Supplemental File of

ggmsa: a visual exploration tool for multiple sequence alignment and associated data

Fig. S1: Examples of amino acid and nucleotide color schemes in ggmsa

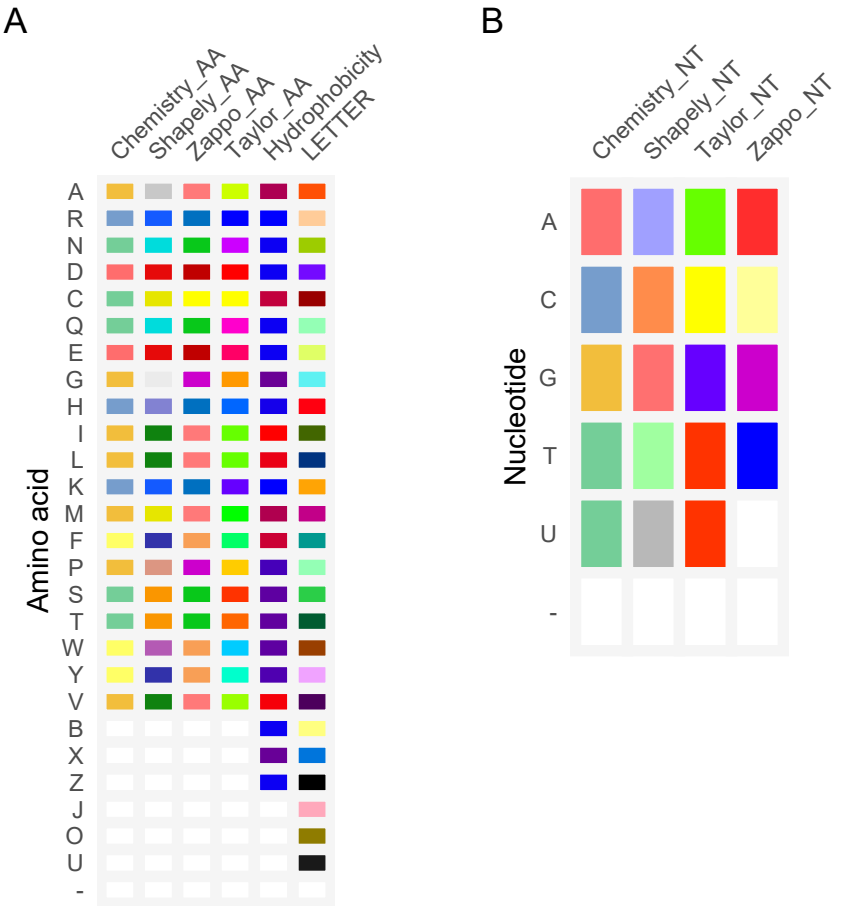


Fig. S1: (A) Examples of amino acid color schemes. Schemes are either quantitative, reflecting empirical or statistical properties of amino acids; or qualitative, reflecting physicochemical attributes. Chemistry is colored according to side-chain chemistry also used in DNASTAR applications (Burland 2000); Shapely matches the RasMol amino acid color schemes, which are, in turn, based on Robert Fletterick's Shapely models. Zappo is a qualitative scheme developed by M. Clamp. The residues are colored according to their physicochemical properties; Taylor (Taylor and W. 1997) is taken from Taylor and is also used in JalView (Waterhouse et al. 2009); Hydrophobicity colors the residues in the alignment based on the hydrophobicity table (Kyte and Doolittle 1982). B, X, Z, J, O, and U are amino acid ambiguity codes: B is aspartate or asparagine; Z is glutamate or glutamine; X, J, O, U is an unknown (or 'other'); "-" indicate a gap. (B) Examples of nucleotide color schemes used by ggmsa.

Fig. S2: miRNA sequence seed region annotation for MSA (asterisks)

`geom_seed()` helps to identify microRNA seed region by asterisks or shaded area. The seed region is a conserved heptameric sequence that is mostly situated at positions 2-7 from the miRNA 5´-end.

```
miRNA_sequences <- system.file("extdata", "seedSample.fa", package = "ggmsa")
ggmsa(miRNA_sequences, char_width = 0.5, color="Chemistry_NT") +
    geom_seed(seed = "GAGGUAG", star = TRUE) + coord_cartesian()
```

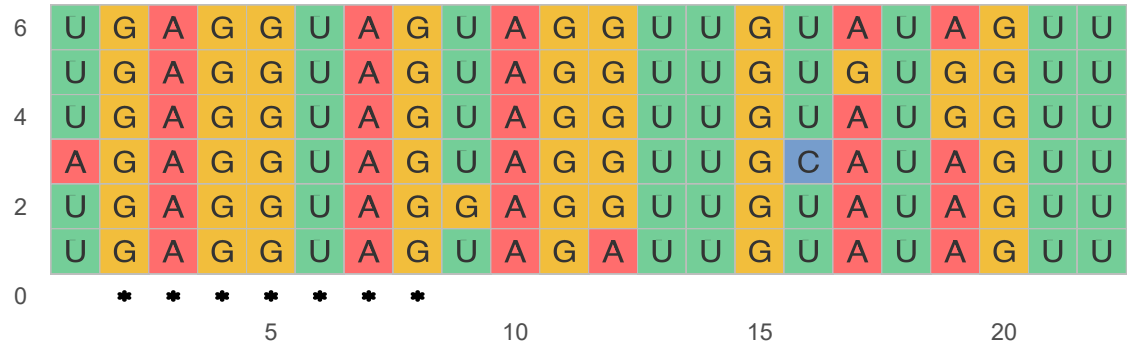


Fig. S2: Example of highlighting miRNA seed region. Asterisks are used for marking the seed region.

Fig. S3: miRNA sequence seed region annotation for MSA (The shaded block)

```
ggmsa(miRNA_sequences, char_width = 0.5, seq_name = T, none_bg = TRUE) +
    geom_seed(seed = "GAGGUAG") + coord_cartesian()
```

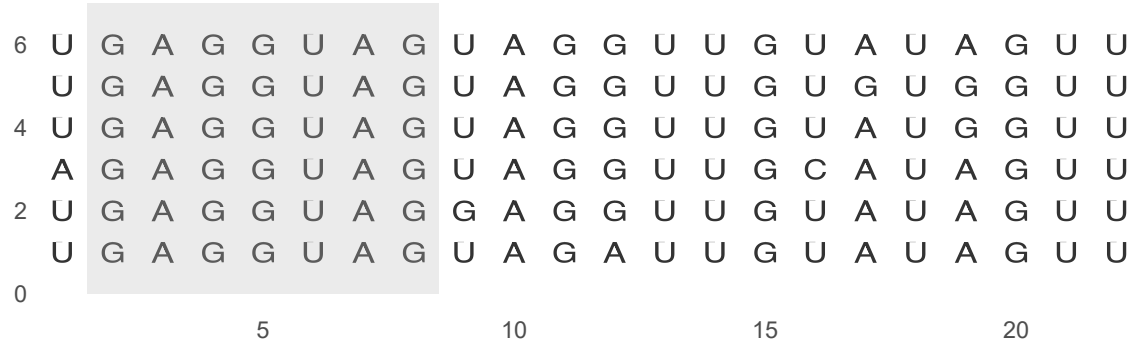


Fig. S3: Example of highlighting miRNA seed region. Annotation of sequence seed region with the shaded block

