Supplemental File of

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

```
library(ggmsa)
library(ggplot2)
library(ggtree)
library(gggenes)
library(ape)
library(Biostrings)
library(ggnewscale)
library(dplyr)
library(ggtreeExtra)
library(phangorn)
library(RColorBrewer)
library(patchwork)
library(ggplotify)
library(aplot)
library(magick)
library(treeio)


protein_sequences <- system.file("extdata", "sample.fasta", package = "ggmsa")
nt_sequence <- system.file("extdata", "LeaderRepeat_All.fa", package = "ggmsa")
miRNA_sequences <- system.file("extdata", "seedSample.fa", package = "ggmsa")
tp53_sequences <-  system.file("extdata", "tp53.fa", package = "ggmsa")
tp53_genes <- system.file("extdata", "TP53_genes.xlsx", package = "ggmsa")
```

Fig. S1: Amino acid and nucleotide color schemes in ggmsa

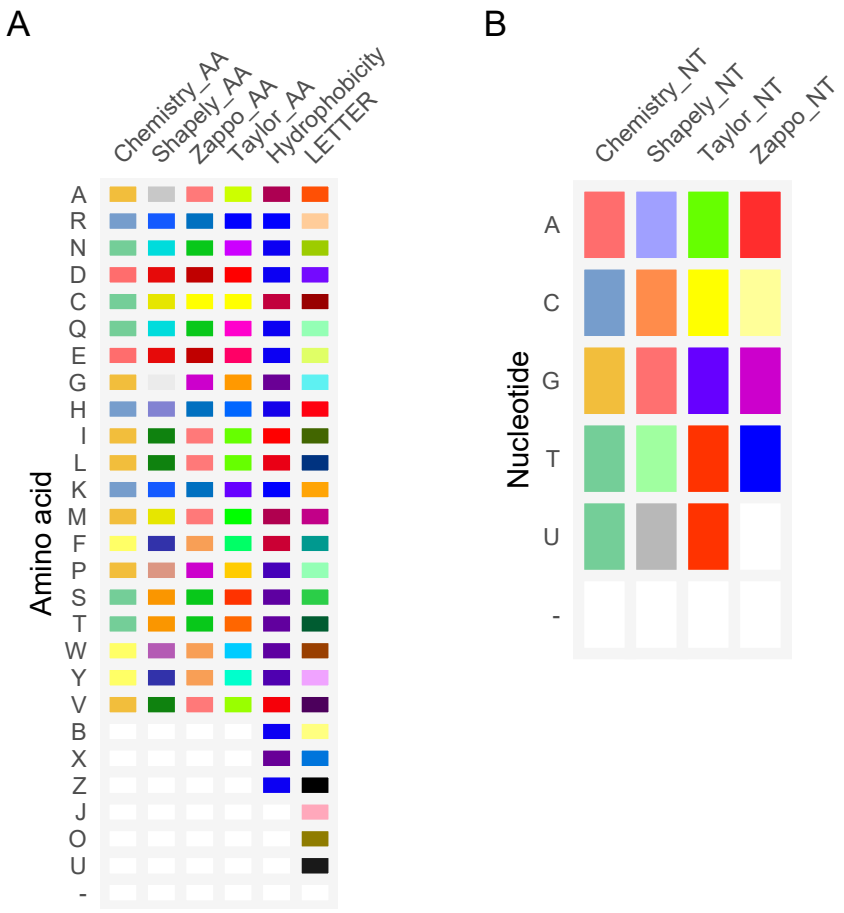

Fig. S1: **Color schemes in ggmsa.** (A) Amino acid color schemes. Schemes are either quantitative, reflecting empirical or statistical properties of amino acids; or qualitative, reflecting physicochemical attributes. Chemistry is colored according to side-chain chemistry and is also used in DNASTAR 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) Nucleotide color schemes used by ggmsa.

Fig. S2: MSA view with break down layout

`facet_msa()` module allows for showing more alignment data in a restricted canvas. The long sequence is 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. S2: **MSA view with break down layout**. The long sequence is displayed in several lines.

Fig. S3: MSA view with circular layout tree

With ggtreeExtra (Xu et al. 2021), displaying alignment in a circular layout can also be supported. The `geom_fruit()` will automatically align the MSA graphs to the tree with a 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)
```

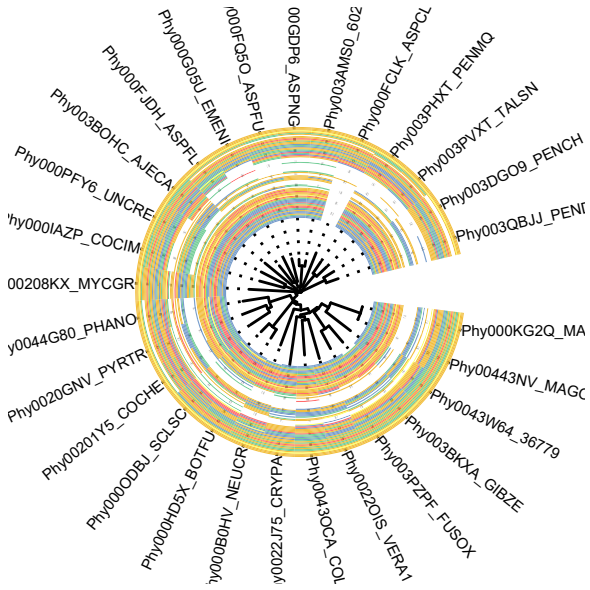

Fig. S3: **Example of MSA view with circular layout phylogenetic tree.**

Fig. S4: Visualizing MSA with a phylogenetic tree

Similar to `geom_fruit()`, the `geom_facet()` provided in ggtree (Yu et al. 2017) also works with ggmsa to display MSA with a phylogenetic tree.

```
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)
```

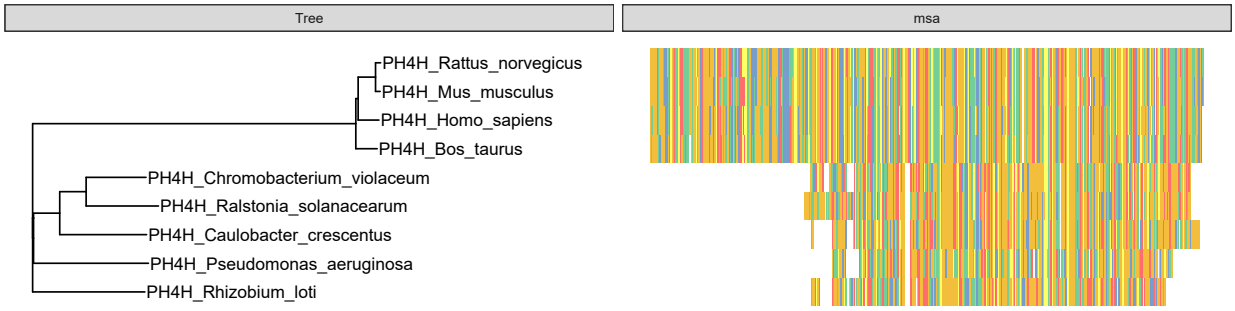

Fig. S4: **A phylogenetic tree with MSA that it was built from.**

Fig. S5: 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"))
```

