Supplemental File of

ggtreeExtra: A universal package to annotate phylogenetic tree using graph alignment

Shuangbin Xu, Zehan Dai, Pingfan Guo, Xiaocong Fu, Shanshan Liu, Lang Zhou, Wenli Tang, Tingze Feng, Meijun Chen, Li Zhan and Guangchuang Yu*

*correspondence: guangchuangyu@gmail.com, gcyu1@smu.edu.cn

1 The purpose of development and overview

Integrating and visualizing associated data to the phylogenetic tree can help to find biological patterns and generate new hypotheses. The associated data type of phylogenetic tree can be roughly divided into continuous data and categorical data (discrete data). The continuous data sets represent measurements, they can be measured but not be counted, such as the height, weight, abundance of species, gene expression and the number of target genes etc. The categorical data sets represent characteristics, they can not be measured but they can be counted, such as endemic region information of virus, taxonomy information of species, type of target gene and sampling location information etc. Certainly, categorical data can also take on numerical values (for example, 1 for target gene A, 2 for target gene B). The associated data sets are also often multi-dimensional. Several tools have been developed to integrate and display associated data to phylogenetic tree. However, they still have some shortcomings, such as don't support annotation circular layout (tree and graphic alignment), provide few geometric layers, need predefined input etc, which make them not universal. Here, we developed *ggtreeExtra* to annotate multi-dimensional data to the outer of phylogenetic tree (Fig. S1). It can link *ggtree*(Yu et al. 2017) and geometric layers function defined in *ggplot2*(Wickham 2016) or other ggplot2-based package. And it supports not only circular layout, but also other layouts defined in *ggtree*(Yu et al. 2017). The tree can be annotated by geometric function defined in ggtree(Yu et al. 2017) before passing it to ggtreeExtra (Fig. S1). In addition, it was developed based on the grammar of graphics(Wilkinson 2012). So user can easily map the variables (abundance of species, length of genome, sampling location) of associated data to aesthetic attributes (size, color, shape) of outer geometric objects (bar, point, box plot) of phylogenetic tree using *ggtreeExtra*. The details (such as legends and theme) of figure can be adjusted by corresponding *scale* function and theme function defined in ggplot2(Wickham 2016) (Fig. S1). Compared other tools, *ggtreeExtra* supports more layouts for phylogenetic tree annotation (tree and graph alignment), it can integrate more geometric layers and don't need predefined input, which make it more universal (Tab. S1).

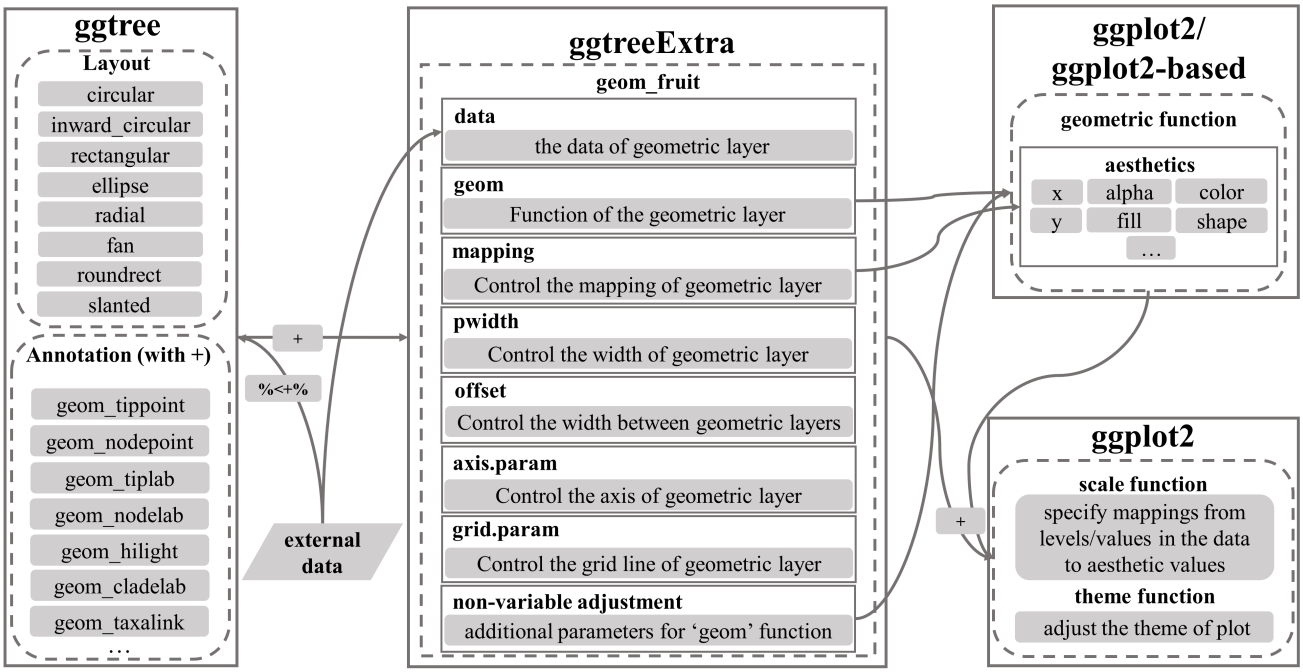

Fig. S1: overview of ggtreeExtra.

Table S1: Comparison list of ggtreeExtra and other tools

Tools	**Platform**	**Supported layouts for tree annotation[1]**	**Annotation layers**	**Supported grammar of graphic**	**Combined freely[2]**	**Methods of figure out**	**Layer operations**
ggtreeExtra	R package	circular, inward circular, rectangular, slanted, ellipse, round rectangular, radial	Heat map, scatter, simple bar, patter bar, stacked or dodged bar, box, pattern box, violin, dot intervals plot, density plot, pie, image plot	Yes	Yes	programming	add, modify, delete
GraPhlAn	Python package	circular	Heat map, scatter, simple bar	No	Yes	command line (configure file)	add
ETE3	Python package	rectangular	Heat map, scatter, simple bar, stacked bar, box, pie, image plot	No	Yes	programming and interaction (mouse click)	add
iTOL	Web tool	circular, rectangular	Heat map, scatter, simple bar, stacked bar, box, pie, image plot	No	Yes	interaction (mouse click configure file)	add
Microreact	Web tool	circular, rectangular	Heat map, scatter	No	Yes	interaction (mouse click and command line configure file)	add
Evolview	Web tool	circular, rectangular, slanted, round rectangular	Heat map, scatter, simple bar, stacked bar, box	No	Yes	interaction (mouse click configure file)	add

[1] tree annotation: tree and geometric alignment;
[2] Combined freely: layers can be combined freely.

2 Geometric layers supported by *geom_fruit* of *ggtreeExtra*

ggtreeExtra is designed to link ggtree(Yu et al. 2017) and some **geom** functions defined in *ggplot2*(Wickham 2016) and other *ggplot2* extension packages (Fig. S1). Here is the list of the geometric layer functions which work seamlessly with *geom_fruit* of *ggtreeExtra* (Tab. S2). Each geometric layer function has own unique geometric attributes (Tab. S2). User can choose appropriate geometric layer functions according the type of associated data. And the variables of associated data can be mapped to the attributes of corresponding geometric layer function. For example, when user want to view the distribution and uncertainty (continuous data) of associated data in different groups (such as the gene expression or species abundance in different samples), Box, violin or dot interval plot etc can be used to display them (Fig. S2). For the simple numeric data (length of genome, abundance of species), they can be visualized with simple (pattern) bar plot or grouped (pattern) bar plot (pattern bar plot is efficient to the visualization that don't want to use color) (Fig. S3). Certainly, the aesthetics parameters of **geom** not only can be mapped by variables, but also can be used to adjust the attributes of geometric layers directly (such as *pattern_fill* in Fig. S3). Since *ggtreeExtra* can work seamlessly with the *geom* functions, it can integrate more geometric layers compare other tools (Tab. S1). Until now, *ggtreeExtra* can integrate heat map, scatter plot, simple (grouped) (pattern) bar plot, simple (grouped) (pattern) box plot, violin, dot intervals plot, density plot, image plot, pie (Tab. S1). Even though other tools might provide the same graphic layers, *ggtreeExtra* is more powerful and flexible since it can inherit the features of the geometric functions, for example, *iTOL*, *Evolview* and *ggtreeExtra* all support displaying image plot on the phylogenetic tree, but *ggtreeExtra* is more flexible, it can integrate *ggplot2* object (Fig. S4.A), and the attributes of subplot integrated can also be mapped to variable of associated data (Fig.S4.B). As the *ggplot2*(Wickham 2016) community keeps expanding and more *geom* functions will be implemented in either *ggplot2*(Wickham 2016) or other extensions, *geom_fruit* will gain more power to present data in future.

Table S2: List of geometric layers supported by 'geom_fruit()'

Package	Geom layer	Visual characteristic	Description
ggdist	geom_dots	alpha, color, fill, size, shape	creates dotplots that automatically determines a bin width that ensures the plot fits within the available space
	geom_dotsinterval	alpha, color, fill, size, shape	creates dots, intervals, and quantile dotplots
	geom_pointinterval	alpha, color, fill, size, shape	creates point and multiple uncertainty interval
	geom_slab	alpha, color, fill	creats slab geom
	geom_slabinterval	alpha, color, fill	creates slab, point and interval meta-geom
ggimage	geom_image	alpha, color, size	visualizes image files
	geom_phylopic	alpha, color, size	queries image files from phylopic database and visualizes them
ggpattern	geom_bar_pattern	pattern_alpha, pattern_color, pattern_fill pattern_angle, alpha, color, fill	draws bar charts with support for pattern fills
	geom_boxplot_pattern	pattern_alpha, pattern_color, pattern_fill pattern_angle, alpha, color, fill	draws box and whiskers plot with support for pattern fills
	geom_col_pattern	pattern_alpha, pattern_color, pattern_fill pattern_angle, alpha, color, fill	draws bar charts using 'stat_identity()' with support for pattern fills
	geom_tile_pattern	pattern_alpha, pattern_color, pattern_fill pattern_angle, alpha, color, fill	draws rectangle by using the center of the tile and its size with support for pattern fills
ggplot2	geom_bar	alpha, color, fill	draws bar charts
	geom_boxplot	alpha, color, fill	draws box and whiskers plot
	geom_col	alpha, color, fill	draws bar charts using 'stat_identity()'
	geom_label	alpha, color, fill, size	draws a rectangle behind the text
	geom_point	alpha, color, fill, shape, size	creats scatterplots
	geom_raster	alpha, fill	a high performance special case for all the tiles are the same size
	geom_text	color, size	adds text to the plot
	geom_tile	alpha, color, fill	draws rectangle by using the center of the tile and its size
ggpmisc	geom_plot	vp.width,vp.height	ggplot objects as insets to the base ggplot, using syntax similar to that of 'geom_label'
	geom_table	size	adds a textual table directly to the ggplot, using syntax similar to that of 'geom_label'
ggrepel	geom_text_repel	color, size	adds text to the plot. The text labels repel away from each other and away from the data points
	geom_label_repel	alpha, color, fill, size	draws a rectangle underneath the text. The text labels repel away from each other and away from the data ponts
ggridges	geom_density_ridges	alpha,fill	arranges multiple density plots in a staggered fashion
	geom_density_ridges2	alpha,fill	arranges multiple density plots in a staggered fashion
	geom_ridgeline	alpha,color,fill	plots the sum of the 'y' and 'height' aesthetics versus 'x', filling the area between 'y' and 'y + height' with a color
	geom_ridgeline_gradient	color,fill	works just like 'geom_ridgeline' except that the 'fill' aesthetic can vary along the x axis
ggstance	geom_barh	alpha, color, fill	horizontal version of 'geom_bar()'
	geom_boxploth	alpha, color, fill	horizontal version of 'geom_boxplot()'
	geom_colh	alpha, color, fill	horizontal version of 'geom_col()'
ggstar	geom_star	alpha, color, fill, size, starshape	creates scatterplots
ggsymbol	geom_symbol	alpha, color, fill, size, symbolshape	creates scatterplots
scatterpie	geom_scatterpie	alpha, color, fill	creates scatter pie plot

```
library(ggtree)
library(ggplot2)
library(ggtreeExtra)
library(patchwork)
library(ggridges)
library(phyloseq)

set.seed(1024)
```

```r
data("GlobalPatterns")
GP <- GlobalPatterns
GP <- prune_taxa(taxa_sums(GP) > 1000, GP)
sample_data(GP)$human <- get_variable(GP, "SampleType") %in%
  c("Feces", "Skin")

mergedGP <- merge_samples(GP, "SampleType")
mergedGP <- rarefy_even_depth(mergedGP, rngseed=1024)
mergedGP <- tax_glom(mergedGP,"Order")

melt_simple <- psmelt(mergedGP) %>%
  dplyr::filter(Abundance < 120) %>%
  dplyr::select(OTU, val=Abundance)

p <- ggtree(mergedGP, size = 0.3) +
     geom_tippoint(aes(color = Phylum),
                   show.legend = FALSE,
                   size=0.6)
p1 <- p +
      geom_fruit(
          data = melt_simple,
          geom = geom_density_ridges,
          mapping = aes(y = OTU, x = val, fill = Phylum),
          offset = 0.12,
          pwidth = 0.4,
          lwd = .05,
          axis.params = list(
                           axis = "x",
                           text.size  = 1.2,
                           hjust = 0,
                           vjust = 1,
                           text.angle = -45
                       ),
          grid.params = list(),
          show.legend = FALSE
      ) + ggtitle("A")

p2 <- p +
      geom_fruit(
          data = melt_simple,
          geom = geom_boxplot,
          mapping = aes(y = OTU, x = val, fill = Phylum),
          offset = 0.12,
          pwidth = 0.4,
          size = 0.1,
          outlier.size = 0.4,
          outlier.stroke = 0.06,
          outlier.shape = 21,
          axis.params = list(
                           axis = "x",
                           text.size = 1.2,
                           hjust = 0,
                           vjust = 1,
                           text.angle = -45
                       ),
          grid.params = list(),
          show.legend = FALSE
      ) + ggtitle("B")

p3 <- p1 + layout_circular() + ggtitle("C")
```

```
p4 <- p2 + layout_circular() + ggtitle("D")
(p1 + p2)/(p3 + p4)
```