Fig. S4: The consensus sequence

```
ggmsa(protein_sequences, 300, 350, char_width = 0.5,
      seq_name = T, consensus_views = T, use_dot = T)
```

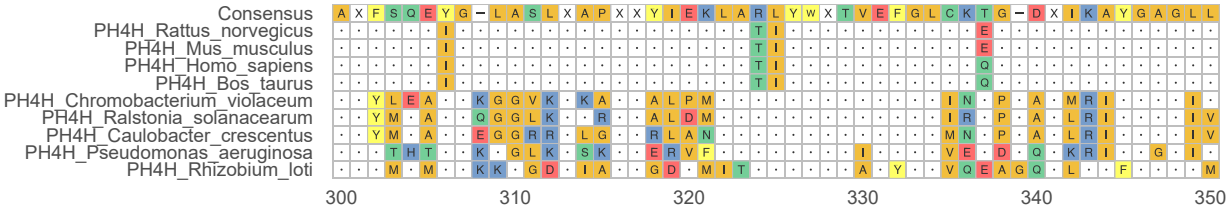


Fig. S4: Example of consensus sequence for an MSA. The consensus sequence is displayed above the alignment and shows which residues are conserved, which residues are variable. A consensus is constructed from the most frequent residues at each site.

Fig. S5: MSA view with break down layout

`facet_msa()` module allows to showing more alignment data in a restricted canvas. The long sequence was broken down and displayed in several lines.

```
# 4 fields
ggmsa(protein_sequences, start = 0, end = 400, font = NULL, color = "Chemistry_AA") +
   facet_msa(field = 100)
```

Fig. S6: MSA view with circular layout tree

A specific layout of the alignment can also be displayed by linking ggtreeExtra (Yu et al. 2021). `geom_fruit` will automatically align MSA graphs to the tree with circular layout

```
library(ggtree)
library(ggtreeExtra)
sequences <- system.file("extdata", "sequence-link-tree.fasta", package = "ggmsa")

x <- readAAStringSet(sequences)
d <- as.dist(stringDist(x, method = "hamming")/width(x)[1])
tree <- bionj(d)
data <- tidy_msa(x, 120, 200)

p1 <- ggtree(tree, layout = 'circular') +
    geom_tiplab(align = TRUE, offset = 0.545, size = 2) +
        xlim(NA, 1.3)
p1 + geom_fruit(data = data, geom = geom_msa, offset = 0,
                pwidth = 1.2, font = NULL, border = NA)
```

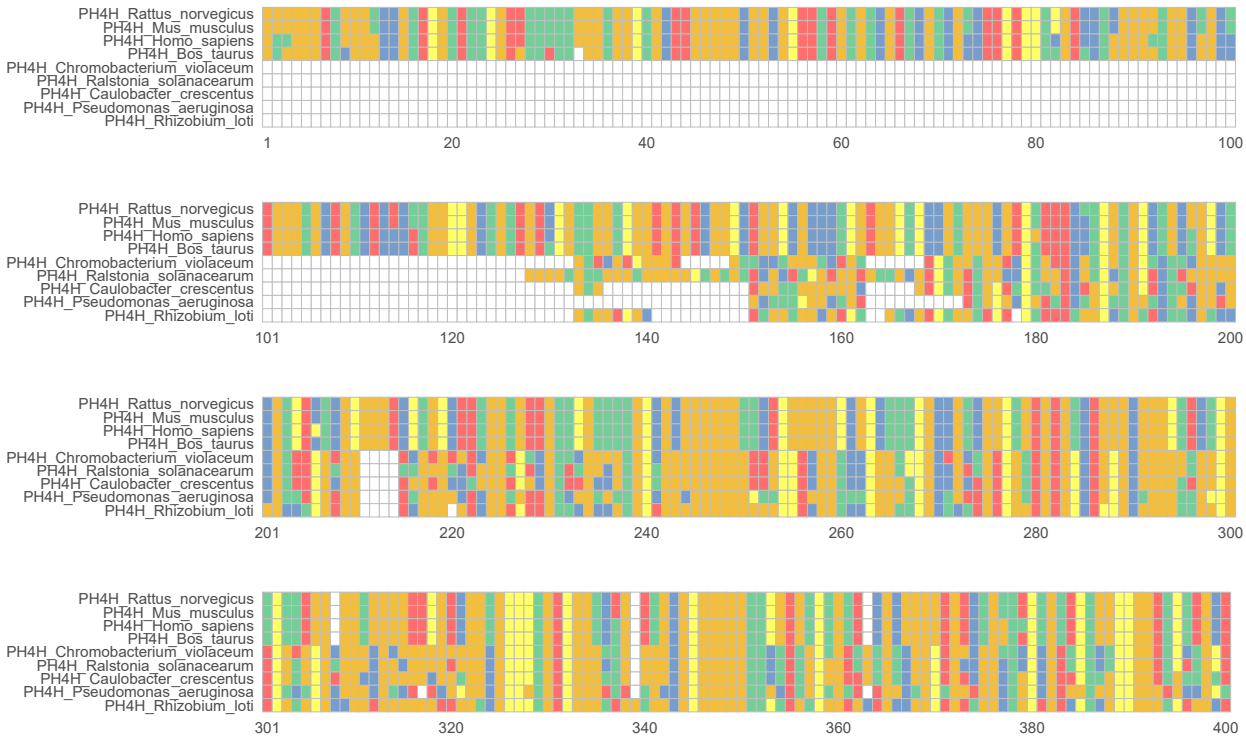


Fig. S5: The long sequence was broken down and displayed in several lines.

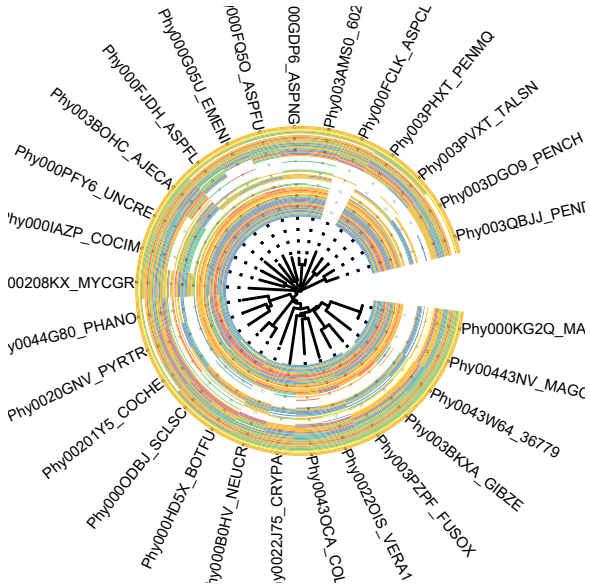


Fig. S6: Example of MSA view with circular layout phylogenetic tree.