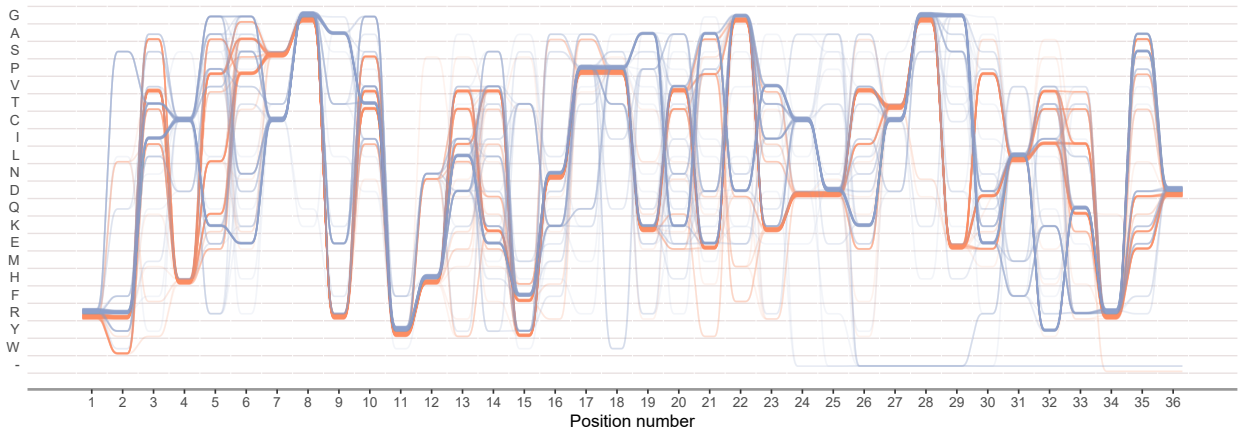

Fig. S5: The preference of residues' physicochemical property can be reflected by adjusting the order of letters on the y-axis. It shows re-dsignaed residues order according to amino acid molecule weight. The large molecule is at the bottom

Fig. S6: The consensus sequence

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

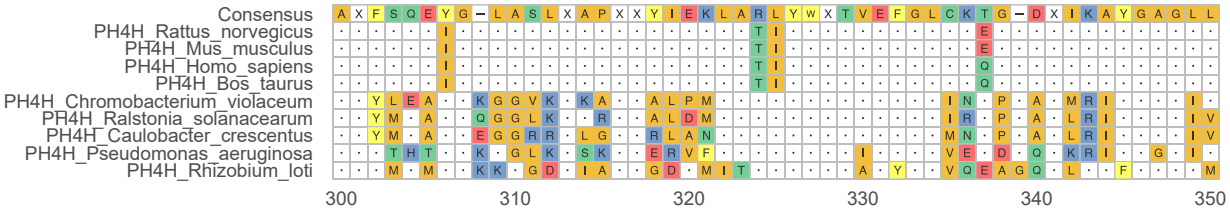

Fig. S6: Example of consensus sequence for an MSA. The consensus sequence is displayed above the alignment and shows which residues are conserved, which residues are variable. A consensus is constructed from the most frequent residues at each site.

Fig. S7: 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()
```

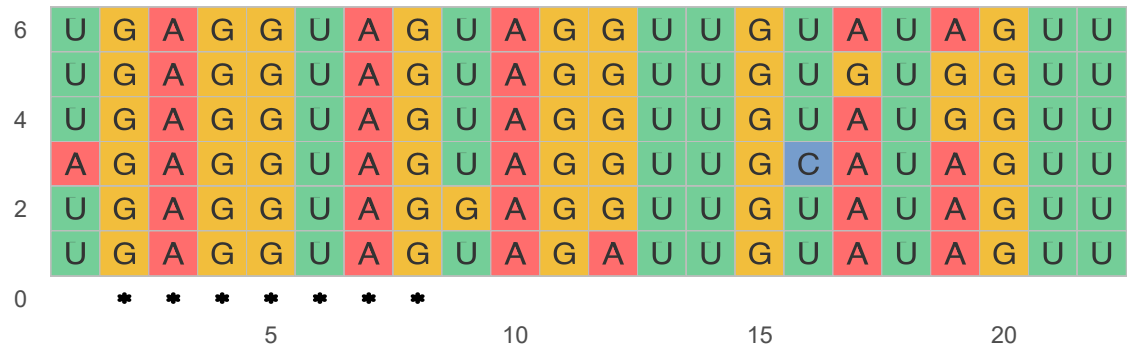

Fig. S7: Example of highlighting miRNA seed region. Asterisks are used for marking the seed region.

Fig. S8: 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()
```

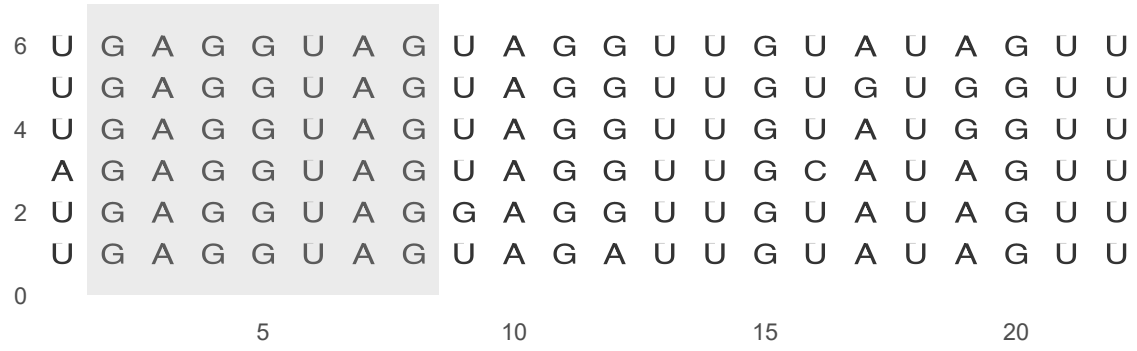

Fig. S8: Example of highlighting miRNA seed region. Annotation of sequence seed region with the shaded block

Fig. S9: Displaying ancestral sequences for in Tree-MSA

```
x <- read.phyloxml("data/msa_phyloxml.xml")
p <- ggtree(x, size = 2) + xlim_tree(0.12)
tidymsa <- extract_seq(p)

p1 <- treeMSA_plot(p,
                   tidymsa,
                   ancestral_node = 11,
                   sub = FALSE,
                   color = "Chemistry_NT")

p2 <- treeMSA_plot(p,
                   tidymsa,
                   ancestral_node = 11,
                   sub = TRUE,
                   color = "Chemistry_NT")

pp <- plot_list(gglist = list(p1,p2),
                nrow = 2,
                tag_levels = "A",
                heights = c(0.4,0.6))
pp
```

