Supplemental File of

ggmsa: an R package for visualizing publication-quality multiple sequence alignment

Fig. S1: Example of a custom color scheme

Customizing the color scheme is allowed. Users can create a data frame with two columns named `names` and `color`. This data frame includes residue letters and coloring code (see below).

```
library(RColorBrewer)
library(pals)
protein_sequences <- system.file("extdata", "sample.fasta", package = "ggmsa")
my_pal <- colorRampPalette(rev(brewer.pal(n = 9, name = "Reds")))
my_cutstom <- data.frame(names = c(LETTERS[1:26],"-"),
                         color = my_pal(27),
                         stringsAsFactors = F)
head(my_cutstom)
```

```
##   names   color
## 1     A #67000D
## 2     B #7A040F
## 3     C #8D0911
## 4     D #A00D14
## 5     E #AD1116
## 6     F #B91319
```

```
ggmsa(protein_sequences, 300, 345,
      custom_color = my_cutstom,
      char_width = 0.5,
      border = "white")
```

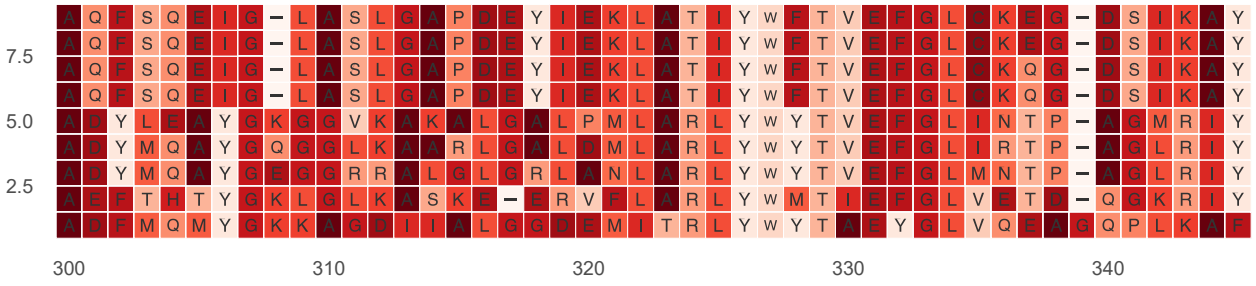

Fig. S1: Assigning color codes for each character, ggmsa enables users to customize color schemes. The same applies to sequence logos.