Fig. S7: Visualizing MSA with phylogenetic tree

ggmsa supports to link ggtree (Yu et al. 2017) by `geom_facet()`. Sequence order will automatically align with left nodes.

```
x <- readAAStringSet(protein_sequences)
d <- as.dist(stringDist(x, method = "hamming")/width(x)[1])
library(ape)
tree <- bionj(d)
library(ggtree)
p <- ggtree(tree) + geom_tiplab()

data = tidy_msa(x)
p + geom_facet(geom = geom_msa, data = data,  panel = 'msa',
               font = NULL, border = NA, color = "Chemistry_AA") +
    xlim_tree(1)
```

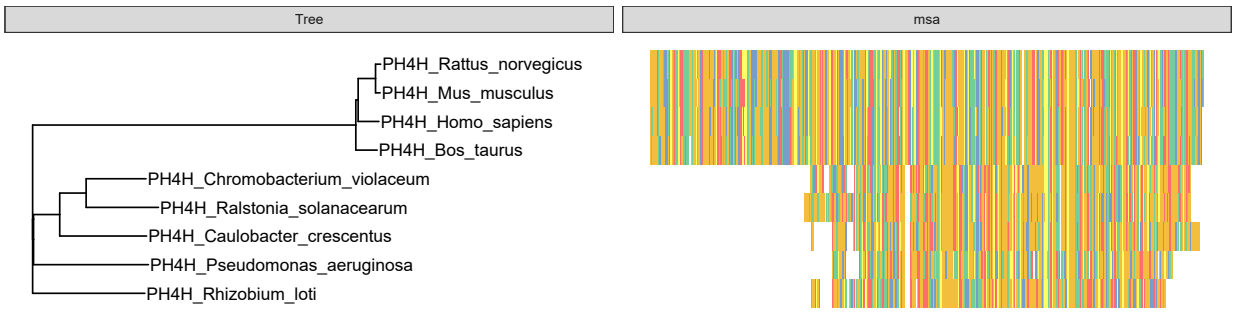


Fig. S7: A tree resulting from the application of phylogenetic analysis to the alignment.

Fig. S8: Re-designated residues order in Sequence bundle

```
negative <-  system.file("extdata", "Gram-negative_AKL.fasta",
                         package = "ggmsa")
positive <-  system.file("extdata", "Gram-positive_AKL.fasta",
                         package = "ggmsa")

ggSeqBundle(list(negative, positive),
                alpha = 0.1,
                bundle_color = c("#FC8D62","#8DA0CB"),
                lev_molecule = c("-", "W", "Y", "R", "F", "H", "M", "E",
                                 "K", "Q", "D", "N", "L", "I", "C", "T",
                                 "V", "P", "S", "A", "G"))
```

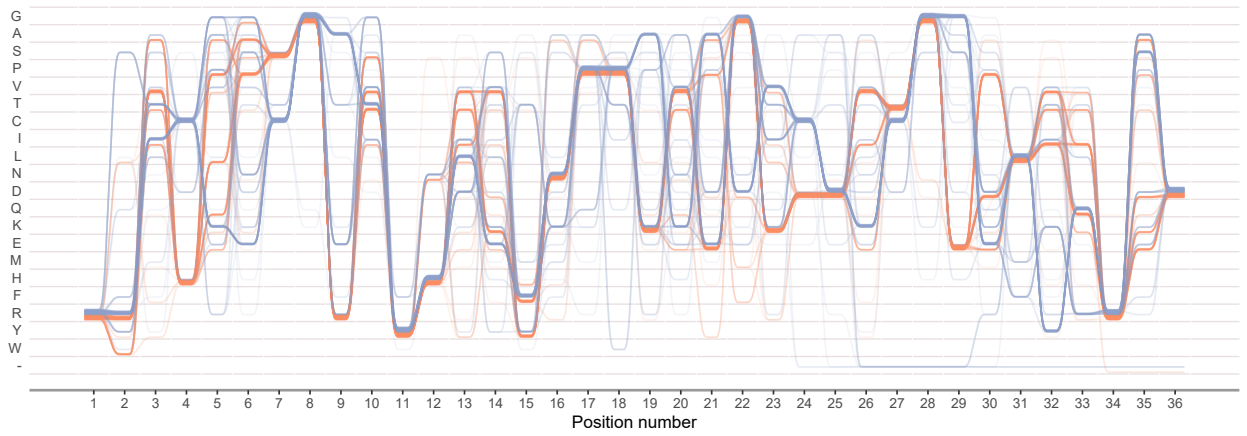


Fig. S8: The preference of residues' physicochemical property can be reflected by adjusting the order of letters on the y-axis. It shows re-dsignaed residues order according to amino acid molecule weight. The large molecule is at the bottom

Table S1: Comparison of ggmsa with popular free MSA visualization tools

Table S1: Comparison of ggmsa with popular free MSA visualization tools

Tools	**Platform**	**Sequence Logos**	**Sequence bundles**	**Stacked MSA[1]**	**Layouts for stacked MSA**	**Integrating external data into stacked MSA[2]**	**Exploring sequence recombination[3]**	**User interface**
ggmsa	R package	YES	YES	YES	rectangular fragmentary circular	YES	YES	programming and command line
msa	R package	YES	NO	YES	rectangular	NO	NO	programming and command line
MSAviewer	Web service	YES	NO	YES	rectangular	NO	NO	interactive
AliView	Desktop application	NO	NO	YES	rectangular	NO	NO	interactive
Jalview	Desktop application and web service	YES	NO	YES	rectangular	NO	NO	interactive
ALVIS	Desktop application	YES	YES	YES	rectangular	NO	NO	interactive

[1] Stacked MSA: it represents all sequences as rows and homologous residue positions as columns
[2] Extended MSA: adding associated into stacked MSA plots
[3] A visualization method that designed for detecting sequence recombination signals

