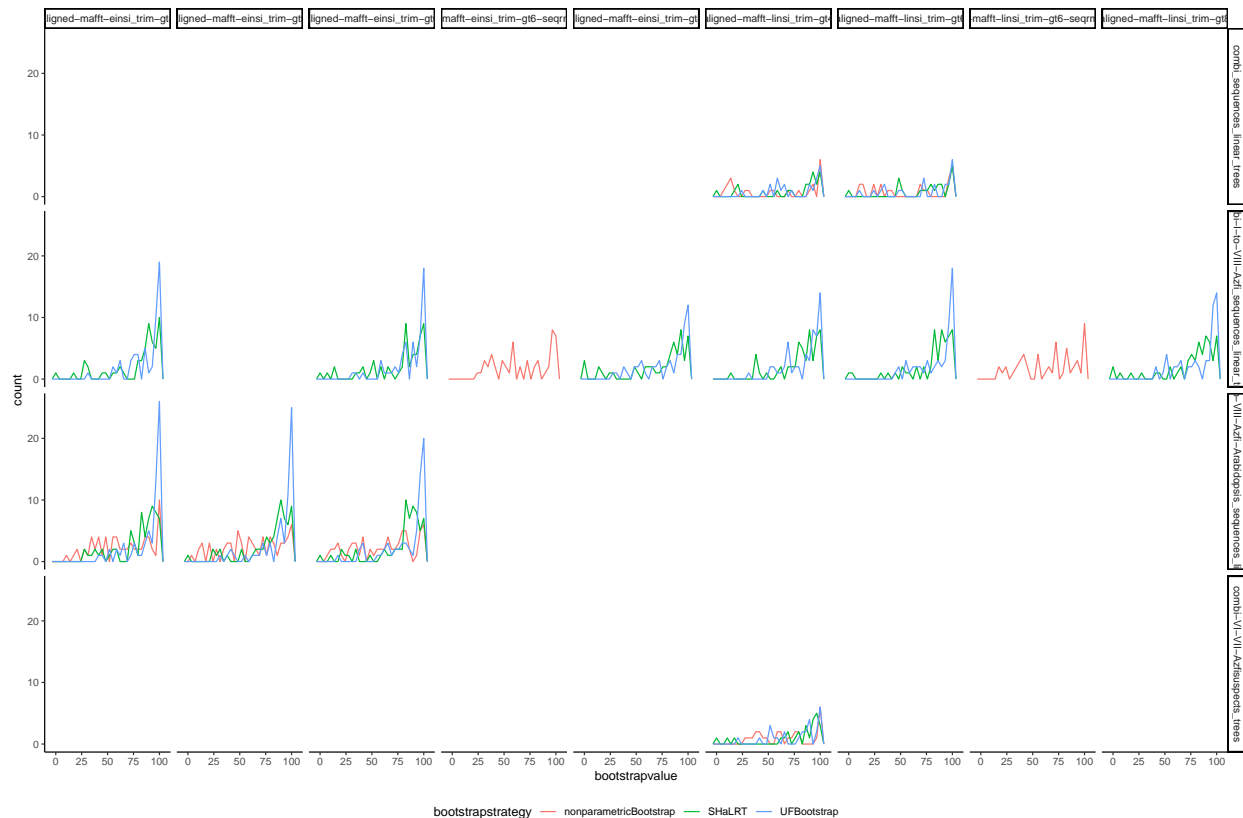
```
combitab <- melt(combitab,variable.name = 'bootstrapstrategy',id.vars = c('dataset','alignment','iqtree'
```

```
#plot
```

```
library(ggplot2)
attach(combitab)
plot <- ggplot(data=combitab,mapping = aes(x=bootstrapvalue,col=bootstrapstrategy))
plot <- plot + geom_freqpoly()
plot <- plot + theme_classic()
plot <- plot + facet_grid(dataset~alignment)
plot <- plot + theme(legend.position = 'bottom')
plot
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 1347 rows containing non-finite values (stat_bin).