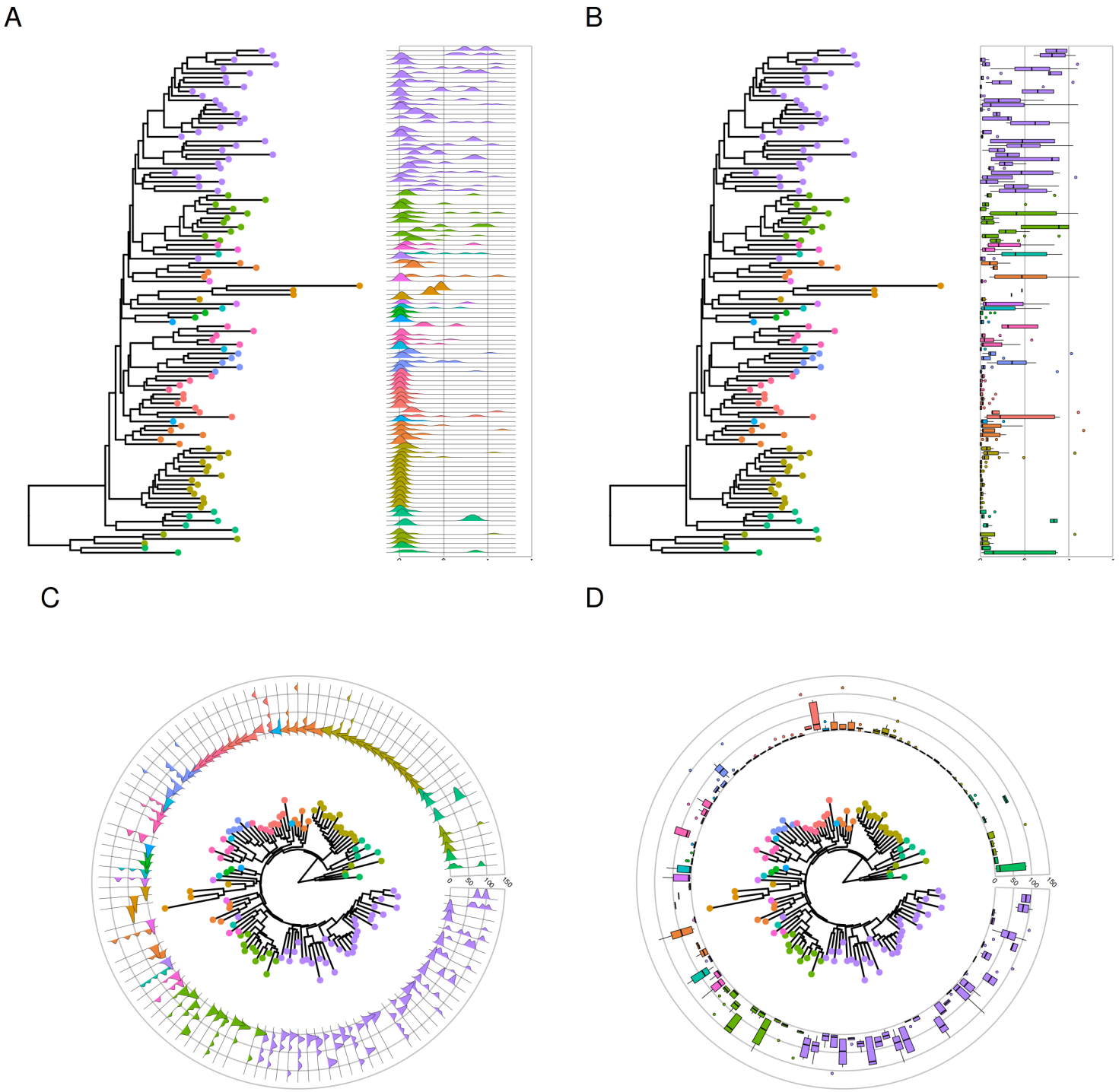

Fig. S2: This example shows *ggtreeExtra* can work with *geom_density_ridges* of *ggridges*(Wilke 2020) and *geom_boxplot* of *ggplot2*(Wickham 2016). This dataset has also been visualized by *facet_plot*, but it can not work with circular layout tree(Yu et al. 2018). The associated data (melt_simple) was imported with *data* of *geom_fruit*, it has a column (*OTU*) contained tip labels. Then the tip labels column was assigned to *y*, the *val* is abundance of species (continuous data), which was mapped to the *x*, and *OTU* is the phylum information of species (categorical data), which was mapped to the *fill* (color).

```
library(ggtree)
library(ggtreeExtra)
library(ggpattern)
library(ggplot2)
library(patchwork)
```

```r
set.seed(1024)
tr <- rtree(20)

dat <- tibble::tribble(
            ~value, ~group,
            abs(rnorm(5, 10, sd = 3)), "A",
            abs(rnorm(3, 12, sd = 2)), "B",
            abs(rnorm(5, 9, sd = 3)),  "C",
            abs(rnorm(7, 6, sd = 2)),  "D"
       ) %>% tidyr::unnest(value)

dat$id <- tr$tip.label

dt <- tibble::tribble(
            ~value, ~class,
            abs(rnorm(40, 10, sd = 3)), "A",
            abs(rnorm(24, 12, sd = 2)), "B",
            abs(rnorm(40, 9, sd = 3)), "C",
            abs(rnorm(56, 6, sd = 2)), "D"
      ) %>% tidyr::unnest(value)

dt$id <- c(rep(tr$tip.label[1:5], 8), rep(tr$tip.label[6:8], 8),
            rep(tr$tip.label[9:13],8), rep(tr$tip.label[14:20], 8)
          )

p1 <- ggtree(tr, size=0.2, branch.length="none")
p2 <- ggtree(tr, size=0.2, layout="slanted", branch.length="none")
p3 <- ggtree(tr, size=0.2, layout="fan", open.angle=180, branch.length="none")
p4 <- ggtree(tr, size=0.2, layout="fan", open.angle=180, branch.length="none")

p1 <- p1 +
      geom_fruit(
          data=dat,
          geom=geom_bar_pattern,
          mapping=aes(y=id, x=value, pattern=group, pattern_angle=group),
          width=0.6, stat="identity",
          pwidth = 0.6, pattern_spacing =0.01,
          pattern_size = 0.1, pattern_density = 0.4,
          fill = "grey", pattern_fill="grey35",
          position=position_identityx(),
          axis.params=list(axis="x", text.size=1.5, text.angle=-45, hjust=0)
      ) + theme(legend.key.size = unit(0.3, 'cm'))

p2 <- p2 +
      geom_fruit(
          data=dat,
          geom=geom_bar_pattern,
          mapping=aes(y=id, x=value, pattern=group, pattern_fill=group),
          width=0.6, stat="identity",
          pwidth = 0.6, pattern_spacing =0.01,
          pattern_size = 0.1, pattern_density = 0.4,
          fill = "grey",
          position=position_identityx(),
          axis.params=list(axis="x",text.size=1.5, text.angle=-45, hjust=0)
      ) + theme(legend.key.size = unit(0.3, 'cm'))

p3 <- p3 +
      geom_fruit(
          data=dt,
```

```
          geom=geom_boxplot_pattern,
          mapping=aes(y=id, x=value, pattern=class, pattern_angle = class),
          size=0.1, outlier.shape=NA,
          pwidth=0.5, pattern_size = 0.1,
          pattern_density = 0.4, pattern_spacing =0.01,
          fill = "grey", pattern_fill="grey35",
          position=position_dodgex(),
          grid.params=list(),
          axis.params=list(axis="x", text.size=1.5, text.angle=-45, hjust=0)
      ) +
      theme(legend.key.size = unit(0.35, 'cm'))

p4 <- p4 +
      geom_fruit(
          data=dt,
          geom=geom_boxplot_pattern,
          mapping=aes(y = id, x = value, pattern = class, pattern_fill = class),
          size = 0.1, outlier.shape = NA,
          pwidth = 0.5, pattern_size = 0.1,
          pattern_density = 0.4, pattern_spacing = 0.01,
          fill = "grey",
          position = position_dodgex(),
          grid.params = list(),
          axis.params = list(axis="x", text.size=1.5, text.angle=-45, hjust=0)
      ) +
      theme(legend.key.size = unit(0.35, 'cm'))

(p1 + p2)/(p3 + p4)
```