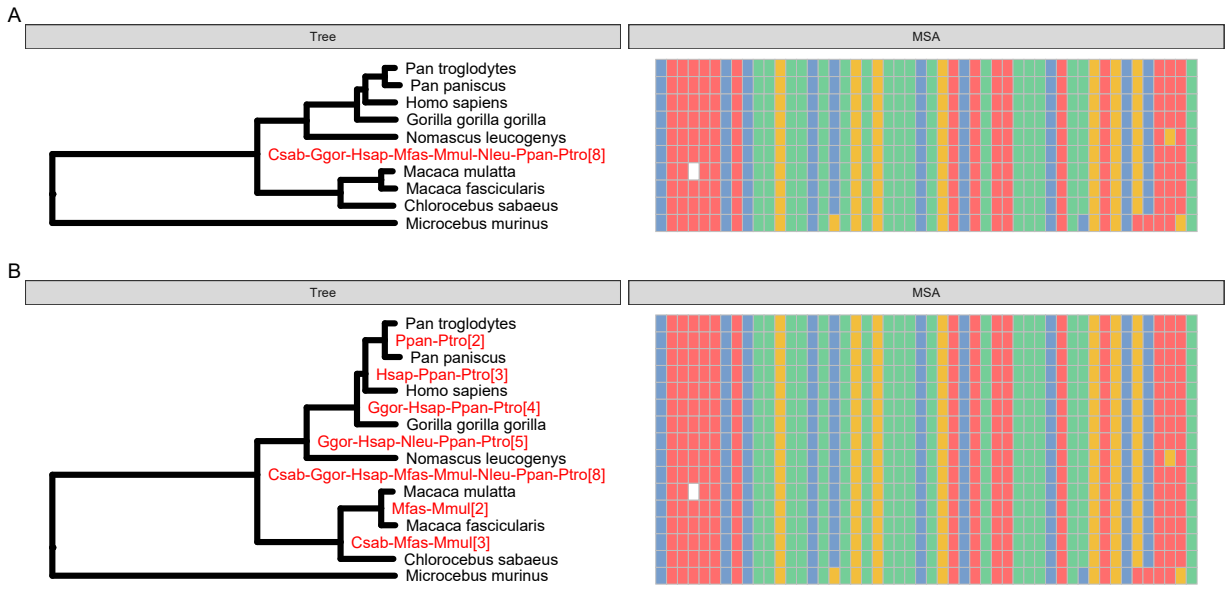

Fig. S9: The ancestral sequence is displayed by selecting the internal nodes in tree-MSA plot. And there are two display modes :(A) only showing the sequence corresponding to the selected ancestral node; (B) Displaying all ancestral sequences of subtree.

Fig. S10: MSA of SARS-CoV-2 Spike protein RBD coloring by RBD-hACE2 binding affinity

We applied DMS data of ACE2-binding affinity to MSA to functionally annotate the mutated sites in the sequence. In addition, the application scope of arc diagram is extended to the interaction process of ACE2-RBD. Through MSA that used ACE2-binding affinity annotation and ACE2-RBD interaction arc diagram, we can observe how the mutation of Omicron affects ACE2-RBD interaction. Although DMS data only reveal the binding affinity of a single amino acid mutation (Fig. S10), we think that MSA plot annotated with DMS data can reflect changes in ACE2-RBD interactions to a certain extent. Arc plot representing ACE2-RBD interactions prove this (Fig. S11,S12). In the ACE2-RBD arc diagram, Omicron mutation sites K417N,G446S and G496S show loss of hydrogen bond interaction, and these sites all show decreased binding ability in the binding affinity annotation diagram. The N501Y mutation in the binding affinity diagram shows an increased binding ability, which may form a new binding site. This has not been shown in Arc plot (possible cause: the interaction site information contained in the 7WPB is not completely accurate), but has been reported in other literature (Yin et al. 2022).

```
colRD <- colorRampPalette(rev(brewer.pal(n = 9, name = "OrRd")))
colBU <- colorRampPalette(colors = rev(c("#185089","#FFF7EC")))

data <- "data/s_RBD.fasta"
dms <- read.csv("data/DMS.csv")
del <- c("expr_lib1", "expr_lib2",
         "expr_avg","bind_lib1",
         "bind_lib2")
dms <- dms[,!colnames(dms) %in% del]

tidymsa <- tidy_msa(data)
tidymsa <- assign_dms(tidymsa, dms)
```

```
#Mapping the position to the protein-protein interaction plot
tidymsa$position <- tidymsa$position + 330

p <- ggplot() +
    geom_msa(data = tidymsa,
             char_width = 0.5,
             dms = TRUE,
             seq_name = TRUE,
             show.legend = TRUE) +
    theme_msa() +
    scale_fill_gradientn(name = "ACE2 binding",
                         colors = c(colRD(75),colBU(25))) +
    facet_msa(50)
p
```

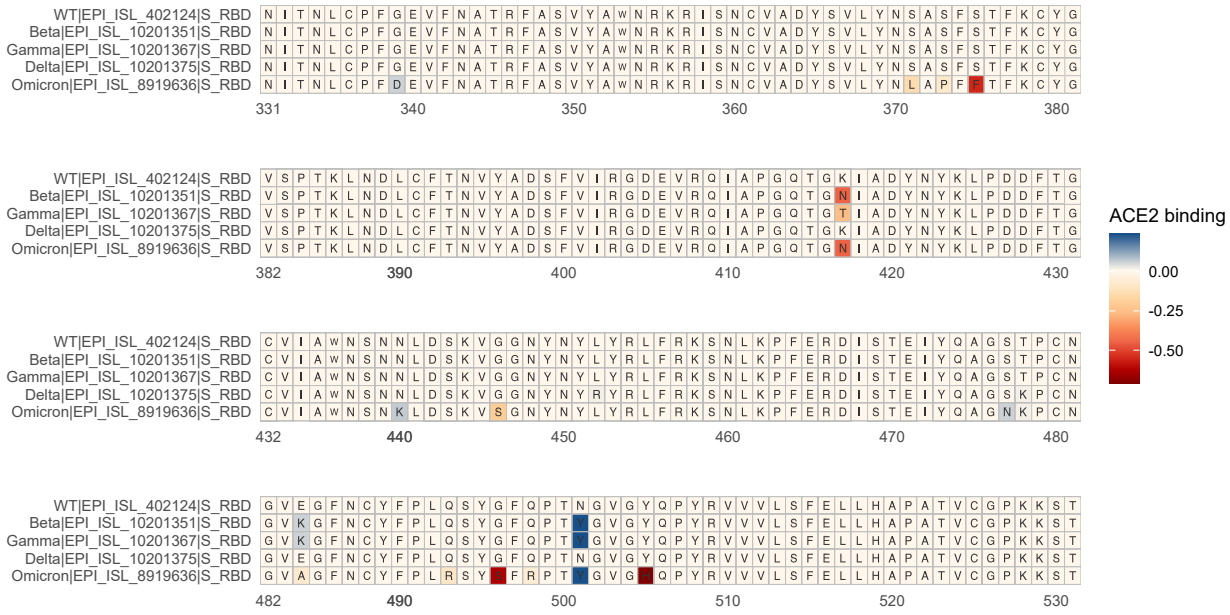

Fig. S10: Mutations in Spike protein Receptor Binding Domain (RBD) are compared with wild-type SARS-CoV-2 and four other strains (Beta, Gamma, Delta ,and Omicron, all from GISAID). The colors in MSA showing RBD-hACE2 (human Angiotensin converting enzyme 2) binding affinity scores. ACE2-binding affinity are shown in shades of blue or red representing higher or lower ACE2 affinity, respectively. The ACE2 binding scores are from Deep Mutational Scanning (DMS) of SARS-CoV-2 RBD. Scores represent binding constants (Δlog10 KD) relative to the wild-type reference amino acid.