Fig. 2A: The combination of sequence logos and sequence bundles (R code)

```
p2a <- seqlogo("data/Gram-NP-merge.fa",
               color = "Chemistry_AA",
               font = "DroidSansMono") +
  coord_cartesian()
```

```
negative <-  system.file("extdata", "Gram-negative_AKL.fasta",
                         package = "ggmsa")
positive <-  system.file("extdata", "Gram-positive_AKL.fasta",
                         package = "ggmsa")

pos <- data.frame(x= c(4, 7, 9, 24, 27, 29,
                       4, 7, 24, 27),
                  y = c(c(21, 11, 20, 17, 12, 18) +.3,
                        c(13, 13, 13, 13) +.5
                        ),
                  label = c("H", "S", "R", "D", "T", "E",
                            "C", "C", "C", "C"),
                  color = c(rep("#ff4700",6),
                            rep("#0443d0",4)))

p2b <- ggSeqBundle(list(negative, positive),
                   alpha = 0.1,
                   bundle_color = c("#FC8D62","#8DA0CB"))+ #RColorBrewer: Set2:2-3
  geom_text(data = pos,
            mapping = aes(x, y,
                          label = label,
                          color = I(color)),
            inherit.aes = FALSE,
            size = 4)

plot_list(gglist = list(p2a, p2b), ncol = 1, heights = c(0.3,1))
```

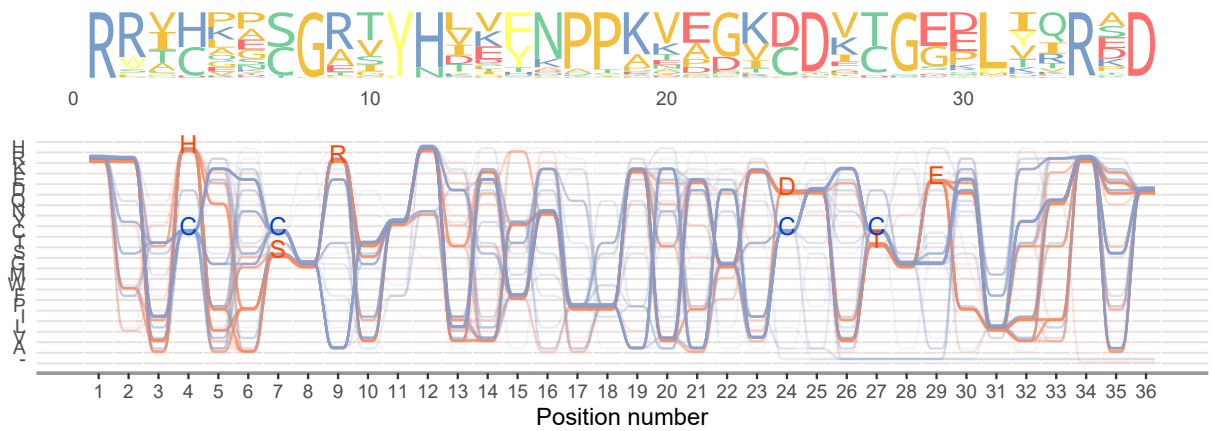


Fig. 2A| The combination of sequence logos and sequence bundles. The data contain adenylate kinase lid (AKL) domain both of Gram-negative and Gram-negative bacteria. 100 sequences for each groups. The sequence logo (top panel) represents the AKL sequence pattern and the sequence bundle (bottom panel) represents the different residues relationship between Gram-negative (orange) and Gram-negative (purple) bacteria. The site at 4, 7, 9, 24, 27 and 29 has exclusive pattern (His4, Ser7, Arg9, Asp24, Thr27, Glu29) in the Gram-negative sequences. And the site at 4, 7, 24 and 27 both contain the Cysteines in the Gram-positive sequences. These residues relationship in agreement with the structural stability with the AKL domain. Gram-negatives form hydrogen bonding network by the exclusive pattern and Gram-positives bound metal ion to form coordinated tetrahedral by Cys. (Date from BioVis2013 and repeated example from Science Practice)

Fig. 2B: An example of MSA visualization and MSA annotations (R code)

```
ggmsa(protein_sequences, 221, 280, seq_name = TRUE, char_width = 0.5) +
   geom_seqlogo(color = "Chemistry_AA") +
       geom_msaBar()
```

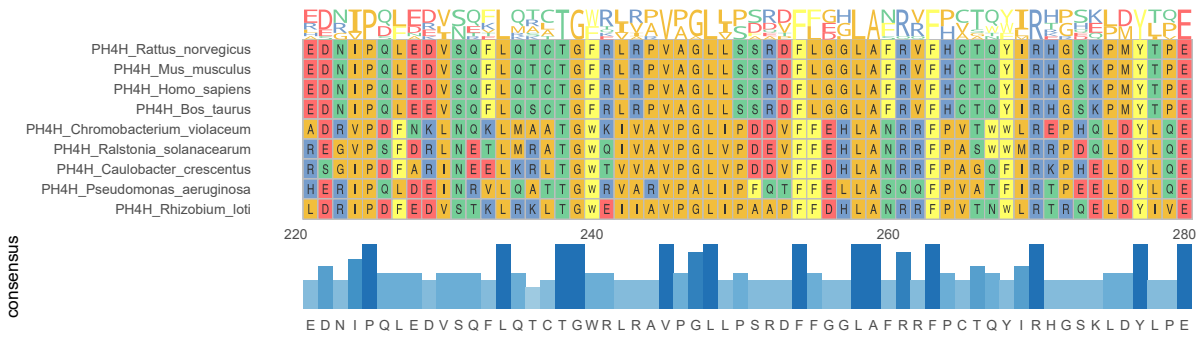


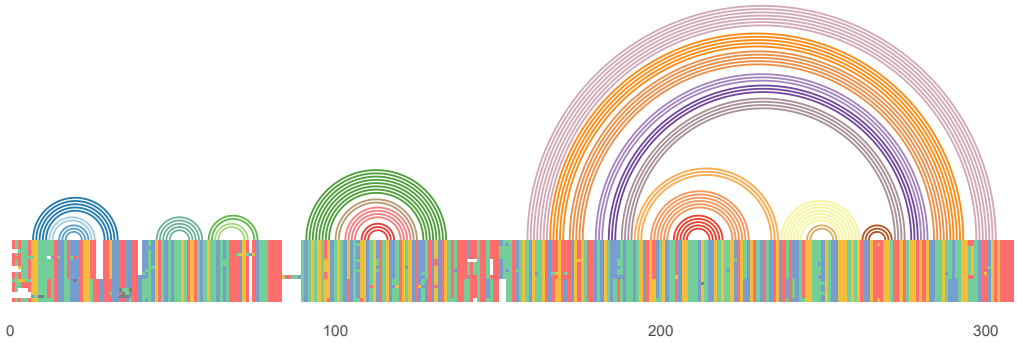
Fig. 2B|A local visualization of the sequence alignment of the phenylalanine hydroxylase protein (PH4H) within nine species. The center panel is the main MSA plot with residues in the alignment colored according to the Chemistry color scheme (amino acids are colored according to their sidechain chemistry). The top and bottom panels are corresponding annotations with MSAs, showing the conservation patterns at each position by sequence logos and the distribution of the high-frequency residue by a bar chart, respectively.