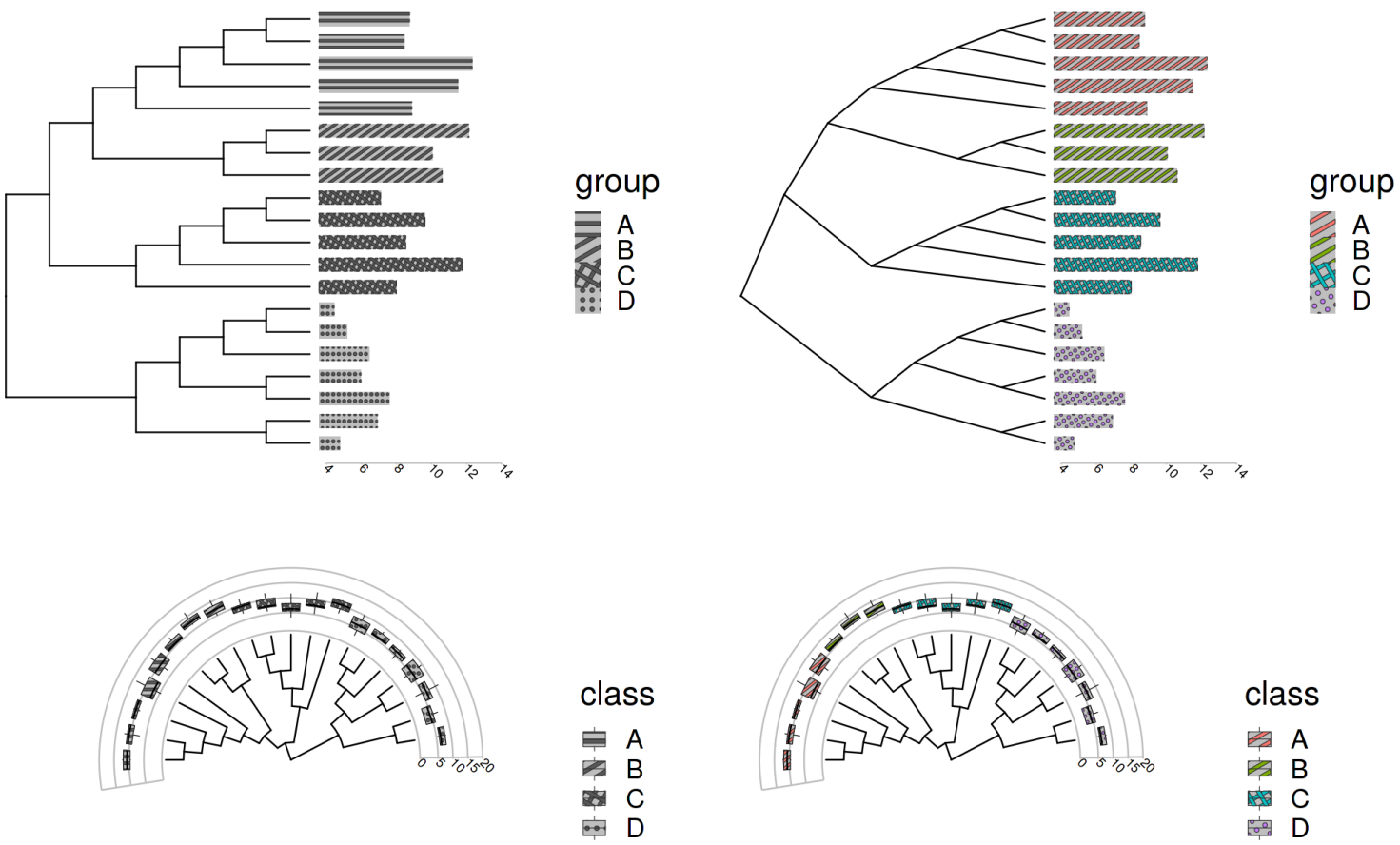

Fig. S3: This example shows *ggtreeExtra* can work with *geom_bar_pattern* and *geom_boxplot_pattern* of *ggpattern*(FC 2020). This example also shows *ggtreeExtra* can work with multiple layouts of tree.

```r
library(ggtreeExtra)
library(ggtree)
library(ggpmisc)
library(ggplot2)
library(ggimage)
library(patchwork)

set.seed(1024)
tr <- rtree(10)

dat1 <- data.frame(value=c(abs(rnorm(10, 3)),abs(rnorm(10, 5))), group=c(rep("A",10),rep("B",10)))
dat2 <- data.frame(value=c(abs(rnorm(15, 6)), abs(rnorm(15, 3))), group=c(rep("A",15),rep("B",15)))

subp1 <- ggplot(dat1) +
         geom_boxplot(aes(x=group, y=value, fill=group),
                      size=0.1, outlier.size = 0.1, show.legend=F) +
         theme_bw() +
         theme(legend.key.size = unit(1, 'mm'),
               plot.background = element_rect(fill=NA, color=NA),
               panel.border = element_rect(size=0.1),
               legend.title = element_text(size=3.6),
               legend.text = element_text(size=3.2),
               axis.ticks = element_line(size=0.1),
               axis.text = element_text(size=2.8),
               axis.title = element_text(size=3))

subp2 <- ggplot(dat2) +
         geom_boxplot(aes(x=group, y=value, fill=group),
         size=0.1, outlier.size = 0.1, show.legend=F) +
         theme_bw() +
         theme(legend.key.size = unit(1, 'mm'),
               plot.background = element_rect(fill=NA, color=NA),
               panel.border = element_rect(size=0.1),
               legend.title = element_text(size=3.6),
               legend.text = element_text(size=3.2),
               axis.ticks = element_line(size=0.1),
               axis.text = element_text(size=2.8),
               axis.title = element_text(size=3))

dt <- tibble::tibble(id=c("t1", "t2"), plot=list(subp1, subp2))

p1 <- ggtree(tr,
             layout="fan",
             open.angle=180
             ) +
      geom_tiplab(align=T, size=1.6)

p2 <- p1 +
      geom_fruit(
                 data=dt,
                 geom=geom_plot,
                 mapping=aes(
                             y=id,
                             label=plot
                            ),
                 offset=0.4,
                 position=position_identityx(),
                 vjust=0.6,
                 hjust=0.6,
                 vp.width=0.22,
```

```
                  vp.height=0.22
      ) +
      ggtitle("A")

newick <- "((Pongo_abelii,(Gorilla_gorilla_gorilla,(Pan_paniscus,Pan_troglodytes)
            Pan,Homo_sapiens)Homininae)Hominidae,Nomascus_leucogenys)Hominoidea;"

tree <- read.tree(text=newick)

p3 <- ggtree(tree,
             layout="fan",
             open.angle=180) +
      geom_tiplab(aes(label=label), offset = .2, size=1.6)

d <- ggimage::phylopic_uid(tree$tip.label)
d$body_mass <- c(52, 114, 47, 45, 58, 6)
d$ids <- tree$tip.label

p4 <- p3 +
      geom_fruit(
          data = d,
          geom = geom_phylopic,
          mapping = aes(y=ids, image=uid, colour=body_mass),
          position = position_identityx(),
          offset = 2,
          size = 0.04
      ) +
      ggtitle("B") +
      xlim(NA, 12) +
      scale_color_viridis_c(guide=guide_colorbar(barheight=2, barwidth=0.4)) +
      theme(legend.text=element_text(size=6),
            legend.title=element_text(size=7))

p2 + p4
```

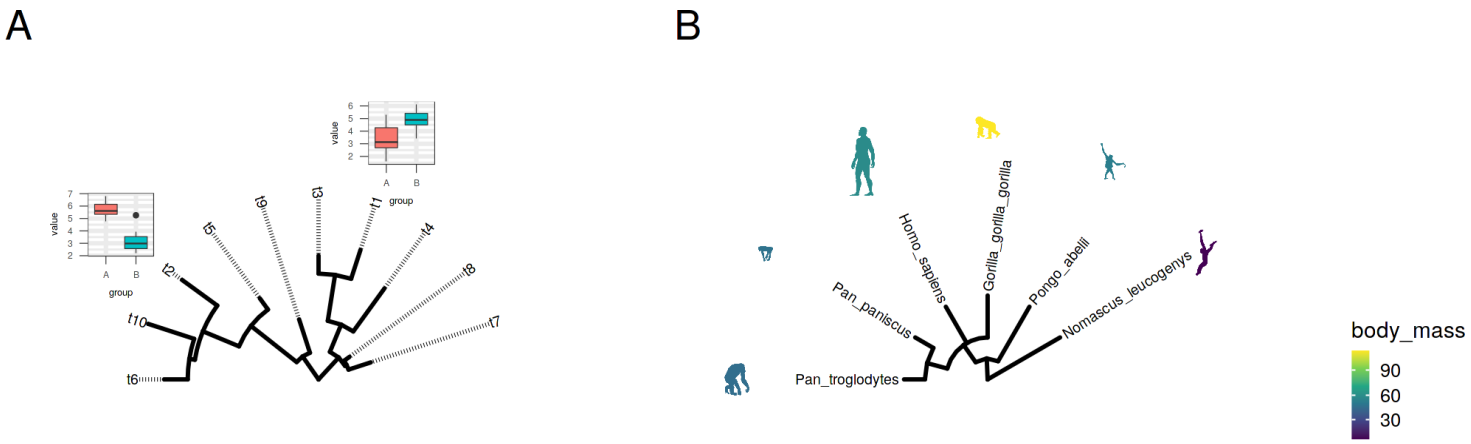

Fig. S4: This example show *ggtreeExtra* can integrate subplot to phylogenetic tree, and the subplot can be *ggplot* object (Fig.S4.A), and the attributes of subplot can be mapped the variables of associated data when silhouette image file was used(Fig.S4.B).

3 Examples of mapping and visualizing associated data on circular layout tree

ggtreeExtra can integrate many geometric layers by linking *geom* function defined in *ggplot2*(Wickham 2016) or ggplot2-extension packages (Tab. S2 and S1). This is the basis for *ggtreeExtra* to be used for the annotation of phylogenetic tree. We presented two examples (Fig. S2 and S3) to show how to use *ggtreeExtra* to map simple associated data to phylogenetic tree by linking

geom funtions. For the multiple associated data sets, circular layout is an efficient way to visualize multi-dimensional data sets, since it can reduce space and make the graph more compact. *ggtreeExtra* support annotation of multiple layouts of tree (Fig. S2 and S3 and Tab. S1), including circular layout. Furthermore, the associated data can be imported with the *data* parameter of *geom_fruit* (Fig. S2 and S3), it can also be integrated to tree data (ggtree graphic object) with %<+% of *ggtree*(Yu et al. 2018), before passing it to *geom_fruit* (Fig. S1), or the tree data from output of the *treeio*, which can parse different file formats as well as the outputs of commonly used software(Wang et al. 2020), can also be visualized by *ggtreeExtra*. Therefore, *ggtreeExtra* supports evolutionary statistics inferred by commonly used software, such as *BEAST*(Drummond and Rambaut 2007), *RAxML*(Stamatakis 2014), *HyPhy*(Pond, Frost, and Muse 2005), *PAML*(Yang 2007), *ASTRAL*(Mirarab and Warnow 2015), *pplacer*(Matsen, Kodner, and Armbrust 2010), *RevBayes*(Höhna et al. 2016), PHYLODOG(Boussau et al. 2013) and EPA(Berger, Krompass, and Stamatakis 2011) to be visualized by *ggtreeExtra*. Then the variables (abundance of species, length of genome, sampling location) of associated data can be mapped to the attributes of outer geometric objects (bar, point, boxplot) of circular phylogenetic tree (Fig. S1 and S2), since ggtreeExtra was developed based on grammar of graphic(Wilkinson 2012). Here, we present several examples to elucidate how to map and display the associated data on the outer rings of circular phylogenetic trees using *ggtreeExtra*. More examples can be found on the *chapter10* of online book[1].