Fig. S11: Arc diagram of ACE2-RBD interaction

```
#read sequences
x <- readAAStringSet("data/ACE2.fasta")
y <- readAAStringSet("data/Spike_RBD.fasta")

#read protein-protein position
```

```
inter1 <- read.csv("data/6m0j_inter.csv")
inter2 <- read.csv("data/7wpb_inter.csv")

#tidy data
t1 <- tidy_msa(x, start = 19)
t2 <- tidy_msa(y, start = 331)
t_merge <- merge_seq(t1, gap = 100, t2)

h1 <- tidy_hdata(100, inter1, t1, t2)
h2 <- tidy_hdata(100, inter2, t1, t2)

#protein-protein interactive plot
p_interactive <- ggplot() +
    geom_msa(data = t_merge,border = NA, font = NULL, seq_name = FALSE) +
    theme_msa() + theme(axis.text = element_blank()) +
    geom_helix(helix_data = list(known = h1, predicted = h2))
p_interactive
```

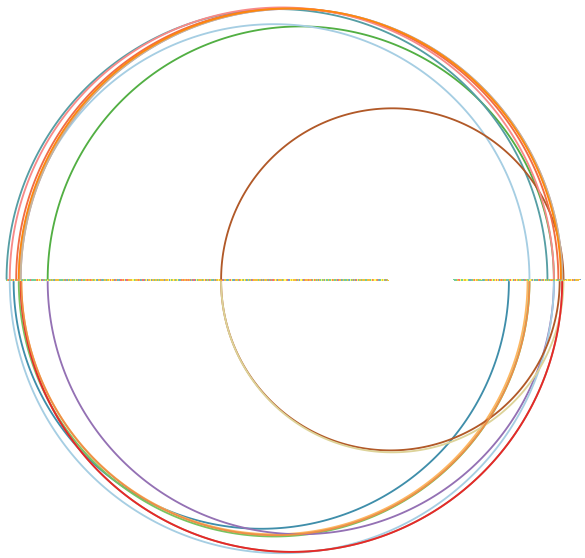

Fig. S11: Arc diagram of ACE2-RBD interaction. The two sequences on the left panel are ACE2, and the two sequences on the right panel are RBD sequences of S protein of wild-type and Omicron, respectively. ACE2 and RBD are placed in the same horizontal position, and the residual sites that generated hydrogen bonds were connected by arcs. The upper arc represents the ACE2-Omicron_RBD and the lower arc represents the ACE2-WT_RBD. Data of interacting sites was obtained from the PDBsum database, and the corresponding PDB ID were 7WPB and 6M0J.

Fig. S12: simplified arc diagram of ACE2-RBD interaction

```
#simplified p-p interactive  plot
ACE2 <- t_merge[t_merge$position %in% c(h1$i,h2$i),]
```

```
spike <- t_merge[t_merge$position %in% c(h1$j,h2$j),]
t1_reset <- reset_pos(ACE2)
t2_reset <- reset_pos(spike)
simplify_merge <- merge_seq(previous_seq = t1_reset,
                            subsequent_seq = t2_reset,
                            gap = 5,
                            adjust_name = FALSE)

sim_h1 <- simplify_hdata(hdata = h1, sim_msa = simplify_merge)
sim_h2 <- simplify_hdata(hdata = h2, sim_msa = simplify_merge)

##break and label
b <- simplify_merge[simplify_merge$character != "-", "position"] %>% unique()
l <-  c(inter1$Res.no.1,inter1$Res.no..2,
        inter2$Res.no.1,inter2$Res.no..2) %>% unique

p_sim <- ggplot() +
    geom_msa(data = simplify_merge,border = NA, char_width = 0.5, seq_name = F) +
    ggmsa:::theme_msa() +
    geom_helix(helix_data = list(known = sim_h1,
                                 predicted = sim_h2),
               overlap = F) +
    geom_text(mapping = aes(x = b,
                            y = 0.25,
                            label = l[order(l)]),
              size = 3) +
    theme(axis.text.x = element_blank(),
          axis.text.y = element_blank()) +
    scale_y_discrete(labels = c("6m0j:A(ACE2)", "7wpb:D(ACE2)")) +
    geom_text(aes(x = c(27.2,27.2),
                  y = c(1,2),
                  label = c("6m0j:A(Spike_RBD)",
                            "7wpb:D(Spike_RBD)")),
              size = 3.5) +
    geom_text(aes(x = c(-1.5,-1.5),
                  y = c(1,2),
                  label = c("6m0j:A(ACE2)",
                            "7wpb:D(ACE2)")),
              size = 3.5)  + xlim(-2,29)
p_sim
```

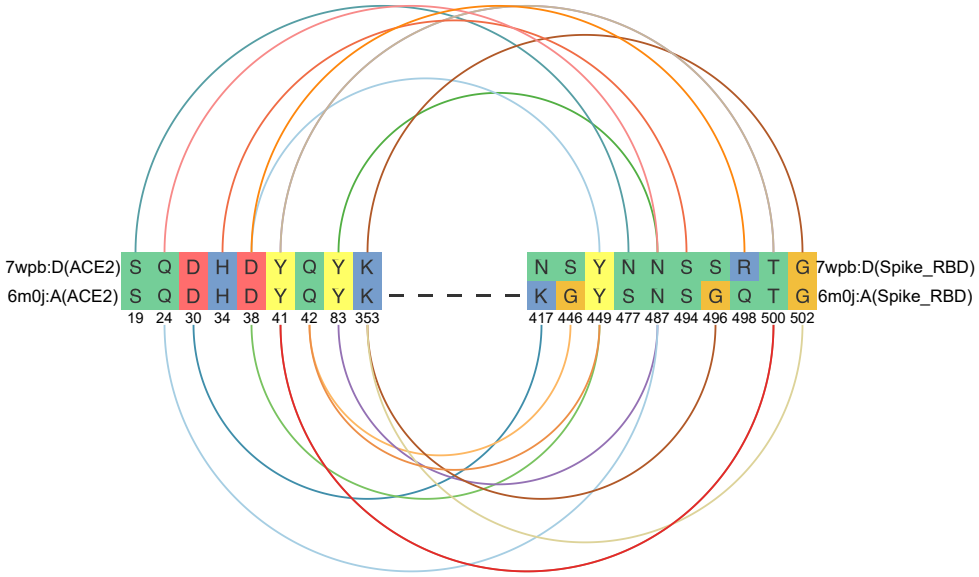

Fig. S12: simplified ACE2-RBD interaction arc plot.The arc plot only shows the residues at the sites of interaction.

