

Chart Combinations

Anamaria Crisan

This document demonstrates how to combine and plot different chart types using minCombinR. This document assumes that you have already run the “Getting started with minCombinR” and the “Simple Chart Types” and that the necessary data has already been loaded into your R workspace.

Loading data

```
devtools::load_all() # TODO: temporary once things are done
library(dplyr)
library(ggtree)

# Table data
tab_dat <- input_data(file = system.file("extdata", "eov_metadata.csv", package = "mincombinr"),
                      dataType = "table")

# Tree data
tree_dat <- input_data(file = system.file("extdata", "eov_tree.nwk", package = "mincombinr"),
                      dataType = "tree")

# Genomic data
genomic_dat <- input_data(file = system.file("extdata", "eov_GIN_genomic_FIXED.fasta", package = "mincombinr"),
                        dataType = "dna")

# Shape files
# Shape files require that .shp, .shx, and .prj files at a minimum to be in the same directory.
# If you would like to add metadata to the shape file, you can also add .dbf files.
gin_shape_dat <- input_data(file = system.file("extdata", "gin_admbnda_adm1_ocha_itos.shp", package = "mincombinr"),
                           dataType = "spatial")

lbr_shape_dat <- input_data(file = system.file("extdata", "lbr_admbnda_adm1_ocha.shp", package = "mincombinr"),
                           dataType = "spatial")

sle_shape_dat <- input_data(file = system.file("extdata", "sle_admbnda_adm1_lm_gov_ocha_20161017.shp",
                                             package = "mincombinr"),
                           dataType = "spatial")
```

Chart Combinations

First, let's store the specifications for a few charts that we want to combine together

```
# Bar chart
bar_chart <- specify_single(chart_type = "bar", data = "tab_dat", x = "country")

# Phylogenetic Tree
phyloTree_chart <- specify_single(chart_type = "phylogenetic tree", data = "tree_dat")

# Scatter Plot
scatter_chart <- specify_single(chart_type = "scatter",
```

```

data = "tab_dat",
x = "latitude",
y = "longitude",
title = "Ebola Scatter Plot")

# Geographic Map
map_chart <- specify_single("geographic map", data = "tab_dat", lat = "latitude", long = "longitude")

```

Unaligned

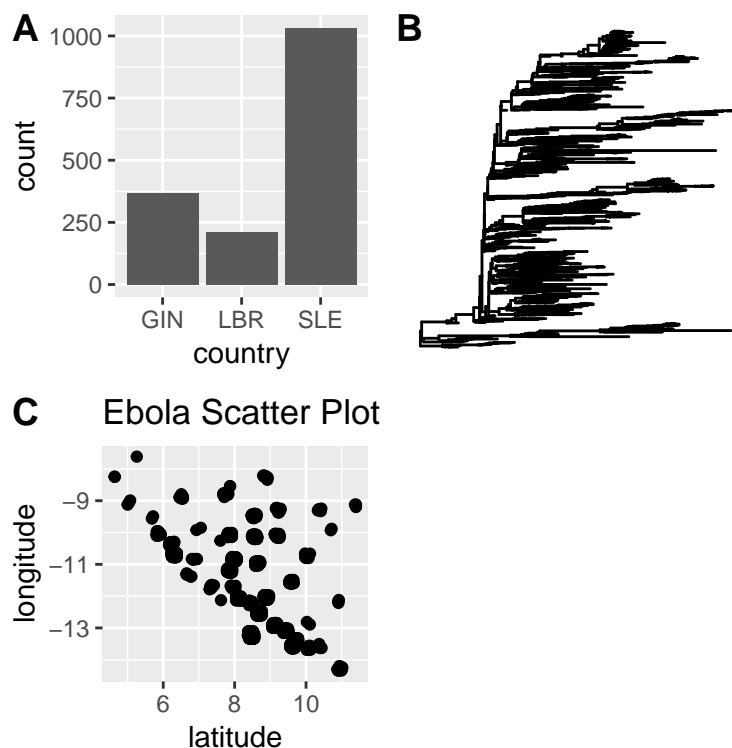
Unaligned combinations can be used when you just want to put a bunch of plots together and there are no spatial or visual linkages between the plots themselves.

```

# Specify that you want to combine the bar_chart, phyloTree_chart and scatter_chart
mg_combo <- specify_combination(combo_type = "unaligned",
                                base_charts = c("bar_chart", "phyloTree_chart", "scatter_chart"))

# Now plot it!
plot(mg_combo)