```



```
rm(plot)
```

```
library(ggplot2)
attach(combitab)
```

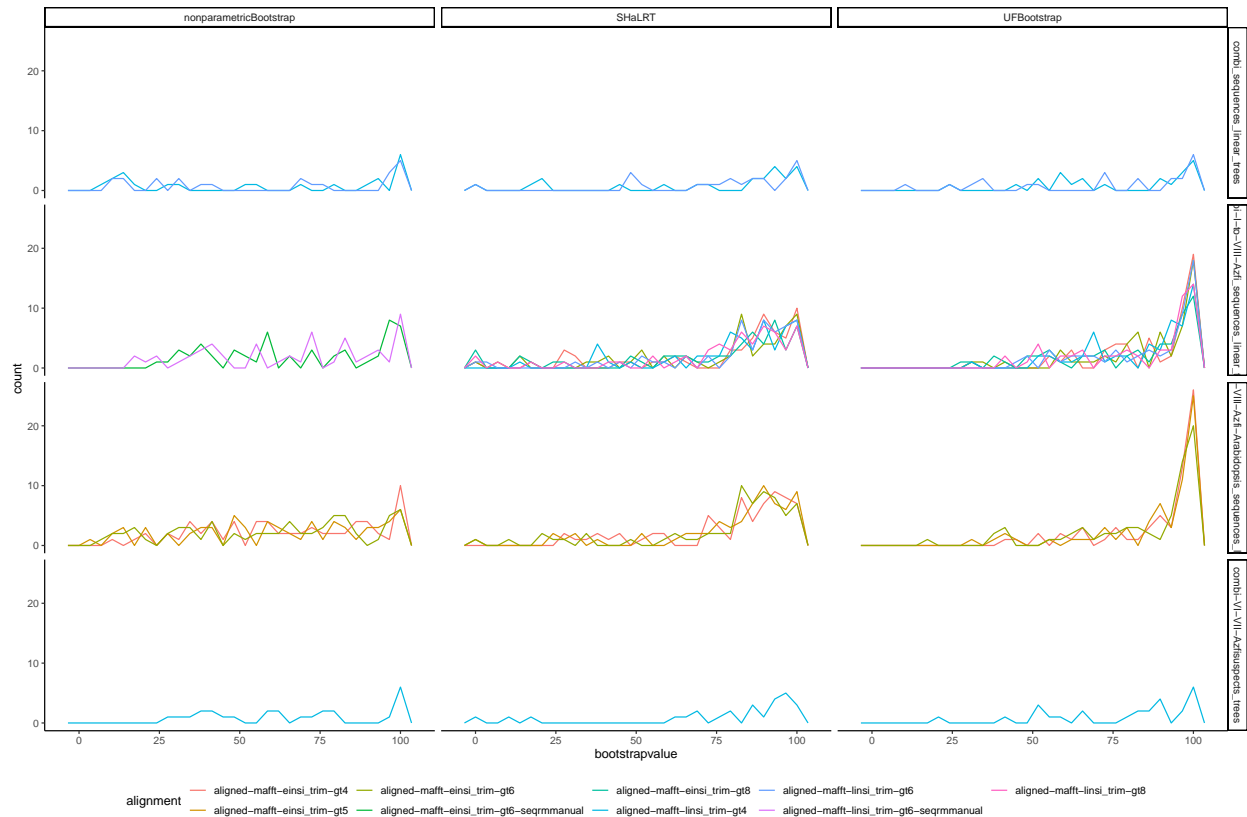
```
## The following objects are masked from combitab (pos = 3):
```

```
##
```

```
## alignment, bootstrapstrategy, bootstrapvalue, dataset, iqtree
```

```
plot <- ggplot(data=combitab,mapping = aes(x=bootstrapvalue,col=alignment))
plot <- plot + geom_freqpoly()
plot <- plot + theme_classic()
plot <- plot + facet_grid(dataset~bootstrapstrategy)
plot <- plot + theme(legend.position = 'bottom')
plot
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 1347 rows containing non-finite values (stat_bin).