3.1 Data supported by *ggtreeExtra*

ggtreeExtra is designed to work with *ggtree*(Yu et al. 2017) and *treeio*(Wang et al. 2020), it supports visualizing the tree data parsed by *treeio*(Wang et al. 2020), any data frame that contains a column of tip labels can also be imported with *data* parameter of *geom_fruit* (Fig. S2). Or the data frame that contains a column of tip labels can also be integrated into *ggtree* graphic object using *%<+%* operator(Yu et al. 2018), before passing it to *geom_fruit*. So *ggtreeExtra* can be used to annotate evolutionary statistics inferred by commonly used software, and It is possible to display the evolutionary statistics with phylogeny-associated data (such as traits, metadata) using *ggtreeExtra*. These features are not available in other tools. We reproduce Figure 3.3 of (Smith and Wrighton 2019) to show how to integrate the associated data to tree data using *%<+%* and visualize it. And the source data is available at this repository[2]. The associated data sets contain the information of Ecosystem type, sequencing type and sample treatment method (all categorical data). The first column of external data is tip labels (the element of the column must be unique). The tree was built using *RAxML*(Stamatakis 2014), it is parsed to tree data using *read.raxml* of *treeio*(Wang et al. 2020), contained bootstrap information. The tree data was visualized with *ggtree*(Yu et al. 2017). Then the associated data sets was imported *ggtree* graphic object using %<+%, *geom_fruit* can extract the tree data integrated automatically, and the attributes of data can be mapped using related attributes of geometry layers. The *y* of *aes* can be ignored. Here we used heat map to display the associated data sets (categorical data). The external ring heat maps represent the different types in corresponding categories (the type of Ecosystem was mapped to the color of innermost ring heat map, the type of sequencing was mapped to the color of middle ring heat map, the type of sample treatment was mapped to the color of outermost ring heat map) (Fig. S5). In addition, compared with other tools, no restriction of data types should be visualized in *geom_fruit* of *ggtreeExtra*. It depends on the data types of various geometric functions (Tab. S2). For example, The multiple density plots is not supported by other tools, but *ggtreeExtra* can support it by linking *geom_density_ridges* of *ggridges*(Wilke 2020) (the data types of *geom_density_ridges* should be provided) (Fig. S2.A and S2.C). The *ggplot* graphic object (one type of image plot) is also not supported by other tools, but *ggtreeExtra* can support it by linking *geom_plot* of *ggpmisc*(Aphalo 2020), only we provides the data types of *geom_plot* (Fig. S4.A). As the *geom* of *ggplot2* (Wickham 2016) or other extensions will be updated and developed, so *ggtreeExtra* will be also more to present datasets in the future.

```
library(ggtreeExtra)
library(ggtree)
library(treeio)
library(ggplot2)
library(ggnewscale)
tree <- read.raxml("../data/Methanotroph/Methanotroph_rpS3_Modified_Alignment_RAxML")
# Root the tree to the archaea sequences
tree@phylo <- root(tree@phylo, node=1402, edgelabel=TRUE)
df <- read.csv("../data/Methanotroph/metadata.csv")
# reset the levels of columns to reproduce the order of original figure.
df$Specific.Ecosystem <- factor(df$Specific.Ecosystem,
                                levels=c("Agriculture", "Alkaline/Hypersaline",
                                         "Contaminated/Wastewater", "Endosymbiont",
                                         "Freshwater", "Forest", "Geothermal",
                                         "Marine", "Natural Seep", "Peat",
                                         "Permafrost", "Wetland", "Unknown"))
df$MetaType <- factor(df$MetaType,
```

[1] http://yulab-smu.top/treedata-book
[2] https://github.com/TheWrightonLab/Methanotroph_rpS3Analyses_SmithWrighton2018

```
                    levels=c("Metatranscriptome", "Metagenome",
                             "Single-amplified genome", "Fosmid", NA))
df$Treatment <- factor(df$Treatment, levels=c("Native", "Enrichment","Isolate", "Unknown"))

p <- ggtree(tree, layout="fan", open.angle=30)
print(as.treedata(p))
```

```
## 'treedata' S4 object'.
##
## ...@ phylo:
## Phylogenetic tree with 727 tips and 726 internal nodes.
##
## Tip labels:
##   3300019787==Ga0182031_12339262, 3300014838==Ga0182030_1009273110, gb_PLVF01000413_pos14750To15429=Methyloc
## Node labels:
##   Root, , , , , , ...
##
## Rooted; includes branch lengths.
##
## with the following features available:
##  'bootstrap'.
```

```
p <- rotate_tree(p, 90)
p1 <- p +
      geom_treescale(x=0.2, y=727*6/11, width=1, offset=20) +
      geom_point2(aes(subset = !isTip,
                      fill = cut(bootstrap, c(0, 70, 90, 100), right = F)),
                  shape=21, size=1.2, stroke=0.3) +
      scale_fill_manual(values = c("black", "grey", "white"),
                        name = "Bootstrap (BP)",
                        breaks = c('[90,100)', '[70,90)', '[0,70)'),
                        labels = expression(BP>=90, 70<=BP * '<90', BP < 70),
                        guide=guide_legend(keywidth=0.5, keyheight=0.6,
                                           override.aes=list(size=2.5, stroke=0.3),
                                           order=1)
                        )
# we can use %<+% to integrate the external datasets to tree structure.
# and the y can not be specified.
p2 <- p1 %<+% df +
      new_scale_fill() +
      geom_fruit(
          geom=geom_tile,
          mapping=aes(fill=Specific.Ecosystem),
          offset=0.13,
          width=0.35,
          axis.params=list(axis="x", text="Ecosystem", text.angle=0,
                           hjust=0, text.size=3, family="Times", fontface="bold")
      ) +
      scale_fill_manual(
          values=c("green3","turquoise", "maroon", "orchid",
           "deepskyblue", "forestgreen","salmon", "cadetblue3",
           "slategray4",  "yellowgreen", "gray90", "chocolate2",
           "yellow"),
          guide=guide_legend(title="Ecosystem", keywidth=0.5, keyheight=0.5, order=4),
          na.translate=FALSE
      )

p3 <- p2 +
      new_scale_fill() +
      geom_fruit(
          geom=geom_tile,
```

```
          mapping=aes(fill=MetaType),
          offset=0.13,
          width=0.35,
          axis.params=list(axis="x", text="Sequencing Type", text.angle=0,
                           hjust=0, text.size=3, family="Times", fontface="bold")
      ) +
      scale_fill_manual(
          values=c("red", "black", "dodgerblue", "gray50"),
          guide=guide_legend(title="Sequencing Type", keywidth=0.5, keyheight=0.5, order=3),
          na.translate=FALSE
      )

p4 <- p3 +
      new_scale_fill() +
      geom_fruit(
          geom=geom_tile,
          mapping=aes(fill=Treatment),
          offset=0.13,
          width=0.35,
          axis.params=list(axis="x", text="Sample Treatment", text.angle=0, hjust=0,
                           text.size=3, family="Times", fontface="bold")
      ) +
      scale_fill_manual(
          values=c("red","gray50", "black", "yellow"),
          guide=guide_legend(title="Sample Treatment", keywidth=0.5, keyheight=0.5, order=2),
          na.translate=FALSE
      ) +
      theme(
            legend.background=element_rect(fill=NA), # the background of legend.
            legend.title=element_text(size=9, family="Times", face="bold"),
            legend.text=element_text(size=7, family="Times"), # the text size of legend.
            legend.spacing.y = unit(0.02, "cm"),
            legend.margin=margin(0.1, 0.9, 0.1,-0.9, unit="cm"), # t, r, b, l, cm
            legend.box.margin=margin(0.1, 0.9, 0.1, -0.9, unit="cm"),
            plot.margin = unit(c(-1.2, -1.2, -1.2, 0.1),"cm")
      )

# optional
cladeda <- data.frame(nodeid = c(793, 791, 1384, 1394, 1440, 1405),
                      label = c("Gammaproteobacteria", "Alphaproteobacteria",
                              "Ca.Methylomirabilis", "Methylacidiphilae",
                              "ANME-1", "ANME-2"),
                      horizontal = c(FALSE, FALSE, TRUE, TRUE, TRUE, TRUE),
                      hjust = c(0.5, 0.5, 0, 0, 0, 0))

p5 <- p4 +
      geom_cladelab(
          data = cladeda,
          mapping = aes(node=nodeid, label=label, horizontal=horizontal, hjust=hjust),
          angle = "auto",
          offset = 1.4,
          align = T,
          fontsize = 2,
          barsize = 1,
          family = "Times",
          fontface="bold",
          offset.text = 0.1
      )
p5
```

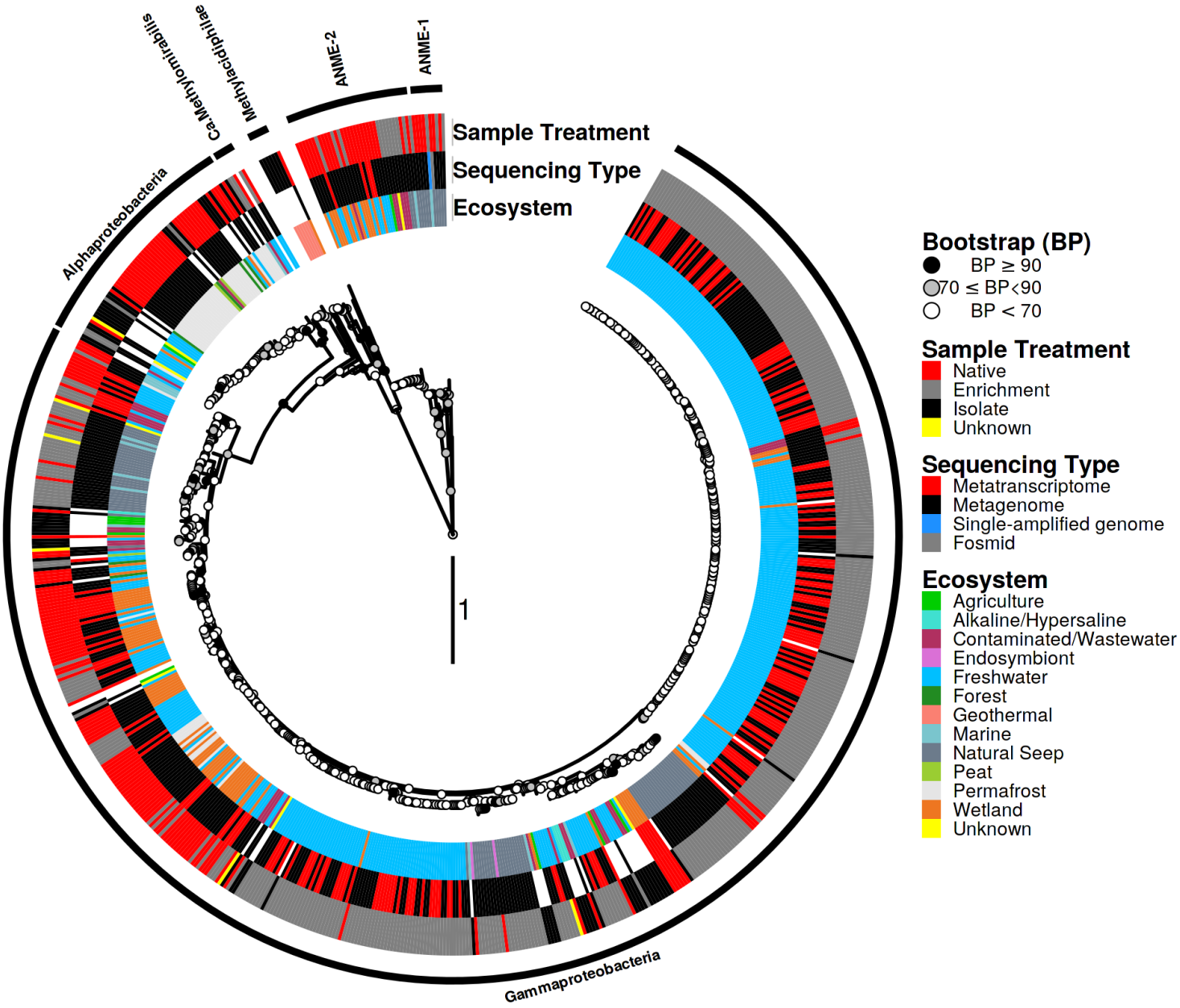

Fig. S5: Phylogeny of methanotroph ribosomal protein S39 (rpS3) genes from Figure 3.3 (Smith and Wrighton 2019). The external ring of circular tree is built with the associated data sets integrated to tree data using %<+%.