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

We compared MSA visualization methods among ggmsa and other tools (**msa** (Bodenhofer et al. 2015), **MSAviewer** (Yachdav et al. 2016), **AliView** (Larsson 2014), **Jalview** (Waterhouse et al. 2009), **ALVIS** (Schwarz et al. 2016))

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))
```

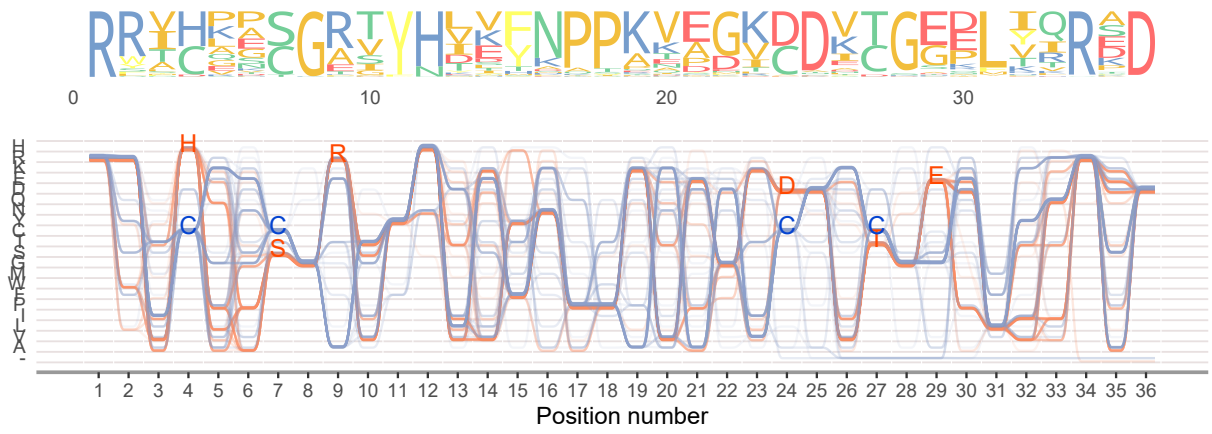

Fig. 2A| The combination of sequence logos and sequence bundles. The data contain adenylate kinase lid (AKL) domain both of Gram-negative and Gram-negative bacteria. 100 sequences for each groups. The sequence logo (top panel) represents the AKL sequence pattern and the sequence bundle (bottom

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()
```

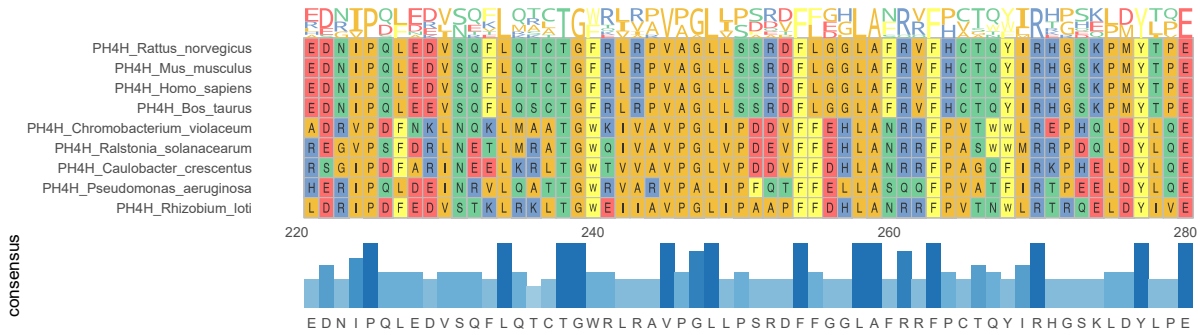

Fig. 2B|A local visualization of the sequence alignment of the phenylalanine hydroxylase protein (PH4H) within nine species. The center panel is the main MSA plot with residues in the alignment colored according to the Chemistry color scheme (amino acids are colored according to their side chain chemistry). 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. 2C: Visual methods of surveying RNA co-variation (R code)

```
RF03120_msa<- system.file("extdata", "Rfam", "RF03120.fasta", package = "ggmsa")
RF03120_ss <- system.file("extdata", "Rfam", "RF03120_SS.txt", package = "ggmsa")
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())
require(patchwork)
p_RF03120_SS <- image_read_svg("data/RF03120_SS.svg")
q_RF03120_SS <- as.ggplot(p_RF03120_SS)
p2C <- p2c + inset_element(q_RF03120_SS,
                           left = 0,
                           bottom = 0.6,
                           right = 0.4,
                           top = 1,
                           align_to = 'full')
p2C
```

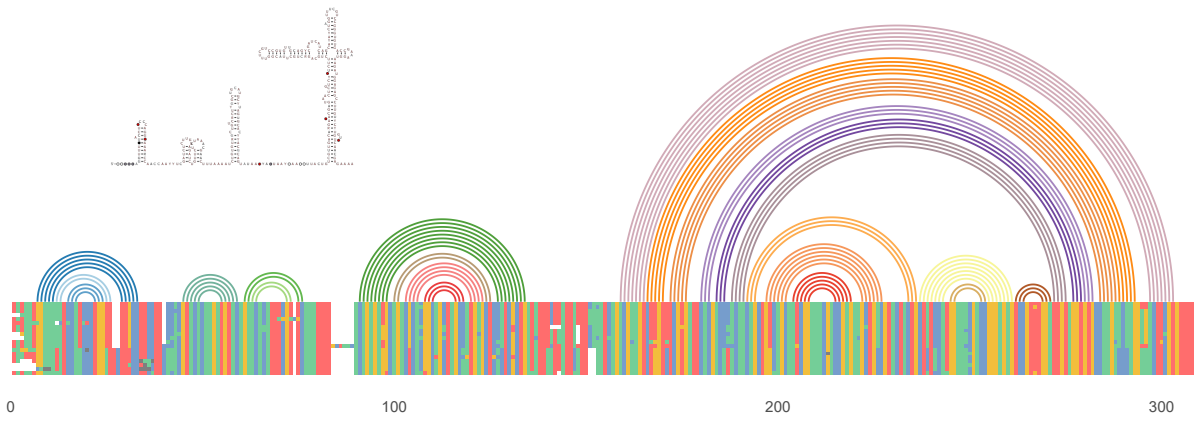

Fig. 2C| The data from the Rfam database [family RF03120] include 19 seed alignments of Sarbecovirus 5'UTR (including 6 SARS-CoV-2 isolates sequences) and the corresponding consensus RNA secondary structure. Compensatory mutations in MSA can be detected by checking alignment columns in positions corresponding to arcs.

Fig. 2D: Visual RNA structural changes (R code)

```
tpp_seq <- "data/tpp_riboswitch.fasta"
arc_4NYG <- "data/riboswitch_thiamine.txt"
```

```
arc_4NYD <- "data/riboswitch_hypoxanthine.txt"
thiamine <- readSSfile(arc_4NYG, type = "Vienna" )
hypoxanthine <- readSSfile(arc_4NYD, type = "Vienna")

p <- ggmsa(tpp_seq,
           color = "Chemistry_NT",
           seq_name = F,
           show.legend = F,
           border = NA) +
    geom_helix(helix_data = list(known = hypoxanthine,
                                 predicted = thiamine)) +
    theme(axis.text.y = element_blank())

p1 <- image_read_pdf(path = "data/bpRNA_PDB_590_ColorCodedStructures_4NYG.pdf",
                     density = 300)
p2 <- image_read_pdf(path = "data/bpRNA_PDB_589_ColorCodedStructures_4NYD.pdf",
                     density = 300)

q1 <- as.ggplot(p1)
q2 <- as.ggplot(p2)

p_loop <- plot_list(gglist = list(q1, q2), ncol = 1)
pp <- plot_list(gglist = list(p_loop, p), ncol = 2)
pp
```