```



Small Multiples

Small multiple charts are visually linked because they show the same underlying chart type while showing different subsets of the data. Another common name for this is facets.

```

# Specify the base chart type with all of the data that you wish to use:
scatter_chart <- specify_single(chart_type = "scatter", data = "tab_dat", x = "latitude", y = "longitude")

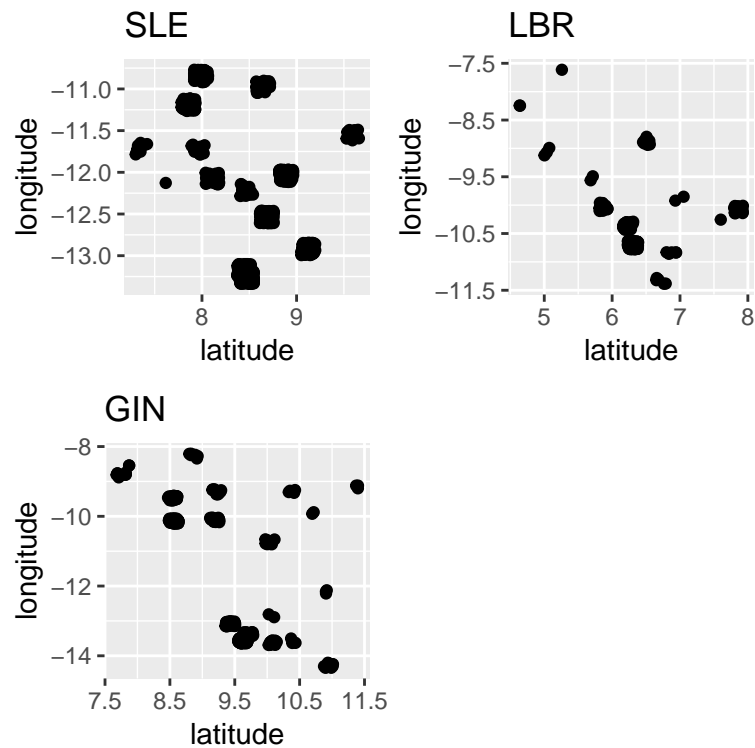
# Now specify the small multiple combination
sm_combo_scatter <- specify_combination(combo_type = "small_multiple",
                                         base_charts = "scatter_chart",