#

Fig. 2CD: Visual methods of surveying RNA co-variation and structural changes (R code)

```
RNA7S  <- "data/3JAJ-2D-dotbracket.txt"
RNAP54 <- "data/4UJE-2D-dotbracket.txt"

RF03120_msa<- system.file("extdata", "Rfam", "RF03120.fasta", package = "ggmsa")
RF03120_ss <- system.file("extdata", "Rfam", "RF03120_SS.txt", package = "ggmsa")

known <- readSSfile(RNA7S, type = "Vienna" )
transat <- readSSfile(RNAP54 , type = "Vienna")


RF_arc <- readSSfile(RF03120_ss, type = "Vienna" )

p2C <- ggmsa(RF03120_msa,
             font = NULL,
             color = "Chemistry_NT",
             seq_name = F,
             show.legend = F,
             border = NA) +
       geom_helix(helix_data = RF_arc) +
       theme(axis.text.y = element_blank())

p2D <- ggmsa("data/5SRNA.fa",
             font = NULL,
             color = "Chemistry_NT",
             seq_name = T,
             show.legend = T,
             border = NA) +
  geom_helix(helix_data = list(known = known,
                               predicted = transat),
             overlap = F)

p2CD <- plot_list(gglist = list(p2C, p2D),
                  ncol = 1,heights = c(0.15),
                  tag_levels = list(c("2C","2D" )))

p2CD
```

2C



2D

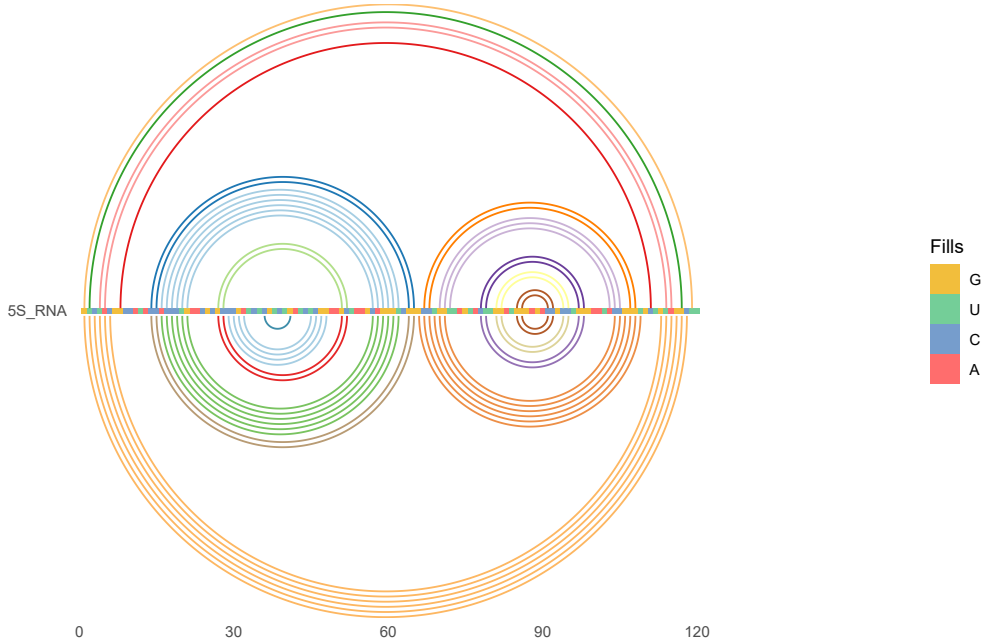


Fig. 2CD| (C) The data from the Rfam database [family RF03120] include 19 seed alignments of Sarbecovirus 5'UTR (including 6 SARS-CoV-2 isolates sequences) and the corresponding consensus RNA secondary structure. Compensatory mutations in MSA can be detected by checking alignment columns in positions corresponding to arcs. **(D)** The sequence and secondary structure of 5S Ribosomal RNA from the PDB database. The arc above the sequence represents the structure of the engaged state of the mammalian SRP-ribosome complex (3JAJ) and the bottom arc depicts the 5S RNA structural changes in a eukaryotic-specific ribosome rearrangement (4UJE).

Fig. 3: Visual exploration for sequence recombination signal (R code)

```
fas <- c("data/HM_KP.fa","data/CK_KP.fa")
xx <- lapply(fas, seqdiff)
plts <- lapply(xx, plot, width = 100)
plts[[3]] <- simplot("data/CK_HM_KP.fa", 'KP827649', smooth = FALSE) + theme(legend.position = "bottom")

plot_list(gglist=plts, ncol=1, tag_levels = list(c("A",' ',"B", ' ',"C")))
```