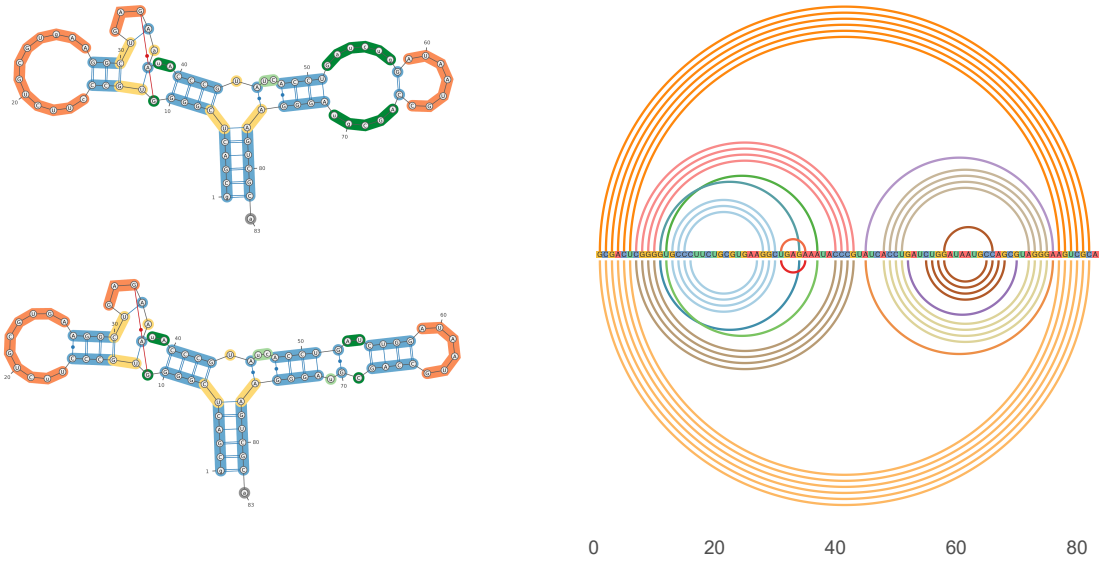

Fig. 2D|Exploring RNA structural changes by geom_helix annotation. Stem-loop diagram and arc diagram of secondary structure changes of TPP-riboswitches after bounding to Hypoxanthine and Thiamine respectively. The sequence of arc plot is the TPP-riboswitch RNA. The arc above the sequence represents the structure of the TPP-riboswitch bound to Thiamine and the bottom arc depicts TPP-riboswitch RNA structural changes binding to Hypoxanthine. The two stem-loop diagrams on the right correspond to the upper and lower arcs respectively. Stem-loop annotation of TPP-riboswitch from bpRNA-1m database (bpRNA ID: bpRNA_PDB_589 and bpRNA_PDB_590), and also corresponding PDB ID: 4NYD, 4NYG.

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")))
```

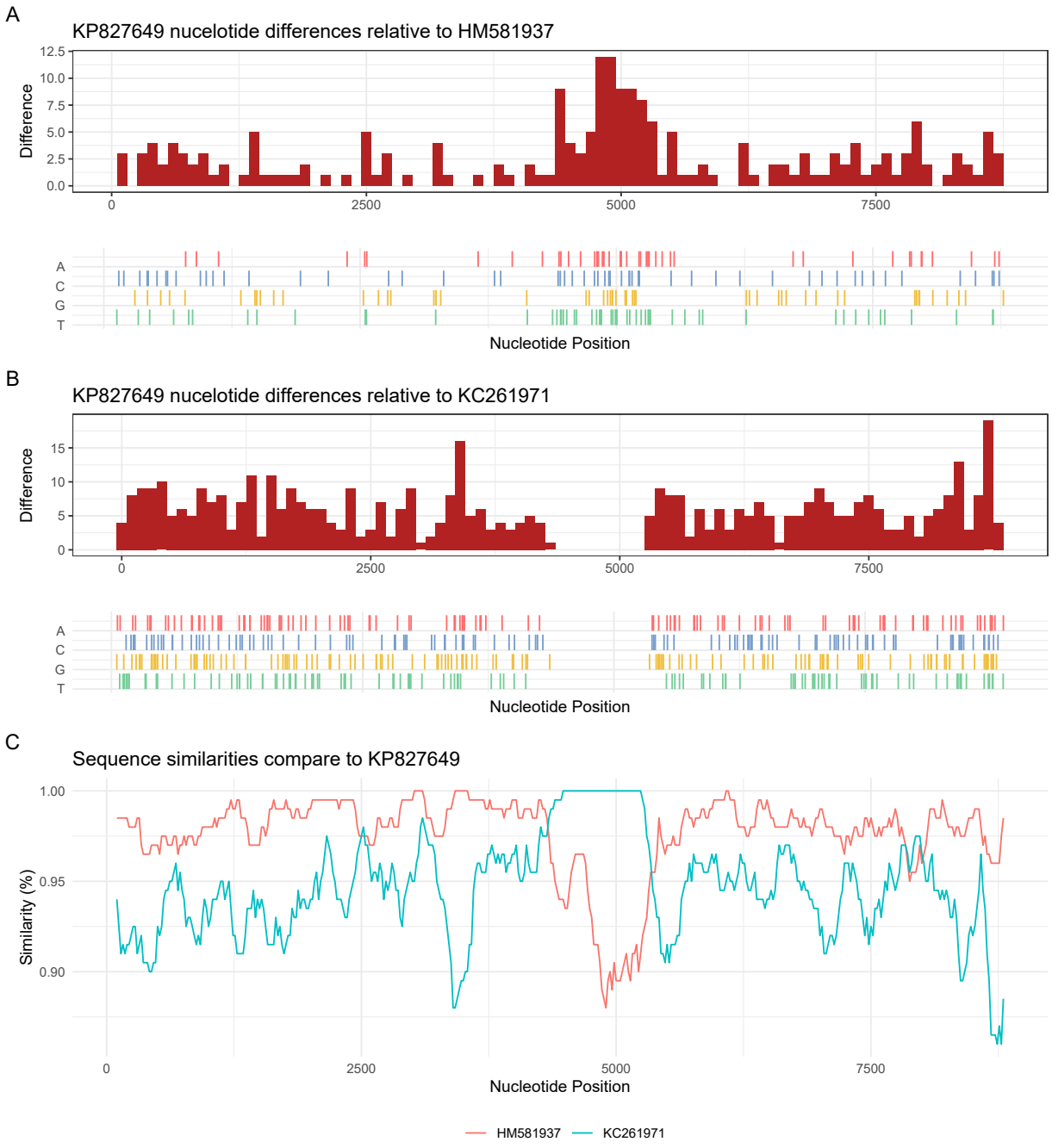

Fig. 3| The nucleotide different plots were generated using the comparison of KP827649 sequence (Tomato Spotted Wilt Virus large RNA genomic segment) with that of (A) HM581979 sequence (Major parent) and (B) KC261971 sequence (Minor parent). (C) The similarity curves investigates the sequence similarity 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
```