```

```

                                facet_by = "country")
plot(sm_combo_scatter)

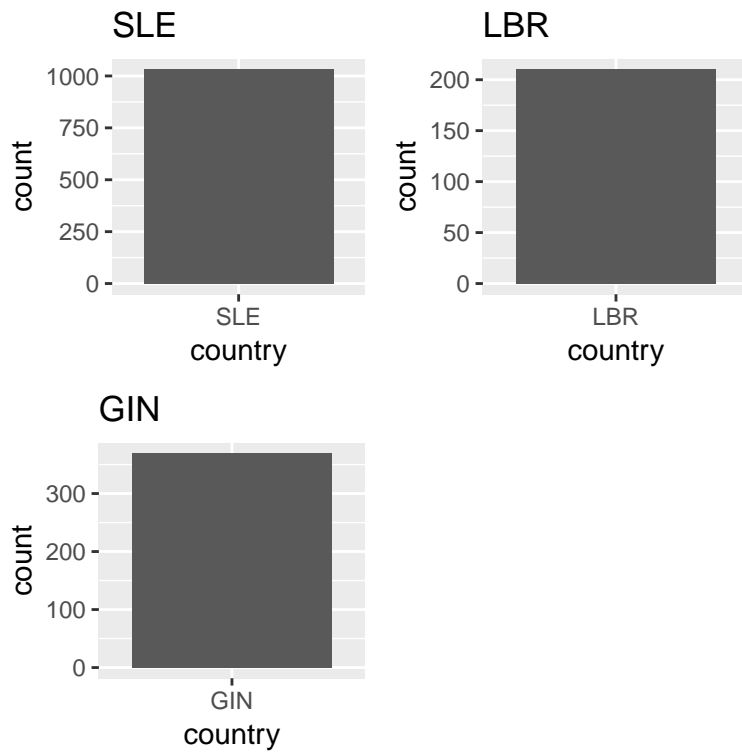
```



```

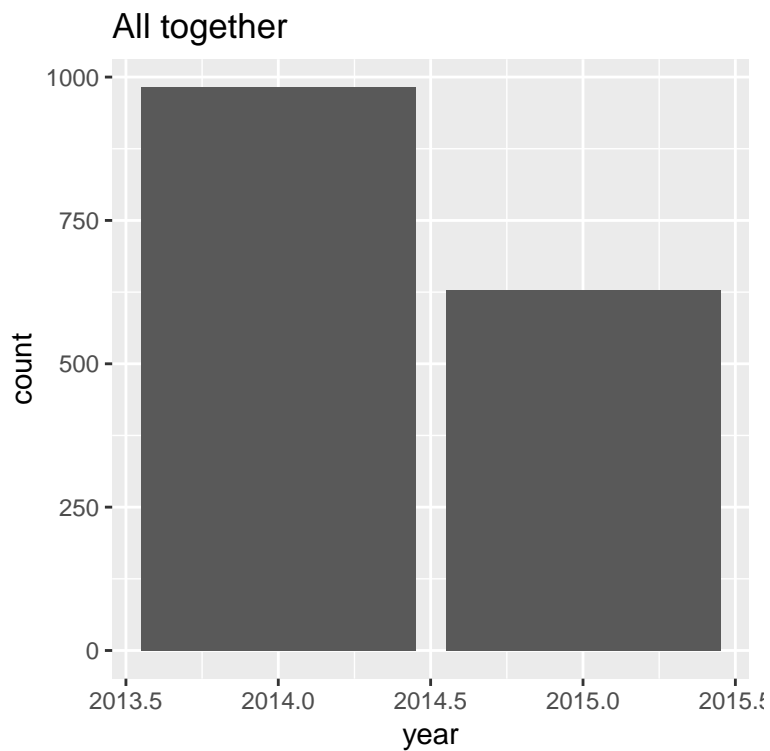
sm_combo_bar <- specify_combination(combo_type = "small_multiple",
                                     base_charts = "bar_chart",
                                     facet_by = "country")
plot(sm_combo_bar)

```

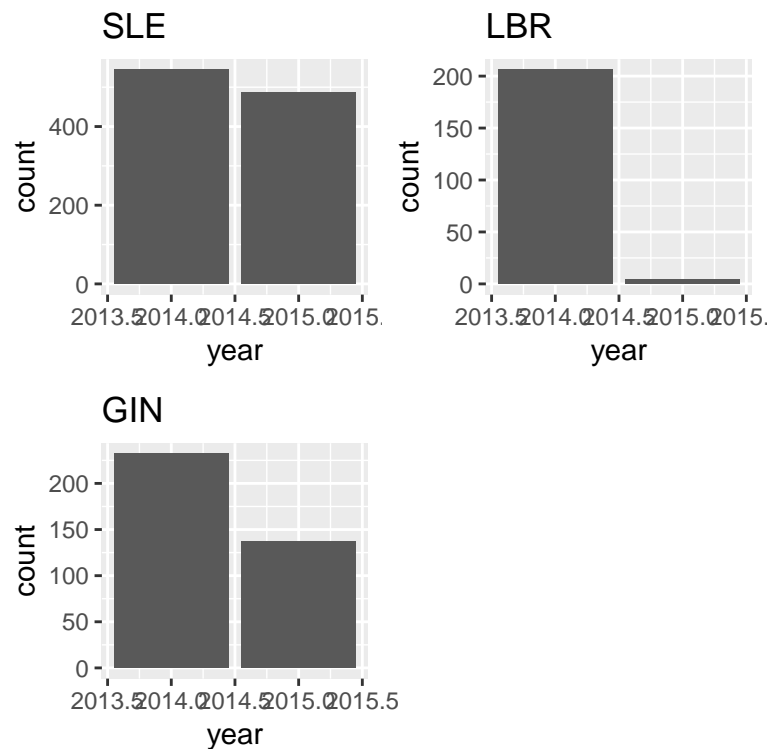


```
# Let's try a more interesting bar_chart small multiple
bar_chart_alt <- specify_single(chart_type = "bar", data = "tab_dat", x = "year", title = "All together")
sm_combo_bar_alt <- specify_combination(combo_type = "small_multiple",
                                         base_charts = "bar_chart_alt",
                                         facet_by = "country")

plot(bar_chart_alt)
```



```
plot(sm_combo_bar_alt)
```



Other chart types cannot be easily subsetted.