Fig. S2: Examples of amino acid and nucleotide color schemes in ggmsa

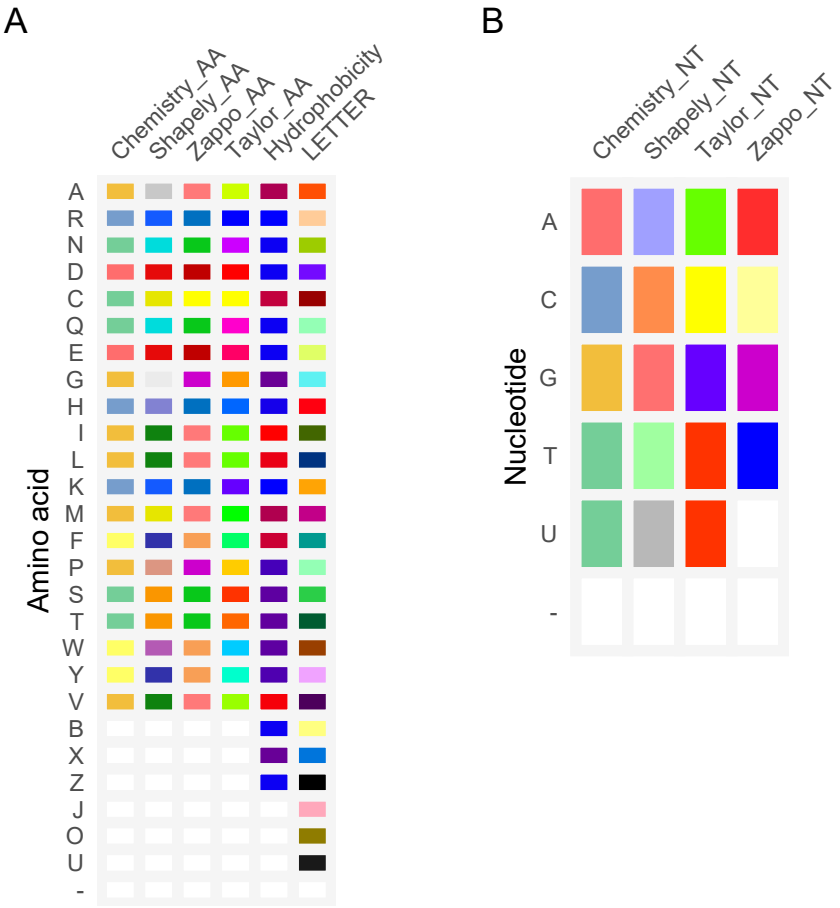

Fig. S2: (A) Examples of amino acid color schemes. Schemes are either quantitative, reflecting empirical or statistical properties of amino acids; or qualitative, reflecting physicochemical attributes. Chemistry is colored according to side-chain chemistry also used in DNASTAR applications (Burland 2000); 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. S3: Bird's-eyes view in ggmsa

```
RF00458_msa<- system.file("extdata", "Rfam", "RF00458.fasta", package = "ggmsa")
ggmsa(RF00458_msa, font = NULL, border = NA, seq_name = F, color = "Chemistry_NT") +
  coord_cartesian()
```

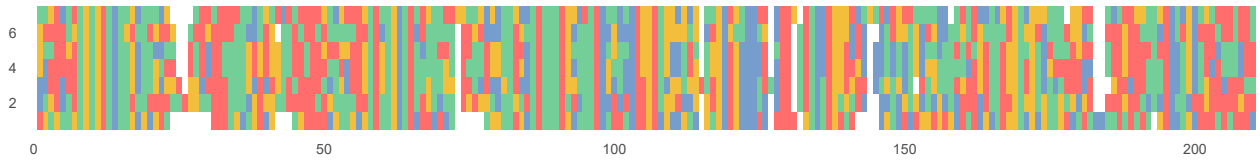

Fig. S3: A compact overview MSA showing a bird's eye view of the full alignment of the Cripavirus Internal Ribosomal Entry Site [family RF00458 from the Rfam database](Nawrocki et al. 2015).

Fig. S4: The shading methods: by_conservation

```
ggmsa(protein_sequences, 300, 350, color = "Hydrophobicity",
      font = NULL, seq_name = T ,border = "white", by_conservation = T)
```

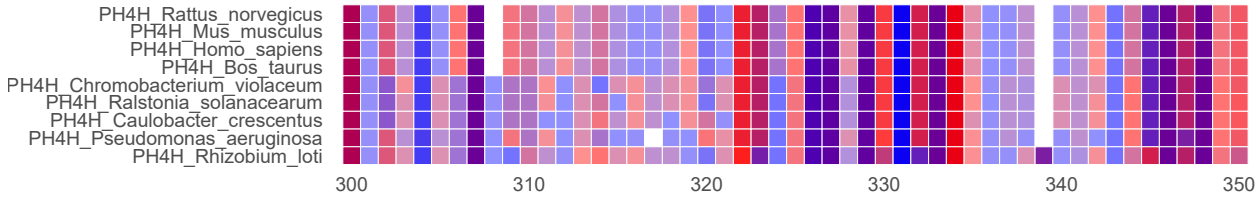

Fig. S4: Example of 'by_conservation' view for MSA. This example shows the results of first coloring by 'hydrophobicity' then performing conservation analysis. Sequence alignment has the most conserved near 330 sites.

Fig. S5: Protein sequence logos annotation for an MSA

The `+` is used to link main function `ggmsa()` and `geom_seqlogo()` function. The `geom_seqlogo()` generates sequence logos according to each column's molecular frequency distribution on MSA.

```
ggmsa(protein_sequences, 300, 350, char_width = 0.5, seq_name = T ) +
    geom_seqlogo(color = "Chemistry_AA")
```

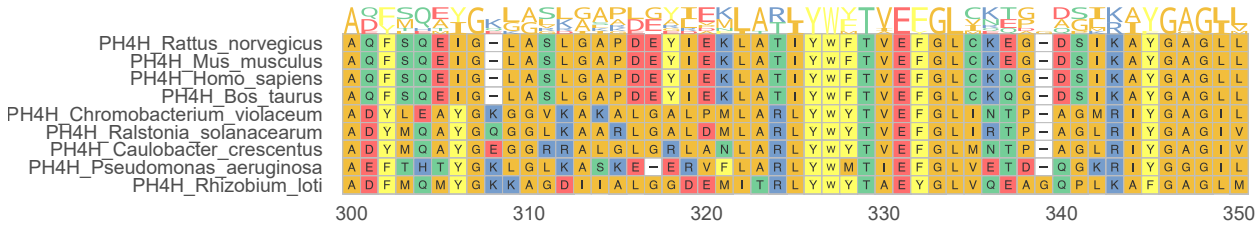

Fig. S5: Sequence Logos between 300 and 350 sites of PH4H which play a major role in indentifying DNA, RNA, and protein binding sites (Schneider and Stephens 1990)