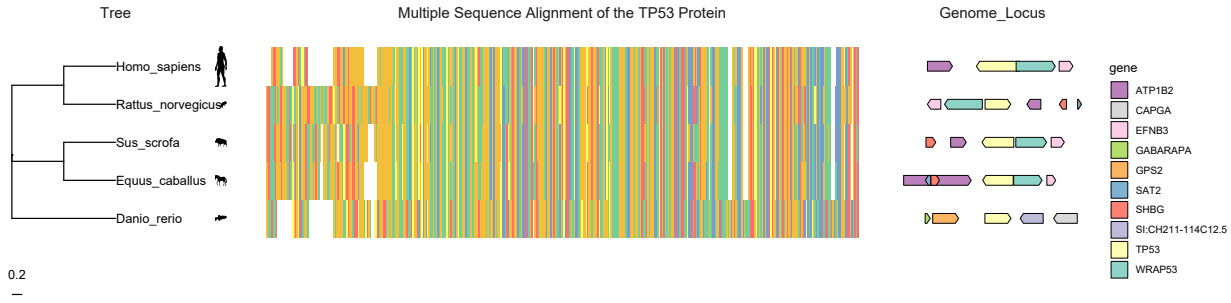

Fig. 4A| The MSA-tree panel in conjunction with external genome locus data set. Comparative genome locus structure (genome_Locus panel), sequence alignment of TP53 protein (the middle panel), and the corresponding phylogenetic tree (Tree panel) among six species. The local genome map shows the 30000 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
```

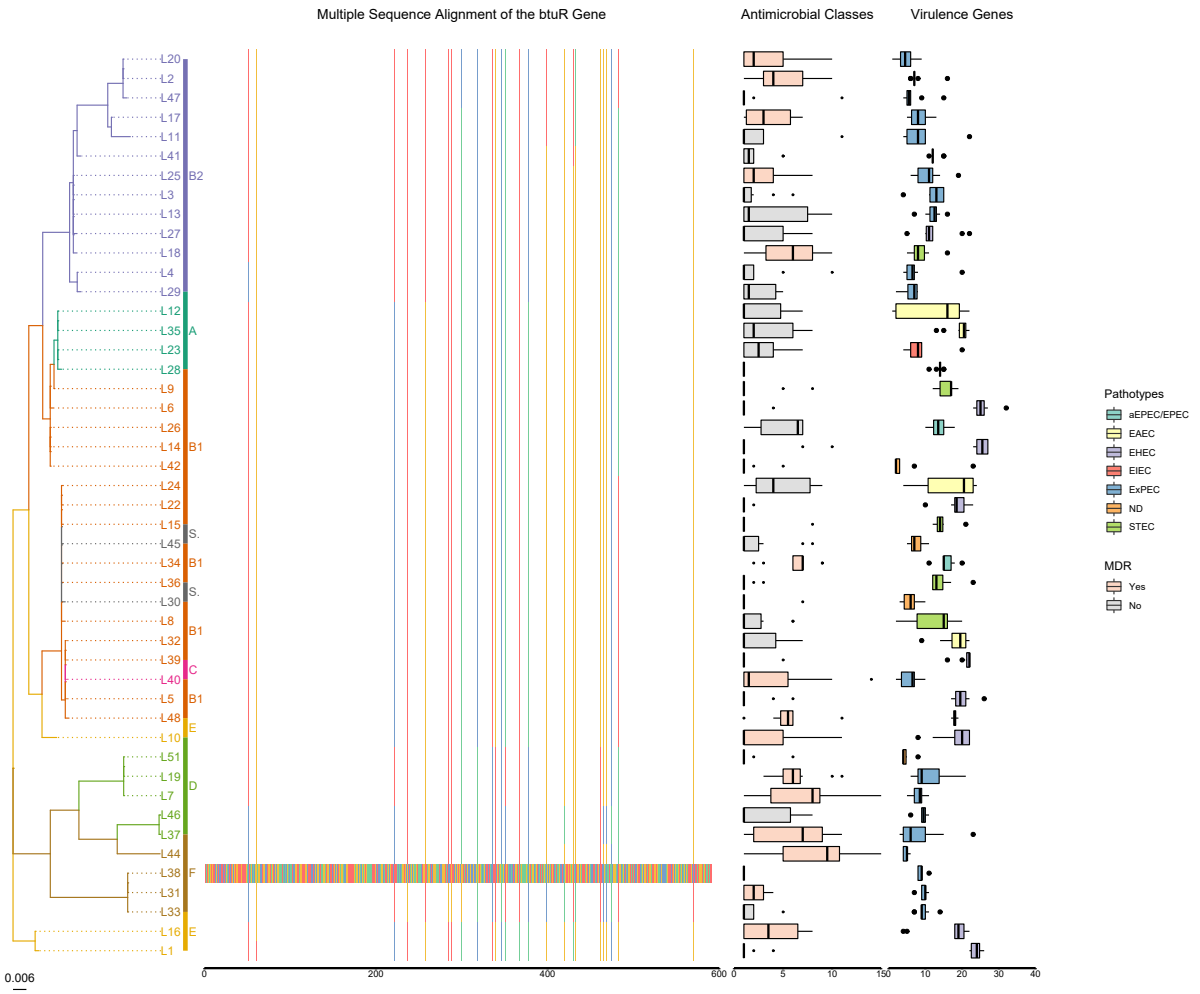

Fig. 4B| The summarized visualization for different E.coli lineages. The alignment of btuR gene was selected to generate a phylogenetic tree which was stained according to the corresponding lineages. The boxplots represent antimicrobial resistance (AMR) and virulence profiles of the lineages. Number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

Fig. 5: Visualization of genome alignment in MAF format (R code)

```
maf <- "data/chr1_KI270707v1_random.txt.maf"
ref = "hg38.chr1_KI270707v1_random"
seq_df <- read_maf(maf)
tidy_df <- tidy_maf_df(seq_df, ref = ref)