For example, it would not be meaningful to make a small multiple of a phylogenetic tree while only showing a subset of the tree. The same is generally true for maps, and networks.

Although there are ways to truly subset them, it's messy and the whole underlying structure matters, so minCombinR will give you the whole network structure.

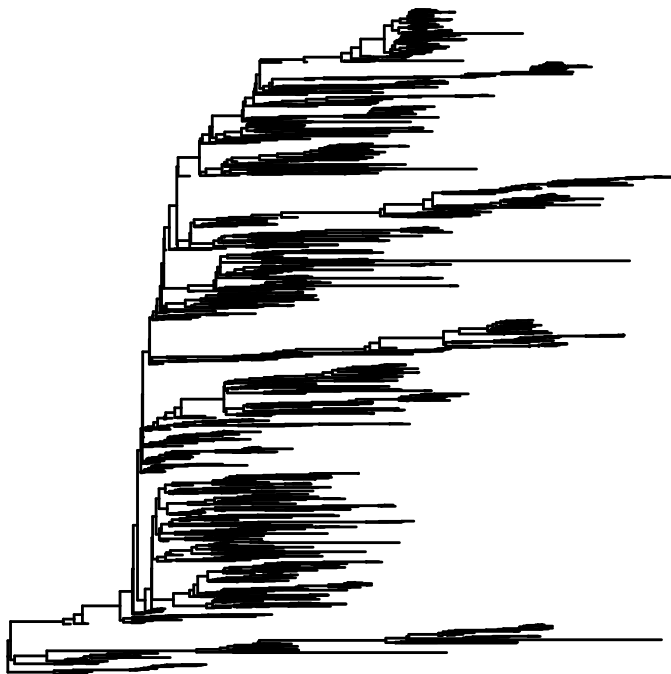
In the current implementation of minCombinR, there needs to be some tabular data associated with non-tabular data in order to understand what should be visualized in the first place.

```
# Tree data
tree_dat <- input_data(file = system.file("extdata", "ebov_tree.nwk", package = "mincombinr"), dataType = "tree")

tree_dat_meta <- input_data(file = system.file("extdata", "ebov_tree.nwk", package = "mincombinr"),
                             dataType = "tree",
                             metadata = system.file("extdata", "ebov_metadata.csv",
                                                     package = "mincombinr"))

# Specify the simple tree
tree_chart <- specify_single(chart_type = "phylogenetic tree",
                              data = "tree_dat_meta")

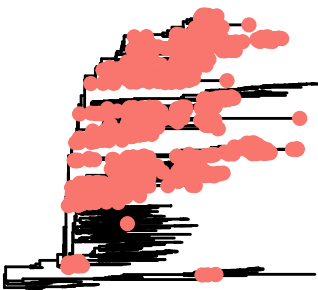
# Let's see what it looks like without facetting
plot(tree_chart)
```



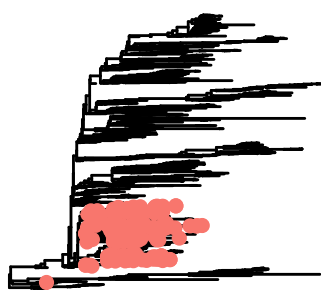
```
# Now let's specify and plot a small multiples combination
sm_combo_tree <- specify_combination(combo_type = "small_multiple",
                                     base_charts = "tree_chart",
                                     facet_by = "country")

plot(sm_combo_tree)
```

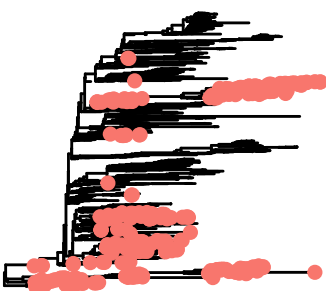
SLE



LBR



GIN



Color Aligned Combinations

Finally, it could be interesting to link several different chart types together by color.

In the above examples, we may want to link the phylogenetic tree with the timeline by their countries.

For the non-tabular data, it's important to have some associated metadata, otherwise, it is not possible to link information. It is up to the user to establish that two variables are actually linkable by the same variable. Some of the code from the spatial aligned combination is borrowed to see if two datasets are even linkable to help with the color linkage.