```



```
rm(plot)
```

```
library(ggplot2)
attach(combitab)
```

```
## The following objects are masked from combitab (pos = 3):
```

```
##
```

```
## alignment, bootstrapstrategy, bootstrapvalue, dataset, iqtree
```

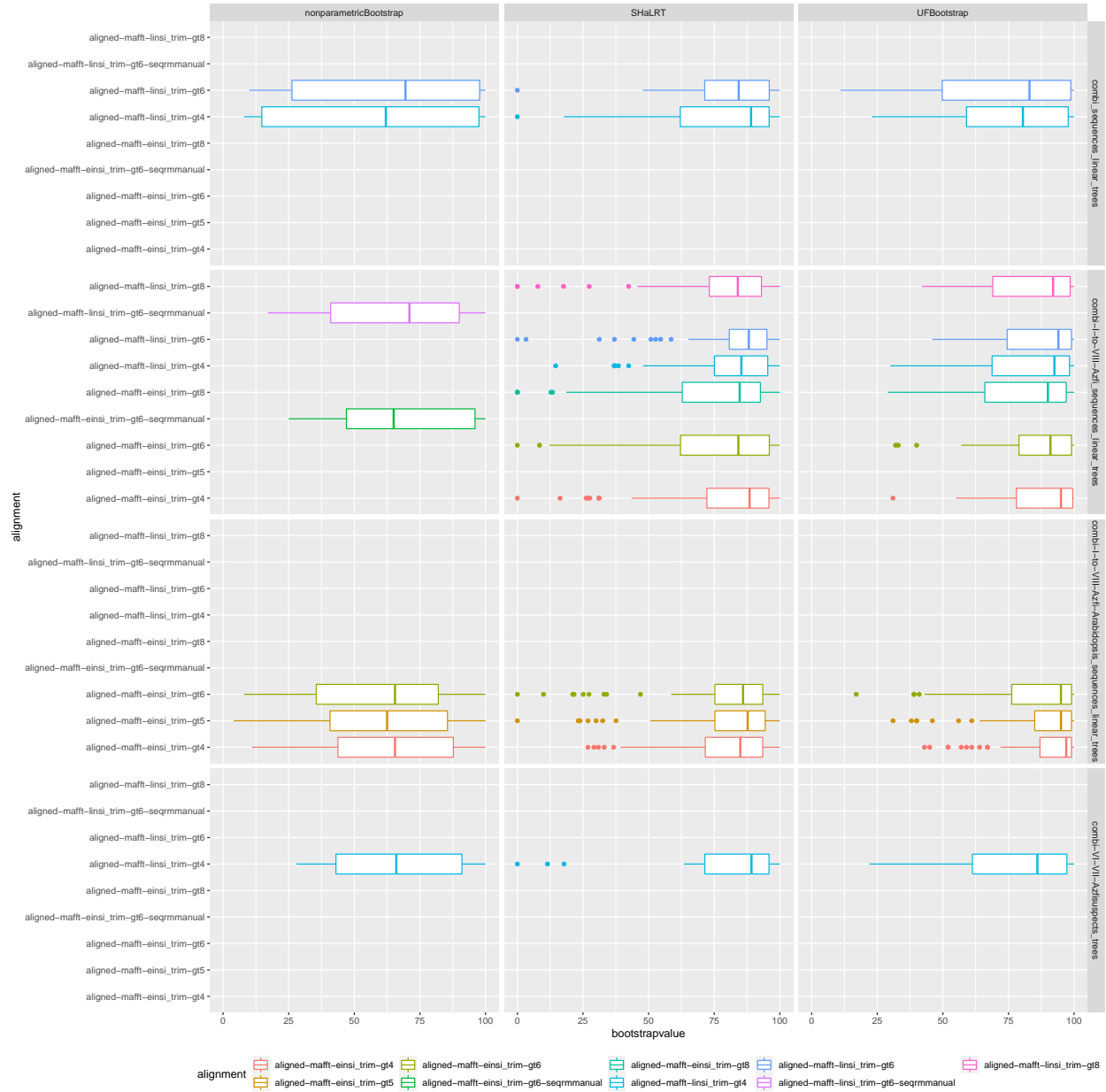
```
## The following objects are masked from combitab (pos = 4):
```

```
##
```

```
## alignment, bootstrapstrategy, bootstrapvalue, dataset, iqtree
```

```
plot <- ggplot(data=combitab, mapping = aes(x=bootstrapvalue, y=alignment, col=alignment))
plot <- plot + geom_boxplot()
#plot <- plot + theme_classic()
plot <- plot + facet_grid(dataset~bootstrapstrategy)
plot <- plot + theme(legend.position = 'bottom')
plot
```

```
## Warning: Removed 1347 rows containing non-finite values (stat_boxplot).
```



`rm(plot)`

All and all, plotting support values like this is not very insightful. I'm quite surprised by how similar support distributions are for the different aligning strategies actually. Still one can make some funny observations. For example the optimal trimming percentage (very naïve interpretation of optimal) differs for the different alignment strategies, but not for the two quick bootstrap methods. Perhaps the bootstrap is artificially high for leniently trimmed alignments. It seems that the UF bootstrap correlates better with the nonparametric bootstrap, at least, if you're trying to pick an alignment strategy and use this naïve method of optimising support in your tree (which you shouldn't, at least not without inspecting the trees and the alignments as well.).