3.2 Displaying multiple associated data to circular phylogenetic tree using grammar of graphic

ggtreeExtra is developed based on grammar of graphic(Wilkinson 2012). The variables of associated data sets can be mapped to the attributes of geometric layers, the graphic can be displayed with provided aesthetic mapping and non-variable setting. And users can specify their own set of mappings from levels or values in the data to aesthetic values using the corresponding *scale* function defined in *ggplot2* or *ggplot2-based* packages (Fig. S1). They does not need providing configure file which mixes associated data and the profiler of graphic (it is needed by many other tools (Tab. S1)), so no restriction of data types should be used in *geom_fruit* of *ggtreeExtra* (see 3.1). Here, we reproduce Fig.2 of (Morgan, Segata, and Huttenhower 2013) to show how to display multiple associated data to circular phylogenetic tree using *ggtreeExtra*. The data sets are provided by GraPhlAn (Asnicar et al. 2015), but they were mixed with the profiler of graphic for GraPhlAn. We has extracted and saved them as the data frame tables for corresponding geometric layer functions. These data sets contain the relative abundance of bacteria (continuous data) at different body sites (categorical data). The associated data sets were imported with *data* parameter of *geom_fruit* and displayed with the corresponding geometric function. We used heat map to display the abundance of tip species by linking *geom_tile* of *ggplot2*(Wickham 2016) (the abundance of species was mapped to the transparency of heat map using *alpha* aesthetic of *geom_tile*, then *alpha* was specified the mapping from values in the data using *scale_alpha_continuous* defined in *ggplot2*. The different body sites was mapped to the color of heat map using *fill* aesthetic of *geom_tile*, then

the *fill* was specified the mapping from levels in the data using *scale_fill_manual* defined in *ggplot2*) (Fig. S6.C), and the most abundance of species (continuous data) at specific site was visualized with bar plot by linking *geom_col* of *ggplot2* (the abundance of species was mapped to the length of bar plot using *x* aesthetic of *geom_col*, the different body sites was mapped to the color of bar plot using *fill* aesthetic of *geom_col*.) (Fig. S6.D). The outer graphic layers can be aligned automatically according to the tip labels of phylogenetic tree (tip labels were mapped to *y* value). The tree annotated by *geom_fruit* can also be annotated beforehand with other geometric layers provided by *ggtree*(Yu et al. 2017), *ggplot2* or ggplot2-based packages, such as *geom_star* of *ggstar*(Xu 2020) (Fig. S6.A), *geom_hilight* and *geom_cladelab* of *ggtree* (Fig. S6.B). The layers and outer graphic layers can be added step-by-step with + symbol (Fig. S6). More step-by-step instructions are also available at the vignette[3].

```
library(ggtreeExtra)
library(ggtree)
library(treeio)
library(tidytree)
library(ggstar)
library(ggplot2)
library(ggnewscale)
library(patchwork)

tree <- read.tree("../data/HMP_tree/hmptree.nwk")
# the abundance and types of microbes
dat1 <- read.csv("../data/HMP_tree/tippoint_attr.csv")
# the abundance of microbes at different body sites.
dat2 <- read.csv("../data/HMP_tree/ringheatmap_attr.csv")
# the abundance of microbes at the body sites of greatest prevalence.
dat3 <- read.csv("../data/HMP_tree/barplot_attr.csv")

# adjust the order
dat2$Sites <- factor(dat2$Sites, levels=c("Stool (prevalence)", "Cheek (prevalence)",
                                          "Plaque (prevalence)","Tongue (prevalence)",
                                          "Nose (prevalence)", "Vagina (prevalence)",
                                          "Skin (prevalence)"))

dat3$Sites <- factor(dat3$Sites, levels=c("Stool (prevalence)", "Cheek (prevalence)",
                                          "Plaque (prevalence)", "Tongue (prevalence)",
                                          "Nose (prevalence)", "Vagina (prevalence)",
                                          "Skin (prevalence)"))

# extract the clade label information. Because some nodes of tree are annotated to genera,
# which can be displayed with high light using ggtree.
# This is optional, since the node information are always not present.
nodeids <- nodeid(tree, tree$node.label[nchar(tree$node.label)>4])
nodedf <- data.frame(node=nodeids)
nodelab <- gsub("[\\.0-9]", "", tree$node.label[nchar(tree$node.label)>4])

# The layers of clade and hightlight (optional)
poslist <- c(1.6, 1.4, 1.6, 0.8, 0.1, 0.25, 1.6, 1.6, 1.2, 0.4,
             1.2, 1.8, 0.3, 0.8, 0.4, 0.3, 0.4, 0.4, 0.4, 0.6,
             0.3, 0.4, 0.3)
labdf <- data.frame(node=nodeids, label=nodelab, pos=poslist)

# The circular layout tree.
p <- ggtree(tree, layout="fan", size=0.15, open.angle=5)

# add tip points with geom_star of ggstar
p <- p %<+% dat1 +
     geom_star(
          mapping=aes(fill=Phylum, starshape=Type, size=Size),
          starstroke=0.05
```

[3]http://bioconductor.org/packages/release/bioc/vignettes/ggtreeExtra/inst/doc/ggtreeExtra.html

```
    ) +
    scale_fill_manual(
        values=c("#FFC125","#87CEFA","#7B68EE","#808080","#800080",
                 "#9ACD32","#D15FEE","#FFC0CB","#EE6A50","#8DEEEE",
                 "#006400","#800000","#B0171F","#191970"),
        guide=guide_legend(keywidth = 0.5, keyheight = 0.5, order=1,
        override.aes=list(starshape=15)),
        na.translate=FALSE
    ) +
    scale_starshape_manual(
        values=c(15, 1),
        guide=guide_legend(keywidth = 0.5, keyheight = 0.5, order=2),
        na.translate=FALSE
    ) +
    scale_size_continuous(
        range = c(0.5, 1.5),
        guide = guide_legend(keywidth = 0.5, keyheight = 0.5, order=3,
        override.aes=list(starshape=15))
    ) +
    new_scale_fill() +
    ggtitle("A") +
    theme(legend.position="none")

# optional for high light and clade labels

p1 <- p +
     geom_hilight(
        data=nodedf,
        mapping=aes(node=node),
        extendto=6.8,
        alpha=0.3,
        fill="grey",
        color="grey50",
        size=0.05
     ) +
     geom_cladelab(
        data=labdf,
        mapping=aes(node=node, label=label, offset.text=pos),
        barsize=NA,
        fontsize=0.7,
        angle="auto",
        hjust=0.5,
        horizontal=FALSE,
        fontface="italic"
     ) +
     ggtitle("B")

p2 <- p1 +
     geom_fruit(
         data=dat2,
         geom=geom_tile,
         mapping=aes(y=ID, x=Sites, alpha=Abundance, fill=Sites),
         color = "grey50",
         offset = 0.04,
         size = 0.02
     )+
     scale_alpha_continuous(
         range=c(0, 1),
         guide=guide_legend(keywidth = 0.3, keyheight = 0.3, order=5)
     ) +
```

```
      scale_fill_manual(
          values=c("#0000FF","#FFA500","#FF0000","#800000",
                   "#006400","#800080","#696969"),
          guide=guide_legend(keywidth = 0.3, keyheight = 0.3, order=4)
      ) +
      ggtitle("C") +
      theme(legend.position="none")

p3 <- p2 +
      geom_fruit(
          data=dat3,
          geom=geom_col,
          mapping=aes(y=ID, x=HigherAbundance, fill=Sites),
          pwidth=0.38,
          orientation="y",
          position=position_stackx(),
      )+
      geom_treescale(fontsize=1.2, linesize=0.3, x=4.9, y=0.1) +
      ggtitle("D") +
      theme(legend.position="none")

p4 <- (p + p1 + plot_layout(width=c(3.4,4)))/ (p2 + p3 + plot_layout(width=c(3.4,4))) +
      plot_layout(heights=c(3,4))

p4
```

3.3 Annotating phylogenetic tree fully with multiple associated data

Since *ggtreeExtra* supports annotation of multiple layouts tree (tree and geometric layers alignment) (Tab. S1 and Fig. S2 and S3). It can also link *ggtree*(Yu et al. 2017) and geometric layer functions defined in *ggplot2*(Wickham 2016) or ggplot2-based packages based on grammar of graphic (Tab. S2 in 2 and Fig. S6 in 3.2). The annotation layers on the outer of phylogenetic tree can be combined freely. So it can be easily used to visualize complex and high-dimensional associated data on the outer of phylogenetic tree including the tree containing thousands of tips. Here, we reproduce Fig 2 of (Asnicar et al. 2015). The phylogenetic tree also was built using a part of 3737 microbes (Segata et al. 2013). The associated data sets have also been extracted from configure file and saved as the data frame types for corresponding geometric layer functions. They contains present or not (discrete data) and type (discrete data) of target gene, the type (discrete data) and capability (continuous data) of fatty acid metabolism, and the length of microbes genome. The present or not of different target gene was visualized with first inner heat map (the type of target gene was mapped to color of heat map, and the subtype of target gene was mapped to the x value(take on numerical values)). The type and capability of fatty acid metabolism was also visualized with middle heat map (the type was mapped to color of heat map, the capability of fatty acid metabolism was mapped to the transparency of heat map). The length of genome was displayed with bar chart (the length of genome was mapped to length of bar, the phylum information of genome was mapped to the color of bar). (Fig. S7). Notably, the same colors of Proteobacteria and Spirochaetes were found in the original configure file(Asnicar et al. 2015). We tried to separate the tips from Proteobacteria and Spirochaetes. Unfortunately, the phylum information of tips was replaced color code, so color of Proteobacteria and Spirochaetes is also identical in Fig. S7. This show that providing configure files mixed associated data and profiler of graphic layer is tedious and error-prone since the profiler of graphic layer (such like color or size) should be consistent for the same information.

```
library(ggtreeExtra)
library(ggtree)
library(treeio)
library(tidytree)
library(ggstar)
library(ggplot2)
library(ggnewscale)

tree <- read.tree("../data/kegg/kegg.nwk")
# The attributes of tip point
dt1 <- read.csv("../data/kegg/tippoint_attr.csv")
```