Fig. S6: DNA sequence logos and GC bubbles for an MSA

`geom_GC()` calculating the GC-content for each DNA/RNA sequence. The sizes of bubbles are identical with GC-content.

```
nt_sequence <- system.file("extdata", "LeaderRepeat_All.fa", package = "ggmsa")
ggmsa(nt_sequence,font = NULL, color = "Chemistry_NT") +
    geom_seqlogo(color = "Chemistry_NT") + geom_GC() + theme(legend.position = "none")
```

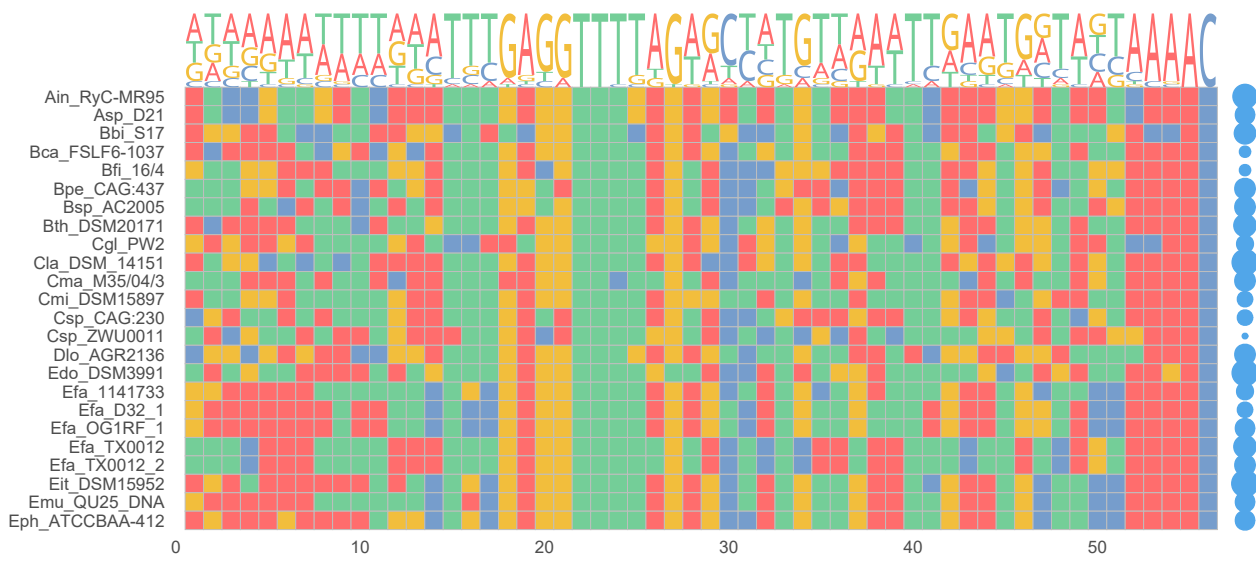

Fig. S6: Example of DNA sequence logos (top) and GC-content (right) automatically generated by ggmsa.The size of the bubbles determined the GC content of each sequence.

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()
```

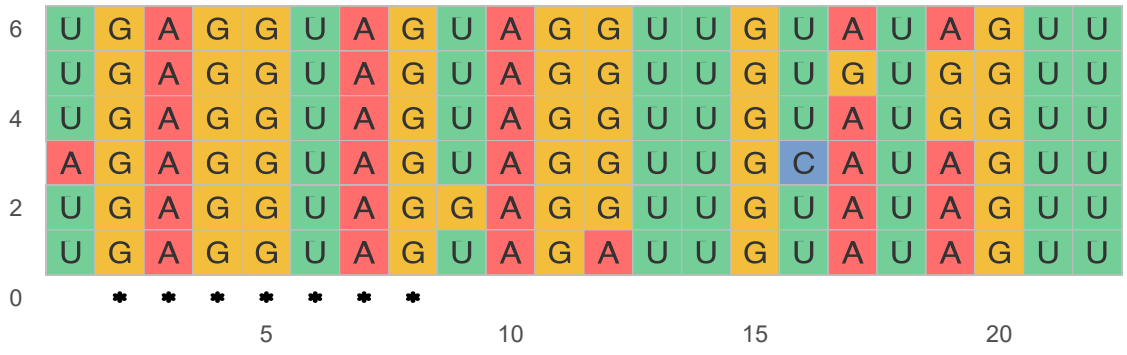

Fig. S7: Example of highlighting miRNA seed region. Asterisks are used for marking the seed region.

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()
```