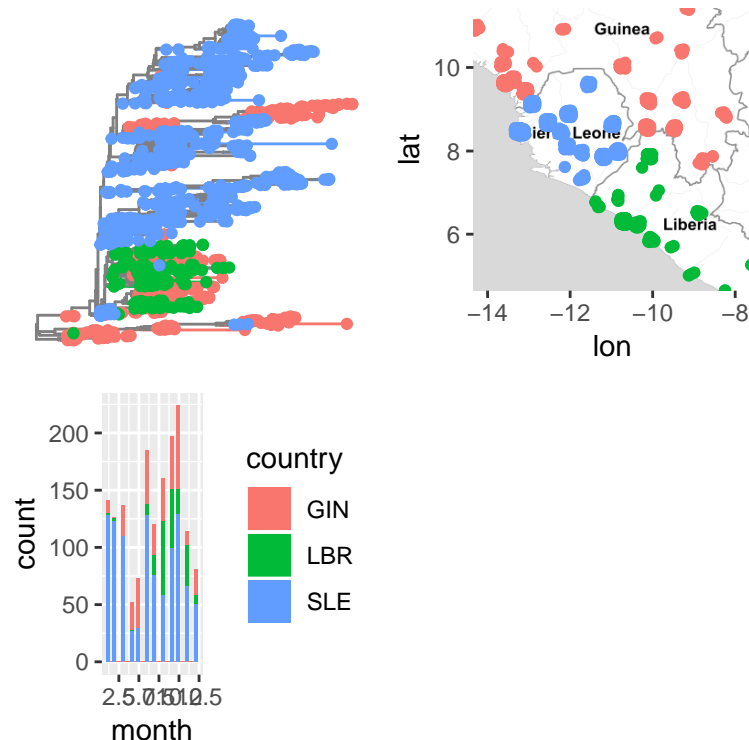
First scenario - no metadata provided for the tree

minCombinR will try to find if there are linkages between the tabular data in one chart and non-tabular data in another chart. If there are EXACT MATCHES, it will link the tabular data to the tree's metadata

```
# Specify the phylogenetic tree and histogram
phyloTree_chart <- specify_single(chart_type = "phylogenetic tree", data = "tree_dat")
epicurve <- specify_single(chart_type = "histogram", data = "tab_dat", x = "month")

# Specify that you want to combine with color
color_combo <- specify_combination(combo_type = "color_aligned",
                                   base_charts = c("phyloTree_chart", "map_chart", "epicurve"),
                                   link_by = "country")

# Now plot!
plot(color_combo)
```



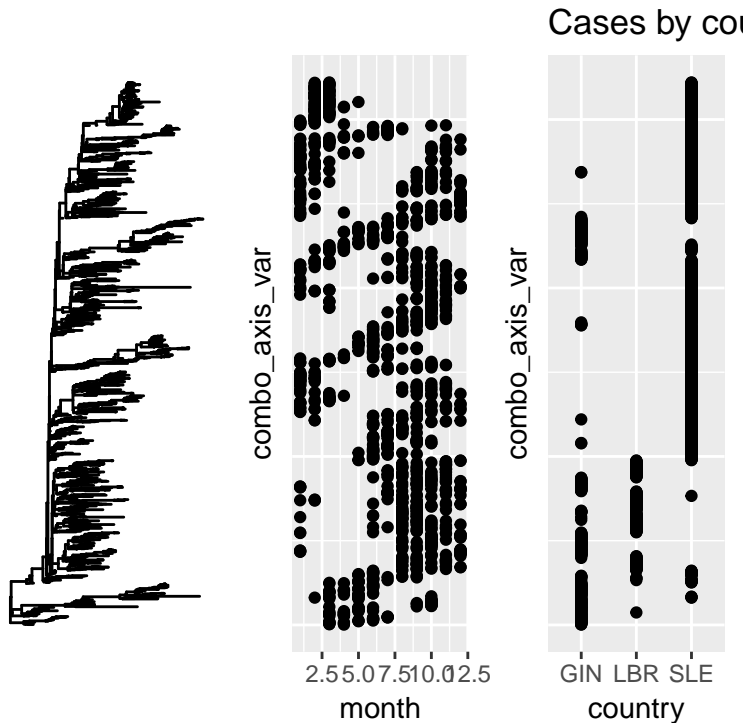
Spatially Aligned Combinations

Spatially aligned combinations line up charts so that they can be read across horizontally and vertically. Critically, this isn't an arbitrary concatenation of charts, but rather, charts are aligned so that the same data are read across.

```
scatter_chart <- specify_single(chart_type = "scatter", data = "tab_dat", x = "month", y = "site_id")
scatter_chart_two <- specify_single(chart_type = "scatter", data = "tab_dat", x = "country",
                                     y = "site_id", title = "Cases by country")

spatial_aligned_combo <- specify_combination(combo_type = "spatial_aligned",
                                             base_charts = c("phyloTree_chart", "scatter_chart", "scatter_chart_two"))

plot(spatial_aligned_combo)
```