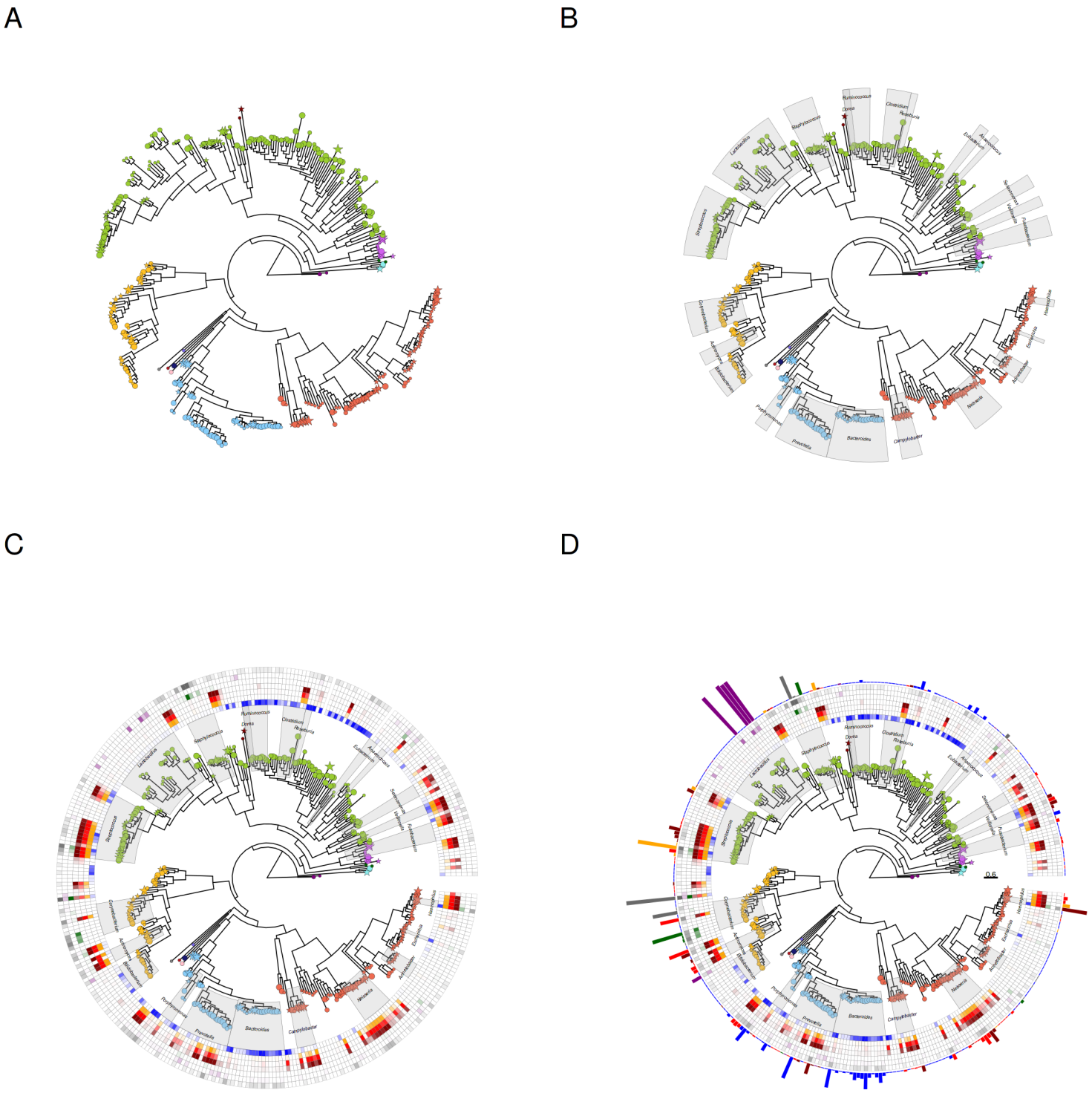

Fig. S6: The abundance of microbes at different sites of human. The shape of tip labels indicated the commensal microbes or potential pathogens. The transparency of heat map indicates the abundance of microbes, and colors of heat map indicate the different sites of human. The bar plot indicates the relative abundance at body site of the most abundance. Fig.S6.A contains the tree layer and tip point layer; Fig.S6.B is from Fig.S6.A added high light layer and clade label layer; Fig.S6.C is from Fig.S6.B added a heatmap layer; Fig.S6.D is from Fig.S6.C added a bar chart layer. This example shows the abilities of annotate phylogenetic tree through grammar of graphic using ggtree and ggtreeExtra.

```
# The attributes of first ring
dt2 <- read.csv("../data/kegg/firstring_attr.csv")
# The attributes of second ring
dt3 <- read.csv("../data/kegg/secondring_attr.csv")
# The attrihutes of bar plot
dt4 <- read.csv("../data/kegg/barplot_attr.csv")
```

```
#reorder the Phyla column
dt1$Phyla <- factor(dt1$Phyla, levels=c("Actinobacteria","Aquificae","Bacteroidetes",
                                        "Chlamydiae","Chlorobi","Chloroflexi","Crenarchaeota",
                                        "Cyanobacteria","Euryarchaeota","Firmicutes","Proteobacteria",
                                        "Spirochaetes","Tenericutes","Thermi","Thermotogae","Other"))

# reorder the Type2 column
dt3$Type2 <- factor(dt3$Type2, levels=c("FA synth init", "FA synth elong",
                                        "acyl-CoA synth", "beta-Oxidation",
                                        "Ketone biosynth"))

dt4$Phyla <- factor(dt4$Phyla, levels=c("Actinobacteria","Aquificae","Bacteroidetes",
                                        "Chlamydiae","Chlorobi","Chloroflexi","Crenarchaeota",
                                        "Cyanobacteria","Euryarchaeota","Firmicutes","Proteobacteria",
                                        "Spirochaetes","Tenericutes","Thermi","Thermotogae","Other"))
# extract node label for the clade layers
nodelab <- tree$node.label[nchar(tree$node.label)>0]
nodeids <- nodeid(tree, nodelab)

# the position of clade label
textex <- c(1.0, 0.4, 0.2, 1.4, 1.4, 0.4, 1.4, 1.4, 0.4, 0.4,
            0.8, 1, 0.6, 0.6, 0.4, 0.3, 0, 0.4, 0.1, 0.25,
            0.2, 0.3, 0.8, 0.8, 0.8, 0.6, 2.4)
# optional for clade layers
cladelabels <- mapply(function(x, y, z){geom_cladelabel(node=x, label=y, barsize=NA, extend=0.3,
                                                        offset.text=z, fontsize=1.2, angle="auto",
                                                        hjust=0.5, horizontal=FALSE, fontface="italic")},
                                      nodeids, nodelab, textex, SIMPLIFY=FALSE)

# high light layers
fills <- c("#808080", "#808080", "#808080", "#808080", "#808080",
           "#191970", "#87CEFA", "#FFC125", "#B0171F", "#B0171F",
           "#B0171F", "#B0171F", "#B0171F", "#B0171F", "#B0171F",
           "#B0171F", "#B0171F", "#B0171F", "#B0171F", "#B0171F",
           "#B0171F", "#B0171F", "#9ACD32", "#9ACD32", "#9ACD32",
           "#006400", "#800000")
# optional for hight light
highlights <- mapply(function(x, y){geom_hilight(node=x, extendto=5.8, alpha=0.3,
                                                 fill=y, color=y, size=0.05)},
                     nodeids, fills, SIMPLIFY=FALSE)

# to reproduce the original figures, we use the same colors.
# uses can custom set it.
colors <- c("#9ACD32", "#EE6A50", "#87CEFA", "#FFC125", "#D15FEE", "#8DEEEE", "#800000",
            "#006400", "#800080", "#808080", "#B0171F", "#B0171F", "#191970", "#7B68EE",
            "#00CD00", "Black")

p1 <- ggtree(tree, layout="circular", size=0.1) + highlights

p2 <- p1 +
     geom_fruit(
         data=dt1,
         geom=geom_point,
         mapping=aes(y=ID, fill=Phyla),
         shape=21,
         size=1.2,
         stroke=0.05,
         position="identity",
         show.legend=FALSE
```

```
    )+
    scale_fill_manual(values=colors) +
    cladelabels +
    new_scale_fill()

p3 <- p2 +
      geom_fruit(
        data=dt2,
        geom=geom_tile,
        mapping=aes(y=ID, x=ring, fill=Type1),
        offset=-0.02,
        pwidth=0.14,
        addbrink=TRUE
    ) +
    scale_fill_manual(
        name="ATP synthesis",
        values=c("#339933", "#dfac03"),
        guide=guide_legend(keywidth=0.5, keyheight=0.5, order=1)
    ) +
    new_scale_fill()

p4 <- p3 +
      geom_fruit(
          data=dt3,
          geom=geom_tile,
          mapping=aes(y=ID, alpha=Abundance, x=Type2, fill=Type2),
          offset=0.001,
          pwidth=0.18
      ) +
      scale_fill_manual(
          name="Fatty Acid metabolism",
          values=c("#b22222", "#005500", "#0000be", "#9f1f9f", "#793a07"),
          guide=guide_legend(keywidth=0.5, keyheight=0.5, order=2)
      ) +
      scale_alpha_continuous(range=c(0, 0.4),
                             guide=guide_legend(keywidth=0.5, keyheight=0.5,order=3)) +
      new_scale_fill()

p5 <- p4 +
      geom_fruit(data=dt4,
           geom=geom_bar,
           mapping=aes(y=ID, x=Length, fill=Phyla),
           stat="identity",
           orientation="y",
           pwidth=0.3,
           position=position_dodgex()) +
      scale_fill_manual(values=colors,
                        guide=guide_legend(keywidth=0.5, keyheight=0.5, order=4)) +
      geom_treescale(fontsize=1.2, linesize=0.3) +
      theme(legend.position=c(0.95, 0.5),
            legend.background=element_rect(fill=NA),
            legend.title=element_text(size=7),
            legend.text=element_text(size=6),
            legend.spacing.y = unit(0.02, "cm"))
p5
```

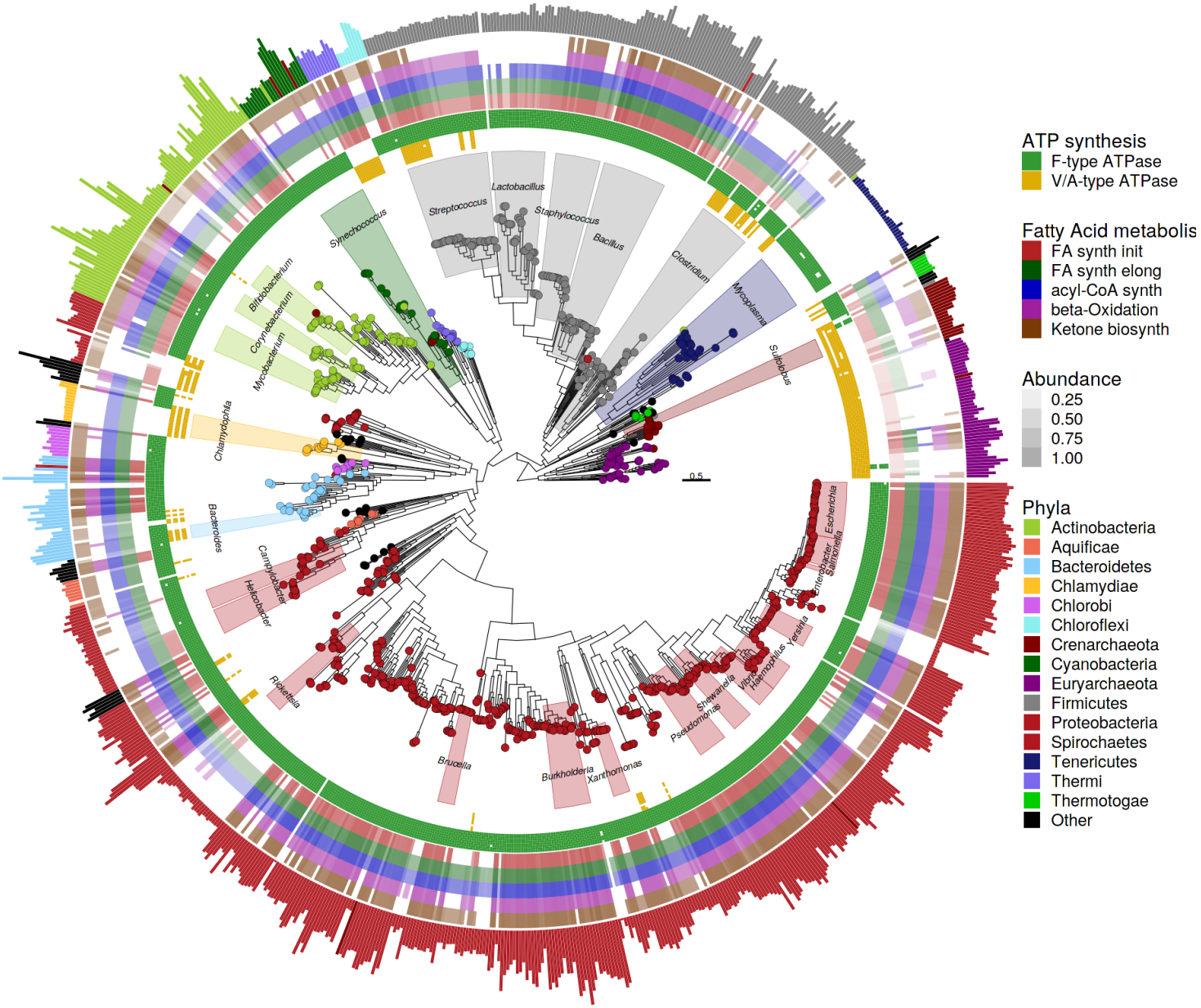

Fig. S7: The phylogenetic tree built using in (Segata et al. 2013) using 963 microbial genomes (a part of 3737 microbes (Segata et al. 2013)). The first and second ring heat map were built with discrete data, it represents the presence or absence of each module, the other heat map rings were created with continuous data, the transparency represents the capability of fatty acid metabolism, the different colors represent the types of fatty acid metabolism. The length of bar represents the genome length of corresponding microbes.