ggmaf(data = tidy_df,
      ref = ref,
      block_start = 1,
      block_end = 10,
      facet_field = 5,
      facet_heights = c(0.4,0.6))
```

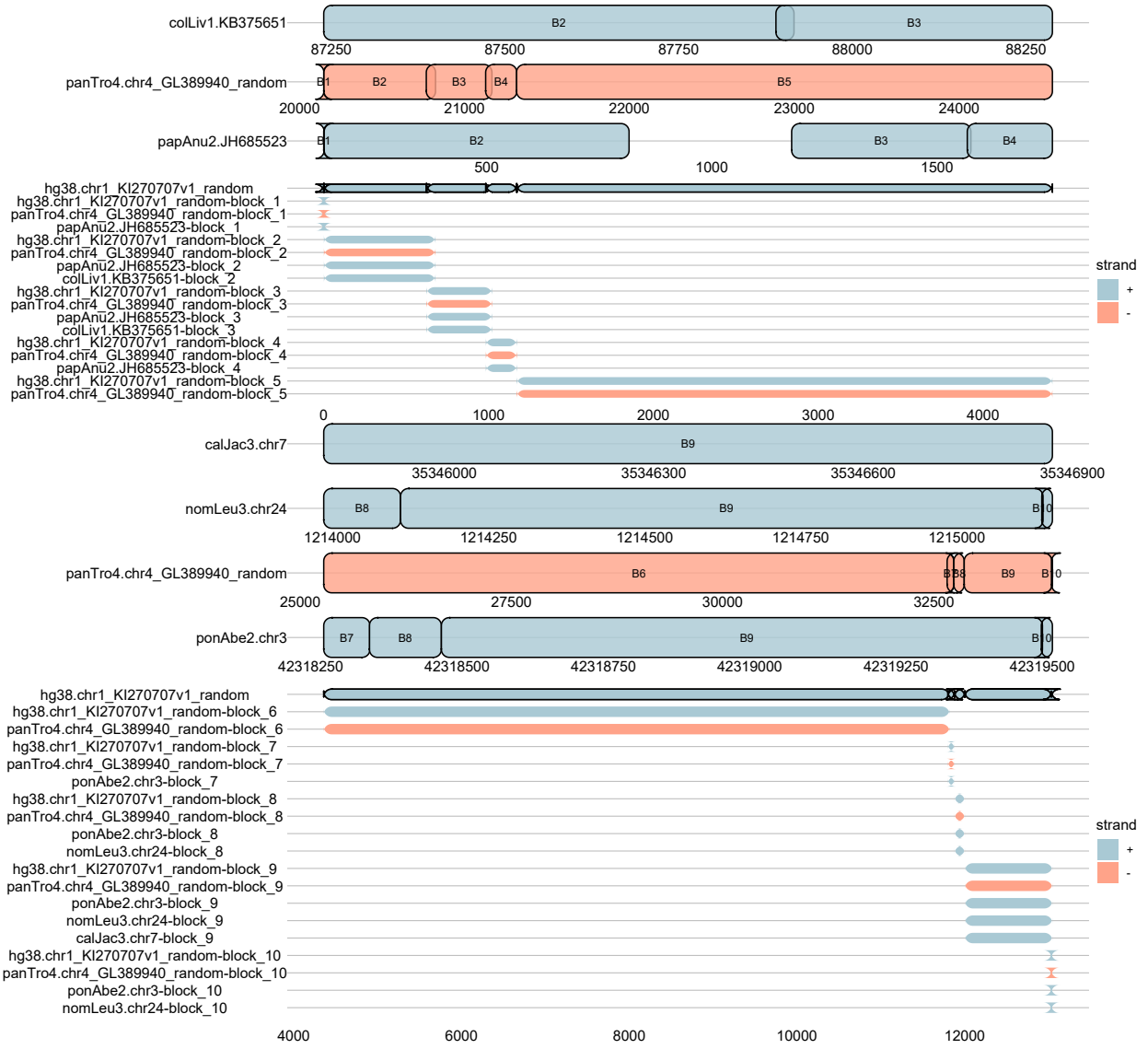

Fig. 5| Visualization of the first 10 blocks of different species genomes aligning to human hg38.chr1. The 10 blocks are divided into two facets: block 1-block 5 is in the first facet, and block 6-block 10 is in the second facet. In each facet, the upper panel shows the original genome coordinates and the lower panel shows genome alignments, with colors indicating positive and negative strands. In the genome alignments panel, the first genome with black border represents the reference genome hg38.chr1 and subsequences in each blocks does not show border color. In each block, the first sequence is the reference genome fragment, followed by other species fragments. Fragments in each block can correspond to labels B1-B10 on the original genome coordinates panel.

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 19044)
##
## 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    treeio_1.18.0       magick_2.7.3
##  [4] aplot_0.1.1         ggplotify_0.1.0     patchwork_1.1.1
##  [7] RColorBrewer_1.1-2  phangorn_2.7.1      ggtreeExtra_1.4.0
## [10] dplyr_1.0.7         ggnewscale_0.4.5    Biostrings_2.62.0
## [13] GenomeInfoDb_1.30.0 XVector_0.34.0      IRanges_2.28.0
## [16] S4Vectors_0.32.2    BiocGenerics_0.40.0 ape_5.5
## [19] gggenes_0.4.1       ggtree_3.3.1.901    ggplot2_3.3.5
## [22] ggmsa_1.1.6         knitr_1.36
##
## 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] statebins_1.4.0        xml2_1.3.2             labeling_0.4.2
## [22] scales_1.1.1           proj4_1.0-10.1         quadprog_1.5-8
## [25] askpass_1.1            systemfonts_1.0.3      stringr_1.4.0
## [28] digest_0.6.28          yulab.utils_0.0.4      R4RNA_1.22.0
## [31] rmarkdown_2.11         svglite_2.0.0          pkgconfig_2.0.3
## [34] htmltools_0.5.2        extrafont_0.17         fastmap_1.1.0
## [37] highr_0.9              maps_3.4.0             readxl_1.3.1
## [40] rlang_0.4.11           rstudioapi_0.13        gridGraphics_0.5-1
## [43] farver_2.1.0           generics_0.1.1         jsonlite_1.7.2
## [46] RCurl_1.98-1.5         magrittr_2.0.1         GenomeInfoDbData_1.2.7
## [49] Matrix_1.3-4           Rcpp_1.0.7             munsell_0.5.0
## [52] fansi_0.5.0            ggfittext_0.9.1        lifecycle_1.0.1
## [55] stringi_1.7.5          yaml_2.2.1             seqmagick_0.1.5
## [58] MASS_7.3-54            zlibbioc_1.40.0        grid_4.1.1
## [61] parallel_4.1.1         crayon_1.4.2           lattice_0.20-45
## [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          pdftools_3.1.1         qpdf_1.1
## [73] ggfun_0.0.6            vctrs_0.3.8            tweenr_1.0.2
## [76] cellranger_1.1.0       Rttf2pt1_1.3.9         gtable_0.3.0
## [79] purrr_0.3.4            polyclip_1.10-0        tidyr_1.1.4
## [82] xfun_0.26              ggforce_0.3.3          rsvg_2.2.0
## [85] tidytree_0.3.9         viridisLite_0.4.0      tibble_3.1.5
## [88] ellipsis_0.3.2
```

References

Bodenhofer, Ulrich, Enrico Bonatesta, Christoph Horejš-Kainrath, and Sepp Hochreiter. 2015. "Msa: An r Package for Multiple Sequence Alignment." *Bioinformatics* 31 (24): 3997–99.

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.

Larsson, Anders. 2014. "AliView: A Fast and Lightweight Alignment Viewer and Editor for Large Datasets." *Bioinformatics* 30 (22): 3276–78.

Schwarz, Roland F, Asif U Tamuri, Marek Kultys, James King, James Godwin, Ana M Florescu, Jörg Schultz, and Nick Goldman. 2016. "ALVIS: Interactive Non-Aggregative Visualization and Explorative Analysis of Multiple Sequence Alignments." *Nucleic Acids Research* 44 (8): e77–77.

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.

Xu, Shuangbin, Zehan Dai, Pingfan Guo, Xiaocong Fu, Shanshan Liu, Lang Zhou, Wenli Tang, et al. 2021. "ggtreeExtra: Compact Visualization of Richly Annotated Phylogenetic Data." *Molecular Biology and Evolution* 38 (9): 4039–42.

Yachdav, Guy, Sebastian Wilzbach, Benedikt Rauscher, Robert Sheridan, Ian Sillitoe, James Procter, Suzanna E Lewis, Burkhard Rost, and Tatyana Goldberg. 2016. "MSAViewer: Interactive JavaScript Visualization of Multiple Sequence Alignments." *Bioinformatics* 32 (22): 3501–3.

Yin, Wanchao, Youwei Xu, Peiyu Xu, Xiaodan Cao, Canrong Wu, Chunyin Gu, Xinheng He, et al. 2022. "Structures of the Omicron Spike Trimer with Ace2 and an Anti-Omicron Antibody." *Science* 375 (6584): 1048–53.

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.