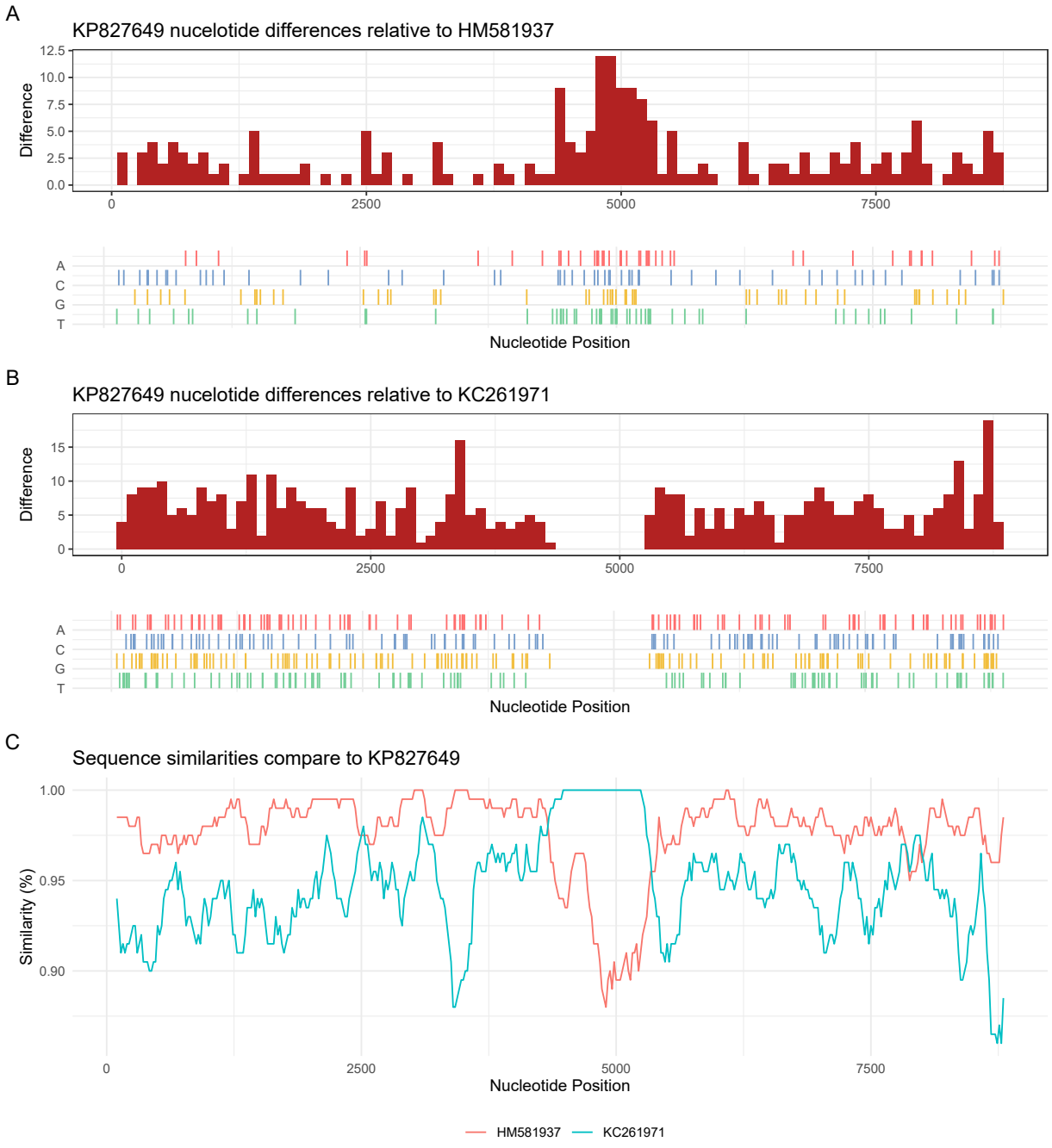


Fig. 3| The nucleotide different plots were generated using the comparison of KP827649 sequence (Tomato

Spotted Wilt Virus large RNA genomic segment) with that of (A) HM581979 sequence (Major parent) and (B) KC261971 sequence (Minor parent). (C) The similarity curves investigates the sequence similarity between KP827649 sequence and potential parents. Two recombination signals (The intersection of two curves in similarity plot, start breakpoint is 4534 and the end breakpoint is 5536) were detected in the TSWV LRNA genomic segment.

Fig. 4A: Examples of graphics combination (R code)

```
##Fig 6A tree + msa + genes locus
dat <- read.aa(tp53_sequences, format = "fasta") %>% phyDat(type = "AA", levels = NULL)
tree <- dist.ml(dat, model = "JTT") %>% bionj()
dd <- ggimage::phylopic_uid(tree$tip.label)

p_tp53 <- ggtree(tree, branch.length = 'none') %<+% dd +
  geom_tiplab(aes(image=uid), geom = "phylopic", offset =1.9) +
  geom_tiplab(aes(label=label)) +
  geom_treescale(x = 0,y = -1)
#msa
data_53 <- readAAMultipleAlignment(tp53_sequences) %>% tidy_msa()
#gene maps
TP53_arrow <- readxl::read_xlsx(tp53_genes)
TP53_arrow$direction <- 1
TP53_arrow[TP53_arrow$strand == "reverse","direction"] <- -1

#color
mapping = aes(xmin = start, xmax = end, fill = gene, forward = direction)
my_pal <- colorRampPalette(rev(brewer.pal(n = 10, name = "Set3")))

#tree + gene maps + msa
p4a <- p_tp53  + xlim_tree(4) +
  geom_facet(geom = geom_msa, data = data_53,
             panel = 'Multiple Sequence Alignment of the TP53 Protein', font = NULL,
             border = NA) +
  new_scale_fill() +
  scale_fill_manual(values = my_pal(10)) +
  geom_facet(geom = geom_motif,
             mapping = mapping, data = TP53_arrow,
             panel = 'Genome_Locus',  on = 'TP53',
             arrowhead_height = unit(3, "mm"),
             arrowhead_width = unit(1, "mm")) +
  theme(strip.background=element_blank(),
        strip.text = element_text(size = 13))
p4A <- facet_widths(p4a, c(Tree = 0.35, Genome_Locus = 0.3))
p4A
```

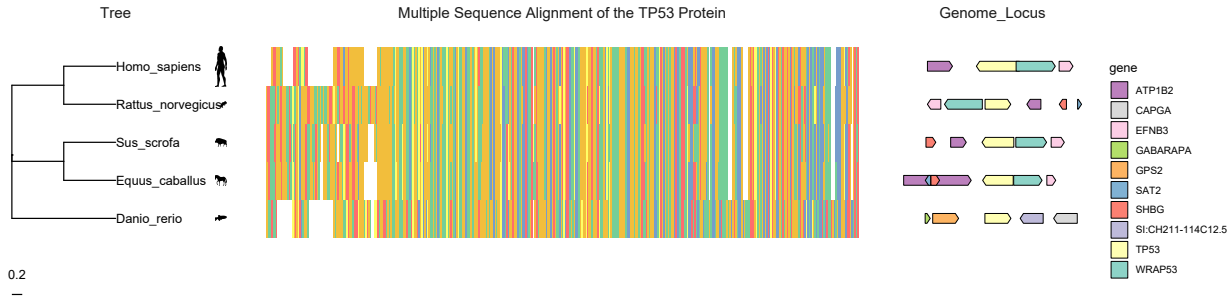


Fig. 4A| The MSA-tree panel in conjunction with external genome locus data set. Comparative genome locus structure (genome_Locus panel), sequence alignment of TP53 protein (the middle panel), and the corresponding phylogenetic tree (Tree panel) among six species. The local genome map shows the 30000 sites around the TP53 gene, and the phylogenetic tree that represents evolutionary relationships inferred from TP53 protein sequences using the Neighbor-Joining method based on the evolutionary distances of JTT matrix-based method.