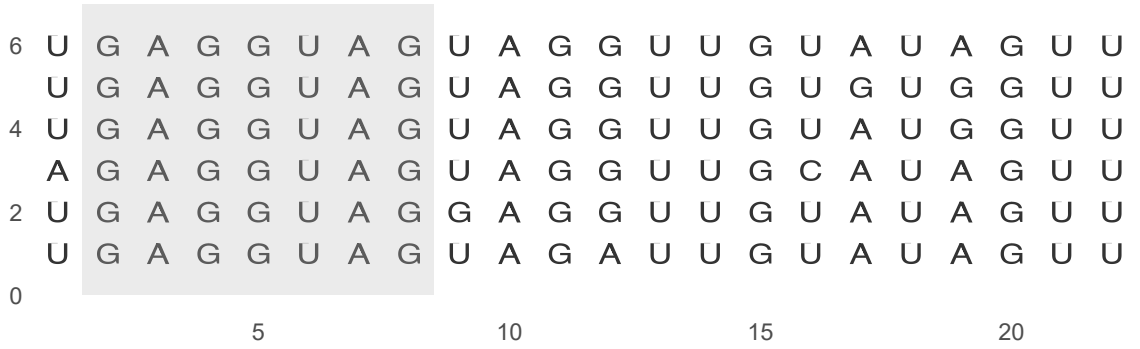

Fig. S8: Example of highlighting miRNA seed region. Annotation of sequence seed region with the shaded block

Fig. S9: The consensus sequence

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

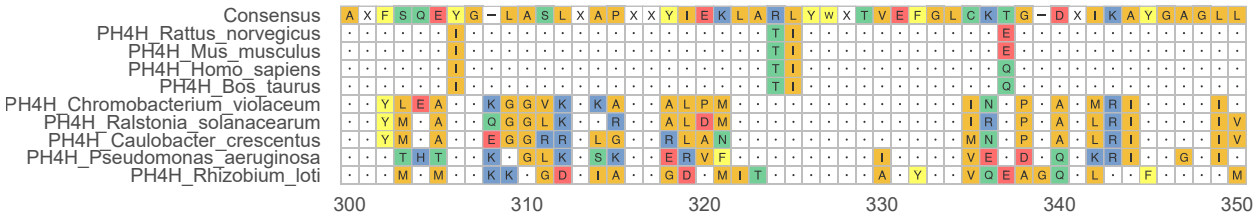

Fig. S9: Example of consensus sequence for an MSA. The consensus sequence is displayed above the alignment and shows which residues are conserved, which residues are variable. A consensus is constructed from the most frequent residues at each site.

Fig. S10: RNA secondary structure visulization using ggmsa

ggmsa supports plotting RNA secondary structure as arc diagram by reference to R4RNA (Lai et al. 2012). The 'overlapping structure diagram' helps to compare RNA secondary structure between R FAM database (known) and base-pair predicted by T RANSAT (Wiebe and Meyer 2010). The structure shown above the horizontal sequence is the known structure, colored by P-value if correctly predicted by T RANSAT (best in dark blue and worst in light blue).

```
transat_file <- system.file("extdata", "helix.txt", package = "R4RNA")
known_file <- system.file("extdata", "vienna.txt", package = "R4RNA")

known <- readSSfile(known_file, type = "Vienna" )
transat <- readSSfile(transat_file, type = "Helix" )
#gghelix(known)
gghelix(list(known = known, predicted = transat), color_by = "value", overlap = T)
```