3.4 Annotating associated data to the phylogenetic tree combined chord diagram

As the mentioned above, *ggtreeExtra* can link *ggtree*(Yu et al. 2017) and geometric layer functions defined *ggplot2*(Wickham 2016) or other ggplot2 extension packages (Tab. S2). It also developed based on grammar of graphic (Fig. S6). The geometric layers of *ggtree*(Yu et al. 2017) and *ggplot2* or other ggplot2 extension packages can be combined freely using *ggtreeExtra* (Fig. S7), and the layers can be added, deleted or modified. *ggtreeExtra* is more flexible and universal than other tools, since it can also integrate more geometric layers and support more layouts for tree annotation (Tab. S1). These features allow *ggtreeExtra* to better explore phylogenetic patterns behind multi-dimensional data. For example, the Fig.1b and Fig.2 of (Helfrich et al. 2018) show the directional interactions and the biosynthetic potential of isolates from *Arabidopsis* leaf microbiome in phylogenetic tree using GraPhlAn(Asnicar et al. 2015). However, because GraPhlAn (Asnicar et al. 2015) is not general, it only support annotation of circular layout tree and support few geometric layers, some phylogenetic patterns behind multi-dimensional data were not be found easily. Here, we can use *ggtree* and *ggtreeExtra* to combine the total information of Fig.1b and Fig.2 of (Helfrich et al. 2018). In details, the interactions of isolates are visualized with chord diagram using *geom_taxalink* of *ggtree*(Yu et al. 2017). The biosynthetic potential of isolates are displayed with heat map by linking *geom_tile* of *ggplot2*(Wickham 2016)

(the number of target gene (continuous data) was mapped to the transparency of heat map, the type of target gene (discrete data) was mapped to the x value and color of heat map). The number of interactions of inhibitions or sensitivities per strain is displayed with stacked bar by linking *geom_bar* of *ggplot2*(Wickham 2016) (the number of interactions (continuous data) was mapped to the x value (length) of bar plot, and the type of interactions (discrete data) was mapped to the color of bar plot.). We found some strains from Firmicutes and from Grammaproteobacteria have more inhibitor interactions. However, many strains from Alphaproteobacteria and Betateobacteria prefer the interaction of sensitivity. In addition, These strains that prefer the interactions of sensitivity might be have more BGCs (biosynthesis gene clusters) from ribosomally synthesized and post-translationally modified peptide (*RiPP*). Notably, other tools do not support annotation of phylogenetic tree combined chord diagram to show relationship data, such as correlation data of species or genes, horizontal gene transfer and syntenic linkage. However, *ggtreeExtra* support this feature (Fig. S8), because of its unique design (Fig. S1). This proves *ggtreeExtra* is more powerful and universal than other tools again.

```
library(ggtree)
library(ggtreeExtra)
library(ggplot2)
library(MicrobiotaProcess)
library(ggstar)
library(ggnewscale)
library(grid)

alltax <- read.csv("../data/Arabidopsis_leaf_microbiome/all_stain_taxonomy.csv")
linktab <- read.csv("../data/Arabidopsis_leaf_microbiome/Interaction_link_tab.csv")
weighttab <- read.csv("../data/Arabidopsis_leaf_microbiome/Interaction_weight.csv")
tippoint <- read.csv("../data/Arabidopsis_leaf_microbiome/stain_tippoint.csv")
BGCsda <- read.csv("../data/Arabidopsis_leaf_microbiome/BGCs_heatmap.csv")

tippoint$Taxa <- factor(tippoint$Taxa,
                        levels=c("Actinobacteria",
                                 "Bacteroidetes",
                                 "Firmicutes",
                                 "Deinococcus-Thermus",
                                 "Alphaproteobacteria",
                                 "Betaproteobacteria",
                                 "Gammaproteobacteria"
                                )
                        )
tippoint$names <- gsub("s__Leaf","",tippoint$Isolation)

BGCsda$BGCs <- factor(BGCsda$BGCs,
                      levels=c("modular.PKS",
                               "modular.PKS.NRPS.hybrid",
                               "non_modular.PKS", "NRPS",
                               "RiPP",
                               "Quorum.sensing",
                               "terpene",
                               "other"
                              )
                     )
BGCsda$Count <- log10(BGCsda$Count+1)
BGCsda$Count <- ifelse(BGCsda$Count==0, NA, BGCsda$Count)

trda <- convert_to_treedata(alltax)
p <- ggtree(
         trda,
         layout="inward_circular",
         size=0.2,
         xlim=c(18,NA)
     )
p <- p %<+% tippoint

p1 <- p +
```

```
      geom_tippoint(
          mapping=aes(
                      color=Taxa,
                      shape=Level
                      ),
          size=1,
          alpha=0.8
      ) +
      scale_color_manual(values=c("#EF3B2C", "#1D91C0", "#FEB24C", "grey60",
                                  "#7FBC41", "#4D9221", "#276419"),
                         guide=guide_legend(
                                   keywidth=0.5,
                                   keyheight=0.5,
                                   order=2,
                                   override.aes=list(shape=c("Actinobacteria"=20,
                                                             "Bacteroidetes" =20,
                                                             "Firmicutes"    =20,
                                                             "Deinococcus-Thermus" =20,
                                                             "Alphaproteobacteria" =18,
                                                             "Betaproteobacteria"  =18,
                                                             "Gammaproteobacteria" =18
                                                             ),
                                                     size=2
                                                  ),
                                   na.translate=TRUE
                                )
      ) +
      scale_shape_manual(values=c("Phylum"=20, "Class"=18), guide="none" )

p2 <- p1 +
      new_scale_color() +
      geom_taxalink(
          data=linktab,
          mapping=aes(
                      taxa1=Inhibitor,
                      taxa2=Sensitive,
                      color=Interaction
                  ),
          alpha=0.6,
          offset=0.1,
          size=0.15,
          ncp=10,
          hratio=1,
          arrow=grid::arrow(length = unit(0.005, "npc"))
      ) +
      scale_colour_manual(values=c("chocolate2", "#3690C0", "#009E73"),
                          guide=guide_legend(
                                    keywidth=0.8, keyheight=0.5,
                                    order=1, override.aes=list(alpha=1, size=0.5)
                          )
      )

p3 <- p2 +
      geom_fruit(
          data=BGCsda,
          geom=geom_tile,
          mapping=aes(y=Strain, x=BGCs, alpha=Count, fill=BGCs),
          offset=-0.9,
          pwidth=1,
          size=0.02,
```

```
        color = "grey50"
    ) +
    scale_alpha_continuous(range=c(0.1, 1),
                           name=bquote(paste(Log[10],"(",.("Count+1"), ")")),
                           guide=guide_legend(keywidth = 0.4, keyheight = 0.4, order=4)
    ) +
    scale_fill_manual(
        values=c("#66C2A5", "#FC8D62", "#8DA0CB", "#E78AC3",
                 "#A6D854", "#FFD92F", "#E5C494", "#B3B3B3"),
        guide=guide_legend(keywidth = 0.4, keyheight = 0.4, order=3)
    )

p4 <- p3 +
    geom_tiplab(
        mapping=aes(label=names),
        align=TRUE,
        size=1,
        linetype=NA,
        offset=7.8
    )

p5 <- p4 +
    new_scale_fill() +
    geom_fruit(
        data=weighttab,
        geom=geom_bar,
        mapping=aes(
                    x=value,
                    y=Strain,
                    fill=Number
                ),
        stat="identity",
        orientation="y",
        offset=0.48,
        pwidth=2,
        axis.params=list(
                        axis="x",
                        text.angle=-45,
                        hjust=0,
                        vjust=0.5,
                        nbreak=4,
                        )
    ) +
    scale_fill_manual(
        values=c("#E41A1C", "#377EB8", "#4DAF4A", "#984EA3"),
        guide=guide_legend(keywidth=0.5,keyheight=0.5,order=5)
    ) +
    theme(
          legend.background=element_rect(fill=NA),
          legend.title=element_text(size=6.5),
          legend.text=element_text(size=5),
          legend.spacing.y = unit(0.02, "cm"),
          legend.margin=margin(0.1, 0.9, 0.1,-0.9, unit="cm"),
          legend.box.margin=margin(0.1, 0.9, 0.1, -0.9, unit="cm"),
          plot.margin = unit(c(-1.2, -1.2, -1.2, 0.1),"cm")
    )
p5
```