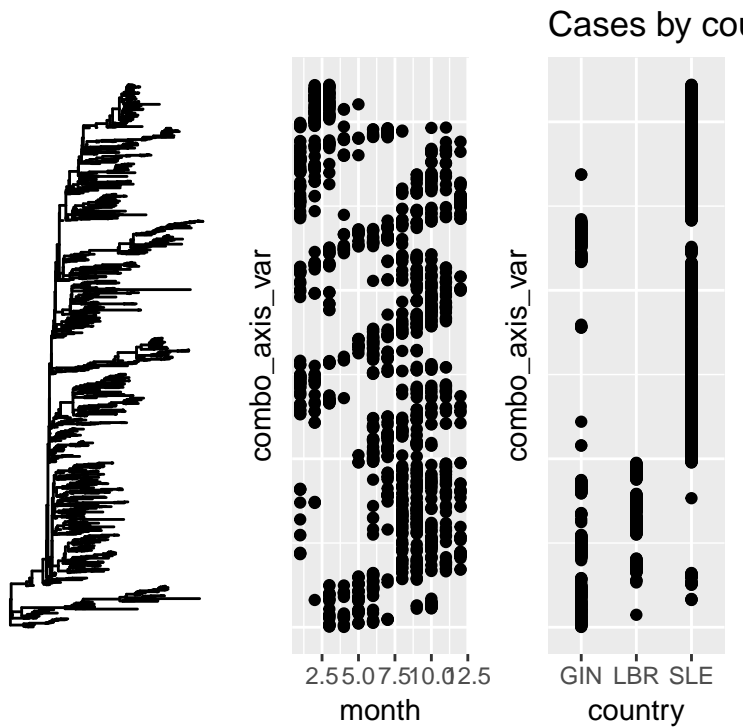


This example produces an error (and this is the right thing for it to do) because pie charts cannot be part of spatial aligned combinations. Pie charts will be automatically removed from the specifications with a warning message to the user.

```
# Specifications
pie_chart <- specify_single(chart_type = "pie", data = "tab_dat", x = "country")
spatial_aligned_combo <- specify_combination(combo_type = "spatial_aligned",
                                             base_charts = c("phyloTree_chart", "scatter_chart", "scatter_chart_two"))

## [1] "The following chart types cannot form a composite combination: pie_chart. Composite combination"

# Plot
plot(spatial_aligned_combo)
```