Fig. 4B: Examples of graphics combination (R code)

```
##Fig 6B tree + msa + 2boxplot
seq <-readDNAStringSet("data//btuR.fa")
aln <- tidy_msa(seq)
btuR_tree <- read.tree("data/btuR.nwk")
meta_dat <- read.csv("data/meta_data_47.csv")

#Pathotype_fill_colors
Pathotype_cols <- RColorBrewer::brewer.pal(7, "Set3")
names(Pathotype_cols) <- meta_dat$Pathotypes %>% factor %>% levels

###tree OTU
Phylo_group <- list(A= meta_dat$Lineage[meta_dat$Phylogroup == "A"]%>% unique,
                    B1=meta_dat$Lineage[meta_dat$Phylogroup == "B1"]%>% unique,
                    B2=meta_dat$Lineage[meta_dat$Phylogroup == "B2"]%>% unique,
                    C=meta_dat$Lineage[meta_dat$Phylogroup == "C"]%>% unique,
                    D =meta_dat$Lineage[meta_dat$Phylogroup == "D"]%>% unique,
                    E =meta_dat$Lineage[meta_dat$Phylogroup == "E"]%>% unique,
                    `F`=meta_dat$Lineage[meta_dat$Phylogroup == "F"]%>% unique,
                    Shigella=meta_dat$Lineage[meta_dat$Phylogroup == "Shigella"]%>% unique)

Phylo_cols <- RColorBrewer::brewer.pal(8, "Dark2")
names(Phylo_cols) <- names(Phylo_group)

## plot tree
p_btuR_tree <- ggtree(btuR_tree) + geom_tiplab(align = T)
p_btuR_tree <- groupOTU(p_btuR_tree ,Phylo_group)+aes(color=group)  +
  scale_color_manual(values = c(Phylo_cols, "black"), na.value = "black", name = "Lineage",
                     breaks = c("A", "B1", "B2", "C", "D", "E", "F", "Shigella"),  guide="none")

p_btuR_tree <- p_btuR_tree +
  geom_strip('L29', 'L20', barsize=2, color=Phylo_cols[["B2"]],
             label="B2",offset =.01,  offset.text = 0.0015) +
  geom_strip('L28','L29', barsize=2, color=Phylo_cols[["A"]],
```

```
            label="A",offset =.01,  offset.text = 0.0015) +
  geom_strip('L15','L28', barsize=2, color=Phylo_cols[["B1"]],
            label="B1",offset =.01,  offset.text = 0.0015) +
  geom_strip('L45','L15', barsize=2, color=Phylo_cols[["Shigella"]],
            label="S.",offset =.01,  offset.text = 0.0015) +
  geom_strip('L36','L45', barsize=2, color=Phylo_cols[["B1"]],
            label="B1",offset =.01,  offset.text = 0.0015) +
  geom_strip('L30','L36', barsize=2, color=Phylo_cols[["Shigella"]],
            label="S.",offset =.01,  offset.text = 0.0015) +
  geom_strip('L39','L30', barsize=2, color=Phylo_cols[["B1"]],
            label="B1",offset =.01,  offset.text = 0.0015) +
  geom_strip('L40','L39', barsize=2, color=Phylo_cols[["C"]],
            label="C",offset =.01,  offset.text = 0.0015)  +
  geom_strip('L48','L40', barsize=2, color=Phylo_cols[["B1"]],
            label="B1",offset =.01,  offset.text = 0.0015) +
  geom_strip('L10','L48', barsize=2, color=Phylo_cols[["E"]],
            label="E",offset =.01,  offset.text = 0.0015) +
  geom_strip('L37','L10', barsize=2, color=Phylo_cols[["D"]],
            label="D",offset =.01,  offset.text = 0.0015)+
  geom_strip('L33','L37', barsize=2, color=Phylo_cols[["F"]],
            label="F",offset =.01,  offset.text = 0.0015)+
  geom_strip('L1','L33', barsize=2, color=Phylo_cols[["E"]],
            label="E",offset =.01,  offset.text = 0.0015)

##tree + meta data boxplots
p4B <- p_btuR_tree  +
  geom_treescale(x = 0,y = -1) +
  geom_fruit(data = aln,
            geom = geom_msa,
            end = 200,
            font = NULL,
            color = "Chemistry_NT",
            border = NA,
            consensus_views = T,
            ref = "L38",
            pwidth = 3.5,
            offset = 0.3,
            axis.params = list(title = "Multiple Sequence Alignment of the btuR Gene",
                               title.height = 0.05,
                               title.size = 4.5,
                               axis = "x",
                               vjust = 1.1,
                               text.size = 3,
                               line.size = 1,
                               line.color = "black")) +
  new_scale_fill() +
  geom_fruit(mapping = aes(x = AMR_genes, y = Lineage, fill = MDR),
            data = meta_dat,
            geom = geom_boxplot,
            outlier.size = 0.5,
            pwidth=1,
            offset = 0.1,
            axis.params = list(title = "Antimicrobial Classes",
```

```
                                title.height = 0.05,
                                title.size = 4.5,
                                axis = "x",
                                vjust = 1.1,
                                text.size = 3,
                                line.size = 1,
                                line.color = "black")) +
  scale_fill_manual(values=c("Yes" = "#fcd7c5", "No" = "#dfdfdf")) +
  new_scale_fill() +
  geom_fruit(mapping = aes(x = virulence_genes, y = Lineage, fill = Pathotypes),
             data = meta_dat,
             geom = geom_boxplot,
             pwidth=1,
             offset = 0.05,
             axis.params = list(title = "Virulence Genes",
                                title.height = 0.05,
                                title.size = 4.5,
                                axis = "x",
                                vjust = 1.1,
                                text.size = 3,
                                line.size = 1,
                                line.color = "black"))+
  scale_fill_manual(values = Pathotype_cols)

p4B
```