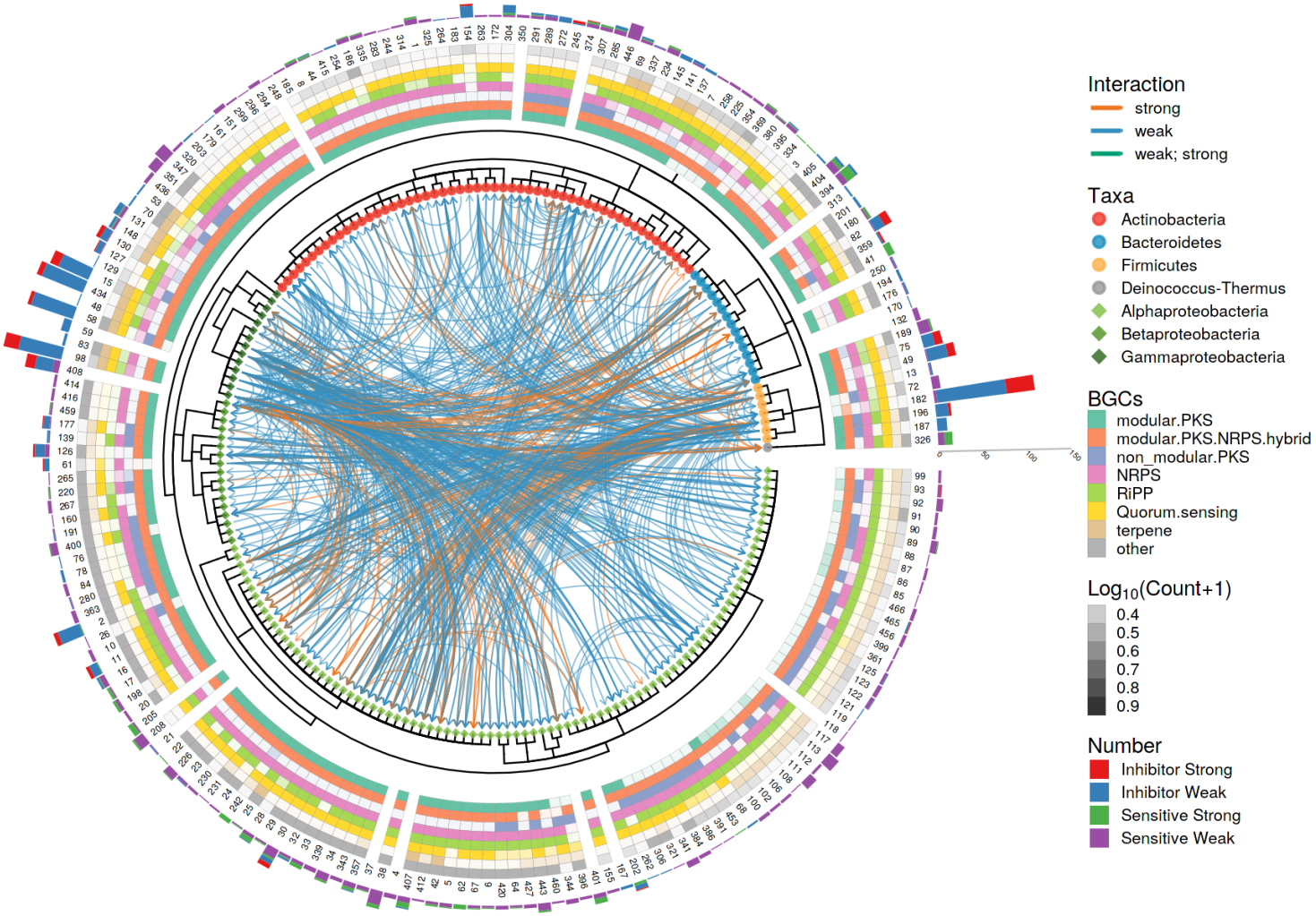

Fig. S8: The Phylogenetic tree of isolates from *Arabidopsis* leaf microbiome (Bai et al. 2015). The lines in inner represent the directional inhibitory interactions, the direction of arrow represents the directional inhibitory interaction, and the color of line represents the weak or strong interactions. The color of tip point represents the taxonomy annotation, circle represents the phylum, square represents the class from Proteobacteria. The heat map of external ring represents the number of detected BGCs (biosynthesis gene clusters). The ring from inner to outside represents the Modular polyketide synthase, Modular PKS non-ribosomal peptide synthetase (NRPS) hybrid, Non modular PKS, NRPS, ribosomally synthesized and post-translationally modified peptide (RiPP), Quorum sensing, Terpene, and Others. The external bar represent the number of interactions of inhibitions or sensitivities per strain.

4 Summary

There are some tools depending on python or other platform to annotate phylogenetic tree with associated data, but they are not general and still have some shortcomings. For example, they support less layouts and less geometric layers for tree annotation (tree and geometric layer alignment) (Tab. S1). And many tools need user providing configure file mixed associated data and profiler of graphic, which make them tedious and error-prone since the profiler of graphic layer (such like color or size) should be consistent for the same information (Fig. S7 in 3.3). There features make these tools not universal. It is worth mentioning that *ggtree* has also provided *facet_plot*(*geom_facet*) to annotated phylogenetic tree with external data (Yu et al. 2018). This function is also a universal function with many unique features (Yu et al. 2018). But it can't not work with circular layout tree, which is an efficient way to visualize multi-dimensional dataset and phylogenetic tree, since circular layout can reduce space and make the graph more compact. Fortunately, *geom_fruit* of *ggtreeExtra* inherits the design concept of *facet_plot* (*geom_facet*) of *ggtree* (Yu et al. 2018) and supports more layouts for tree annotation (tree and graph alignment), So *geom_fruit* has also some features contained in *facet_plot*(*geom_facet*). For example:

1. No restriction of input data types or how the data should be plotted in *geom_fruit*, it depends on the data types of various geometric functions. (Tab. S2 and 3.1)
2. Associated data integrated by %<+% can also be used in "geom_fruit". (Fig. S5)
3. Combining different *geom* functions to visualize associated data is supported. (Fig. S7)

4. Supporting ggplot object subplot or image. (Fig. S4)
5. Supporting grammar of graphic. (Fig. S6)

In addition, the phylogenetic tree annotated by *ggtreeExtra* can also be converted to another layout tree (Fig. S2). Moreover, although other tools can also integrate similar geometric layer to *ggtreeExtra*, such as bar plot, box plot, *ggtreeExtra* is more powerful and flexible since it inherits the features of corresponding **geom** functions (Fig. S3 and S4). Furthermore, *ggtreeExtra* also supports annotation of phylogenetic combined chord diagram, which is an efficient way to display relationship data, such as correlation data of species or genes, horizontal gene transfer and syntenic linkage. (Fig. S8) These features is also not available in other tools.

References

Aphalo, Pedro J. 2020. *Ggpmisc: Miscellaneous Extensions to 'Ggplot2'*. https://CRAN.R-project.org/package=ggpmisc.

Asnicar, Francesco, George Weingart, Timothy L Tickle, Curtis Huttenhower, and Nicola Segata. 2015. "Compact Graphical Representation of Phylogenetic Data and Metadata with Graphlan." *PeerJ* 3: e1029. https://doi.org/10.7717/peerj.1029.

Bai, Yang, Daniel B Müller, Girish Srinivas, Ruben Garrido-Oter, Eva Potthoff, Matthias Rott, Nina Dombrowski, et al. 2015. "Functional Overlap of the Arabidopsis Leaf and Root Microbiota." *Nature* 528 (7582): 364–69. https://doi.org/10.1038/nature16192.

Berger, Simon A., Denis Krompass, and Alexandros Stamatakis. 2011. "Performance, Accuracy, and Web Server for Evolutionary Placement of Short Sequence Reads under Maximum Likelihood." *Systematic Biology* 60 (3): 291–302. https://doi.org/10.1093/sysbio/syr010.

Boussau, Bastien, Gergely J. Szöllősi, Laurent Duret, Manolo Gouy, Eric Tannier, and Vincent Daubin. 2013. "Genome-Scale Coestimation of Species and Gene Trees." *Genome Research* 23 (2): 323–30. https://doi.org/10.1101/gr.141978.112.

Drummond, Alexei J, and Andrew Rambaut. 2007. "BEAST: Bayesian Evolutionary Analysis by Sampling Trees." *BMC Evolutionary Biology* 7 (1): 1–8. https://doi.org/10.1186/1471-2148-7-214.

FC, Mike. 2020. *Ggpattern: Geoms with Patterns*.

Helfrich, Eric J. N., Christine M. Vogel, Reiko Ueoka, Martin Schäfer, Florian Ryffel, Daniel B. Müller, Silke Probst, Markus Kreuzer, Jörn Piel, and Julia A. Vorholt. 2018. "Bipartite Interactions, Antibiotic Production and Biosynthetic Potential of the Arabidopsis Leaf Microbiome." Journal Article. *Nature Microbiology* 3 (8): 909–19. https://doi.org/10.1038/s41564-018-0200-0.

Höhna, Sebastian, Michael J. Landis, Tracy A. Heath, Bastien Boussau, Nicolas Lartillot, Brian R. Moore, John P. Huelsenbeck, and Fredrik Ronquist. 2016. "RevBayes: Bayesian Phylogenetic Inference Using Graphical Models and an Interactive Model-Specification Language." *Systematic Biology* 65 (4): 726–36. https://doi.org/10.1093/sysbio/syw021.

Matsen, Frederick A, Robin B Kodner, and E Virginia Armbrust. 2010. "Pplacer: Linear Time Maximum-Likelihood and Bayesian Phylogenetic Placement of Sequences onto a Fixed Reference Tree." *BMC Bioinformatics* 11 (1): 538. https://doi.org/10.1186/1471-2105-11-538.

Mirarab, Siavash, and Tandy Warnow. 2015. "ASTRAL-II: Coalescent-Based Species Tree Estimation with Many Hundreds of Taxa and Thousands of Genes." *Bioinformatics (Oxford, England)* 31 (12): i44–52. https://doi.org/10.1093/bioinformatics/btv234.

Morgan, Xochitl C., Nicola Segata, and Curtis Huttenhower. 2013. "Biodiversity and Functional Genomics in the Human Microbiome." *Trends in Genetics* 29 (1): 51–58. https://doi.org/10.1016/j.tig.2012.09.005.

Pond, Sergei L. Kosakovsky, Simon D. W. Frost, and Spencer V. Muse. 2005. "HyPhy: Hypothesis Testing Using Phylogenies." *Bioinformatics* 21 (5): 676–79. https://doi.org/10.1093/bioinformatics/bti079.

Segata, Nicola, Daniela Börnigen, Xochitl C. Morgan, and Curtis Huttenhower. 2013. "PhyloPhlAn Is a New Method for Improved Phylogenetic and Taxonomic Placement of Microbes." Journal Article. *Nature Communications* 4 (1): 2304. https://doi.org/10.1038/ncomms3304.

Smith, Garrett J, and Kelly C Wrighton. 2019. "Metagenomic Approaches Unearth Methanotroph Phylogenetic and Metabolic Diversity." *Curr Issues Mol Biol* 33: 57–84. https://doi.org/10.21775/9781912530045.03.

Stamatakis, Alexandros. 2014. "RAxML Version 8: A Tool for Phylogenetic Analysis and Post-Analysis of Large Phylogenies." *Bioinformatics*, January, btu033. https://doi.org/10.1093/bioinformatics/btu033.

Wang, Li-Gen, Tommy Tsan-Yuk Lam, Shuangbin Xu, Zehan Dai, Lang Zhou, Tingze Feng, Pingfan Guo, et al. 2020. "Treeio: An R Package for Phylogenetic Tree Input and Output with Richly Annotated and Associated Data." *Molecular Biology and Evolution* 37 (2): 599–603. https://doi.org/10.1093/molbev/msz240.

Wickham, Hadley. 2016. *Ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. https://ggplot2.tidyverse.org.

Wilke, Claus O. 2020. *Ggridges: Ridgeline Plots in 'Ggplot2'*. https://CRAN.R-project.org/package=ggridges.

Wilkinson, Leland. 2012. "The Grammar of Graphics." In *Handbook of Computational Statistics: Concepts and Methods*, edited by James E. Gentle, Wolfgang Karl Härdle, and Yuichi Mori, 375–414. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-21551-3_13.

Xu, Shuangbin. 2020. *Ggstar: Star Layer for 'Ggplot2'*. https://CRAN.R-project.org/package=ggstar.

Yang, Ziheng. 2007. "PAML 4: Phylogenetic Analysis by Maximum Likelihood." *Molecular Biology and Evolution* 24 (8): 1586–91. https://doi.org/10.1093/molbev/msm088.

Yu, Guangchuang, Tommy Tsan-Yuk Lam, Huachen Zhu, and Yi Guan. 2018. "Two Methods for Mapping and Visualizing Associated Data on Phylogeny Using Ggtree." *Molecular Biology and Evolution* 35 (2): 3041–3. https://doi.org/10.1093/molbev/msy194.

Yu, Guangchuang, David Smith, Huachen Zhu, Yi Guan, and Tommy Tsan-Yuk Lam. 2017. "Ggtree: An R Package for Visualization and Annotation of Phylogenetic Trees with Their Covariates and Other Associated Data." *Methods in Ecology and Evolution* 8 (1): 28–36. https://doi.org/10.1111/2041-210X.12628.