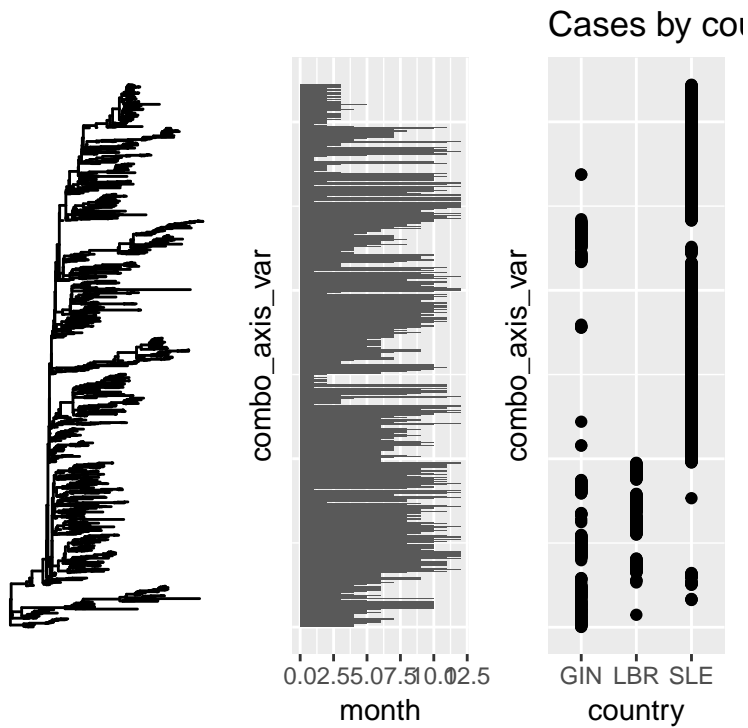



Let's try combining a phylogenetic tree, bar chart, and scatter chart

```
# Specifications
bar_alt <- specify_single(chart_type = "bar", data = "tab_dat", x = "site_id", y = "month", rm_x_label=

spatial_aligned_combo <- specify_combination(combo_type = "spatial_aligned",
                                             base_charts = c("phyloTree_chart", "bar_alt", "scatter_chart_two

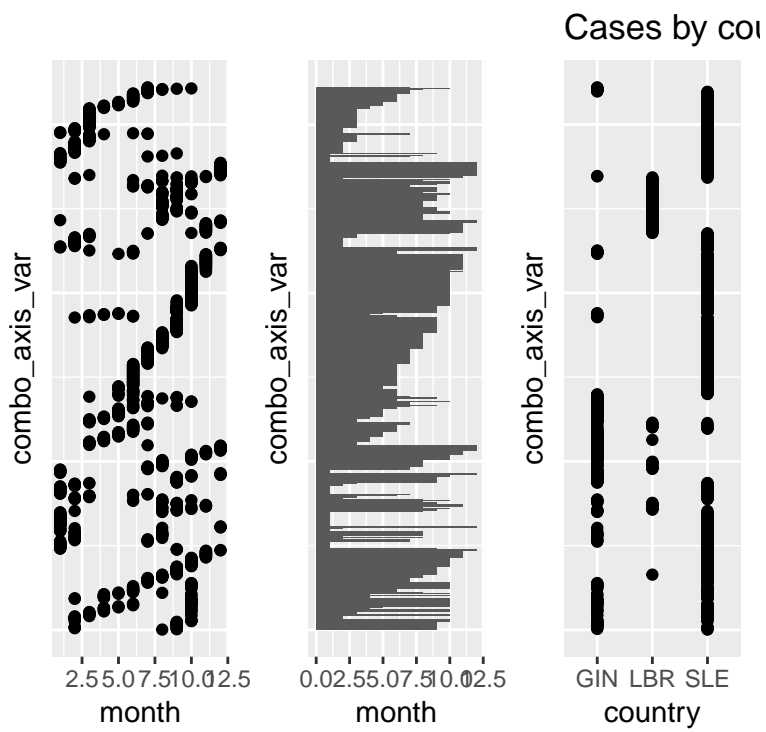
# Plot
plot(spatial_aligned_combo)
```



It also works when there isn't a tree involved

```
spatial_aligned_combo <- specify_combination(combo_type = "spatial_aligned",
                                             base_charts = c("bar_alt", "scatter_chart", "scatter_chart_two"))

plot(spatial_aligned_combo)
```