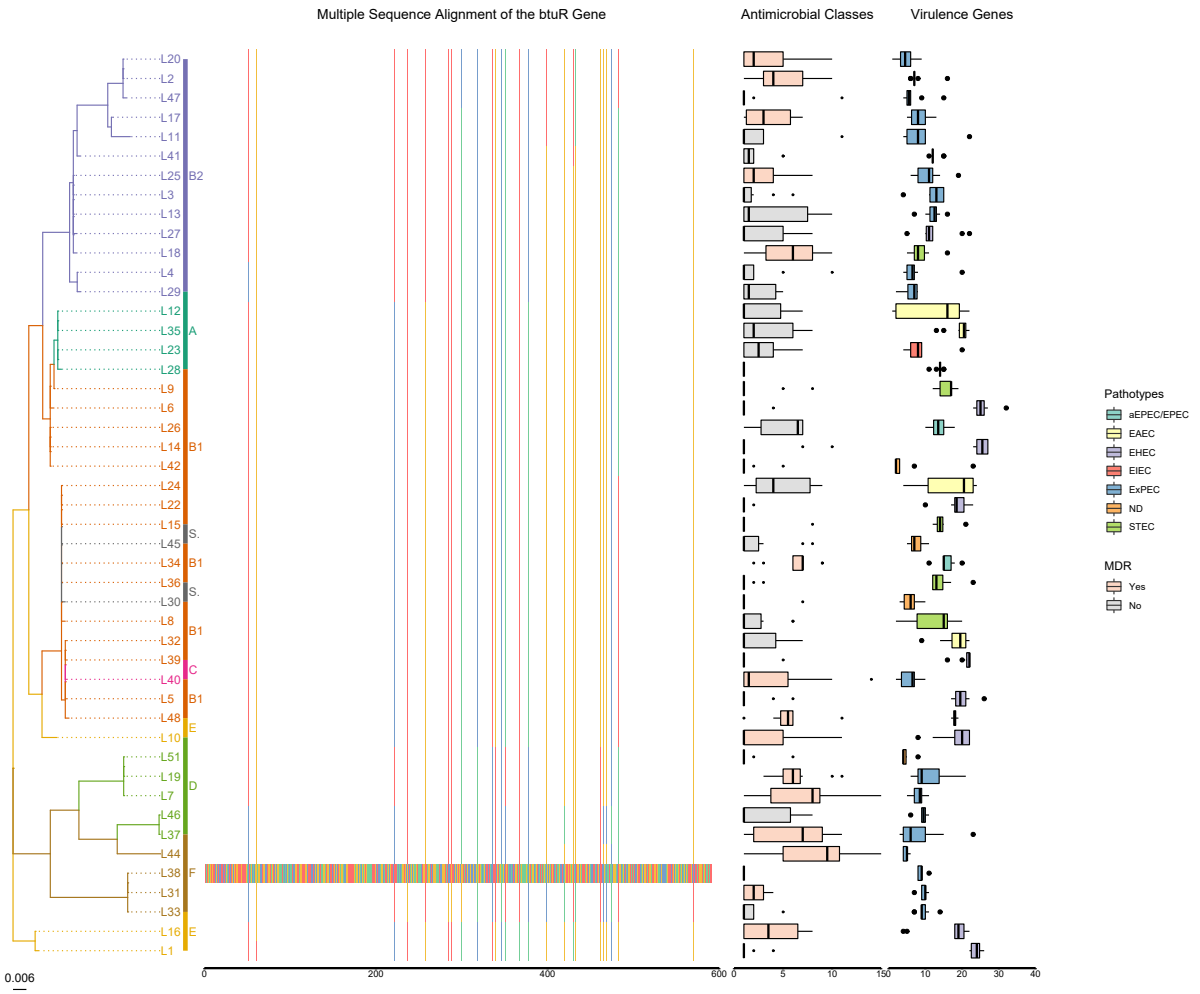


Fig. 4B| The summarized visualization for different E.coli lineages. The alignment of btuR gene was selected to generate a phylogenetic tree which was stained according to the corresponding lineages. The boxplots represent antimicrobial resistance (AMR) and virulence profiles of the lineages. Number of antimicrobial classes and virulence genes per isolate colored by multidrug-resistant (MDR) classes and the most prevalent predicted pathotype in the lineage.

Session Info

Here is the output of sessionInfo() on the system on which this document was compiled:

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Chinese (Simplified)_China.936
## [2] LC_CTYPE=Chinese (Simplified)_China.936
## [3] LC_MONETARY=Chinese (Simplified)_China.936
## [4] LC_NUMERIC=C
```

```
## [5] LC_TIME=Chinese (Simplified)_China.936
##
## attached base packages:
## [1] stats4    stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
##  [1] kableExtra_1.3.4    knitr_1.36          aplot_0.1.1
##  [4] ggplotify_0.1.0     patchwork_1.1.1     RColorBrewer_1.1-2
##  [7] phangorn_2.7.1      ggtreeExtra_1.4.0   dplyr_1.0.7
## [10] ggnewscale_0.4.5    Biostrings_2.62.0   GenomeInfoDb_1.30.0
## [13] XVector_0.34.0      IRanges_2.28.0      S4Vectors_0.32.2
## [16] BiocGenerics_0.40.0 ape_5.5             gggenes_0.4.1
## [19] ggtree_3.2.0        ggplot2_3.3.5       ggmsa_1.1.4
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-153           bitops_1.0-7           webshot_0.5.2
##  [4] ash_1.0-15             httr_1.4.2             tools_4.1.1
##  [7] utf8_1.2.2             R6_2.5.1               KernSmooth_2.23-20
## [10] lazyeval_0.2.2         colorspace_2.0-2       withr_2.4.2
## [13] tidyselect_1.1.1       ggalt_0.4.0            curl_4.3.2
## [16] compiler_4.1.1         extrafontdb_1.0        rvest_1.0.2
## [19] xml2_1.3.2             labeling_0.4.2         scales_1.1.1
## [22] proj4_1.0-10.1         quadprog_1.5-8         systemfonts_1.0.3
## [25] stringr_1.4.0          digest_0.6.28          yulab.utils_0.0.4
## [28] R4RNA_1.22.0           rmarkdown_2.11         svglite_2.0.0
## [31] pkgconfig_2.0.3        htmltools_0.5.2        extrafont_0.17
## [34] fastmap_1.1.0          highr_0.9              maps_3.4.0
## [37] readxl_1.3.1           rlang_0.4.11           rstudioapi_0.13
## [40] gridGraphics_0.5-1     farver_2.1.0           generics_0.1.1
## [43] jsonlite_1.7.2         RCurl_1.98-1.5         magrittr_2.0.1
## [46] GenomeInfoDbData_1.2.7 Matrix_1.3-4           Rcpp_1.0.7
## [49] munsell_0.5.0          fansi_0.5.0            ggfittext_0.9.1
## [52] lifecycle_1.0.1        stringi_1.7.5          yaml_2.2.1
## [55] seqmagick_0.1.5        MASS_7.3-54            zlibbioc_1.40.0
## [58] grid_4.1.1             parallel_4.1.1         crayon_1.4.2
## [61] lattice_0.20-45        splines_4.1.1          magick_2.7.3
## [64] pillar_1.6.4           igraph_1.2.7           codetools_0.2-18
## [67] fastmatch_1.1-3        glue_1.4.2             evaluate_0.14
## [70] ggimage_0.3.0          ggfun_0.0.4            vctrs_0.3.8
## [73] treeio_1.18.0          tweenr_1.0.2           cellranger_1.1.0
## [76] Rttf2pt1_1.3.9         gtable_0.3.0           purrr_0.3.4
## [79] polyclip_1.10-0        tidyr_1.1.4            xfun_0.26
## [82] ggforce_0.3.3          tidytree_0.3.5         viridisLite_0.4.0
## [85] tibble_3.1.5           ellipsis_0.3.2
```

References

Burland, T G. 2000. "DNASTAR's Lasergene Sequence Analysis Software." *Methods Mol Biol* 132: 71–91.

Kyte, Jack, and Russell F Doolittle. 1982. "A Simple Method for Displaying the Hydropathic Character of a Protein." *Journal of Molecular Biology* 157 (1): 105–32.

Taylor, and R. W. 1997. "Residual Colours: A Proposal for Aminochromography." *Protein Eng* 10 (7): 743–46.

Waterhouse, Andrew M, James B Procter, David MA Martin, Michèle Clamp, and Geoffrey J Barton. 2009. "Jalview Version 2—a Multiple Sequence Alignment Editor and Analysis Workbench." *Bioinformatics* 25 (9): 1189–91.

Yu, Guangchuang, Zehan Dai, Pingfan Guo, Xiaocong Fu, Shanshan Liu, Lang Zhou, Wenli Tang, et al. 2021. "ggtreeExtra: Compact Visualization of Richly Annotated Phylogenetic Data."

Yu, Guangchuang, David K 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.