Here's a combination with a genomic map.

```

# Genomic chart
# For illustrative purposes, show fewer positions
diff_seq <- get_diff_pos(genomic_dat)
genome_chart <- specify_single(data = "genomic_dat",
                               chart_type = "alignment",
                               title="Genome Alignment",
                               show_pos=diff_seq[1:20])

# Timeline, with some fake end_dates and using the existing tabular data
time_dat <- tab_dat@data[[1]]
time_dat$collection_date <- as.Date(time_dat$collection_date)

# Let's add some end dates to keep it interesting
time_dat$collection_date_end <- time_dat$collection_date + sample(10:30,nrow(time_dat), replace = TRUE)

time_dat$collection_date_end <- sapply(as.character(time_dat$collection_date_end),
                                       function(x){
                                         if(runif(1)>0.9)
                                           return(x)
                                         return(NA)
                                       })

time_dat <- dplyr::filter(time_dat,!is.na(collection_date_end))

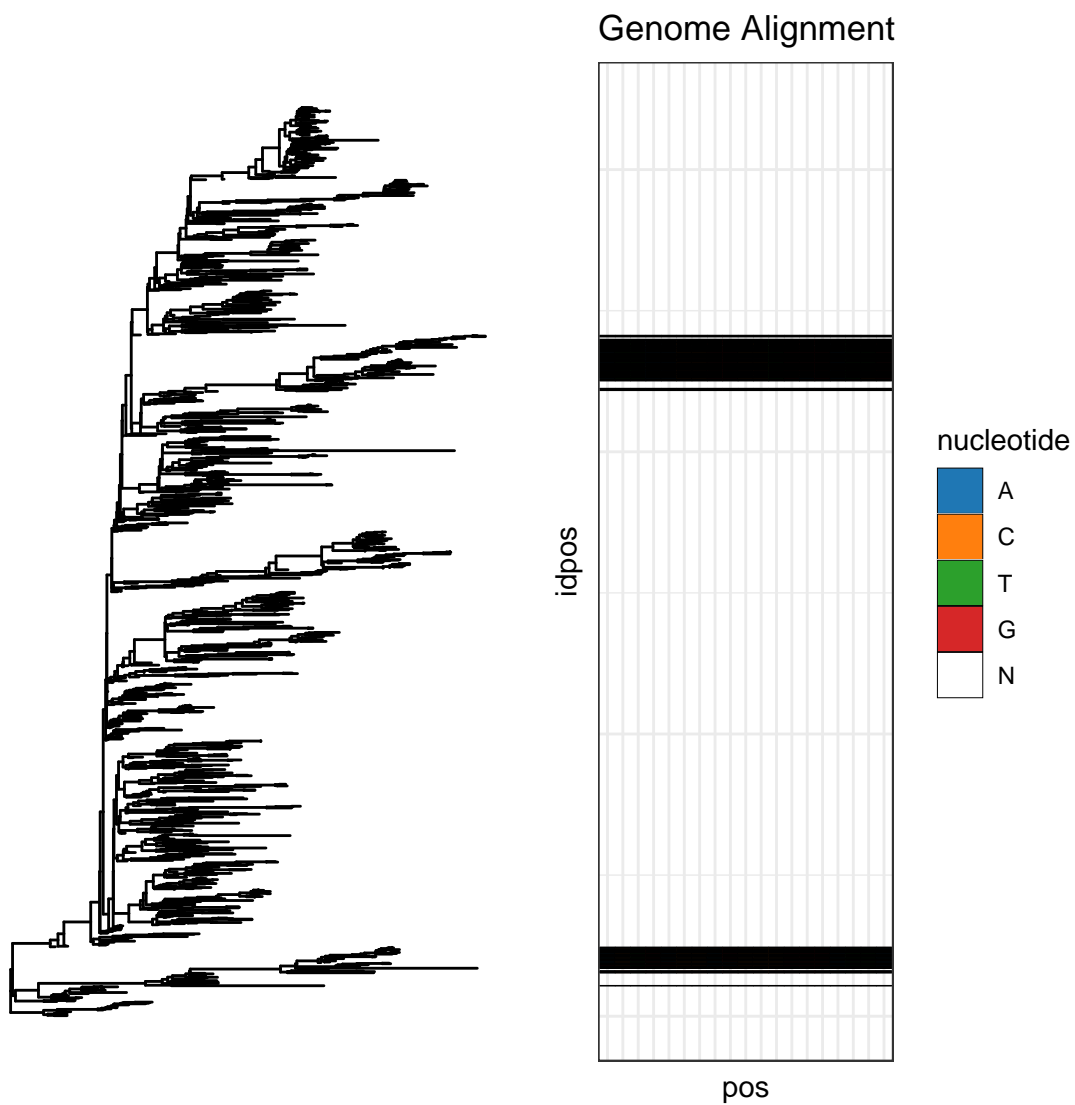
# Specifications
timeline_chart <- specify_single(chart_type = "timeline",
                                  data="time_dat",
                                  start = "collection_date",
                                  end = "collection_date_end",
                                  y = "site_id",
                                  title="Timeline")

spatial_aligned_combo <- specify_combination(combo_type = "spatial_aligned",
                                              base_charts = c("phyloTree_chart",
                                                            "scatter_chart_two",
                                                            "timeline_chart",
                                                            "genome_chart"))

spatial_aligned_combo <- specify_combination(combo_type = "spatial_aligned",
                                              base_charts = c("phyloTree_chart",
                                                            "genome_chart"))

plot(spatial_aligned_combo)

```



Note that in the above case, genomic data is not available for all of the items in the phylogenetic tree. In fact, our source data is for Guinea only. The composite algorithm is able to adjust in instances where one dataset is a perfect subset of the other. It is up to the user to ensure this.