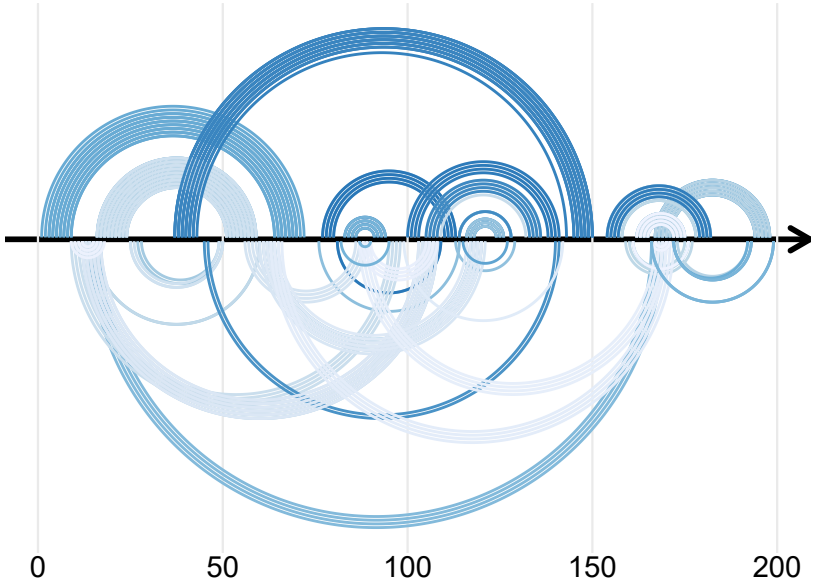

Fig. S10: An example of an overlapping structure diagram, showing the Cripavirus Internal Ribosomal Entry Site. The structure shown above the horizontal sequence is known and the structure below is predicted by T RANSAT

Fig. S11: 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. S11: The long sequence was broken down and displayed in several lines.

Fig. S12: 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)
```

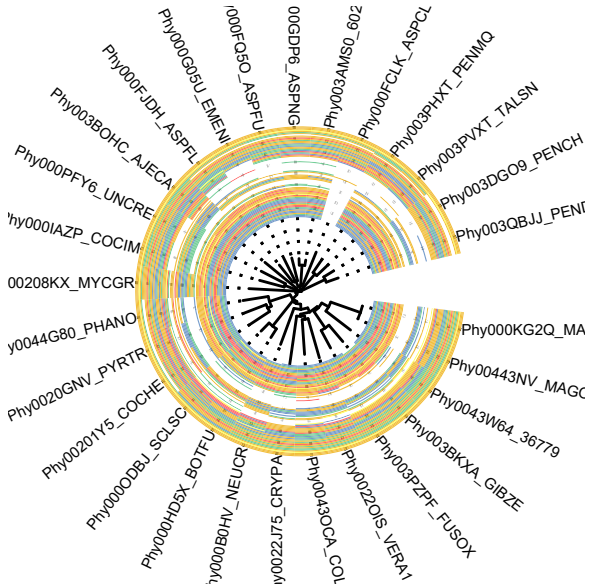

Fig. S12: Example of MSA view with circular layout phylogenetic tree.

Fig. S13: 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)
```

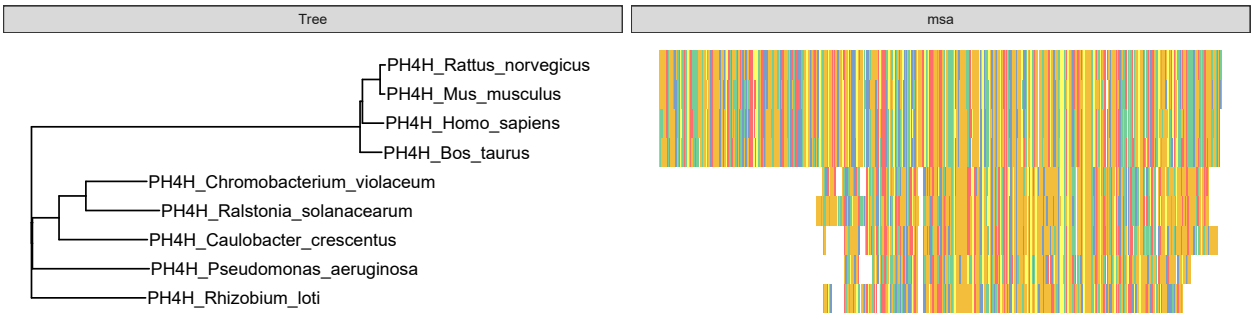

Fig. S13: A tree resulting from the application of phylogenetic analysis to the alignment.

Fig. 3A: 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()
```

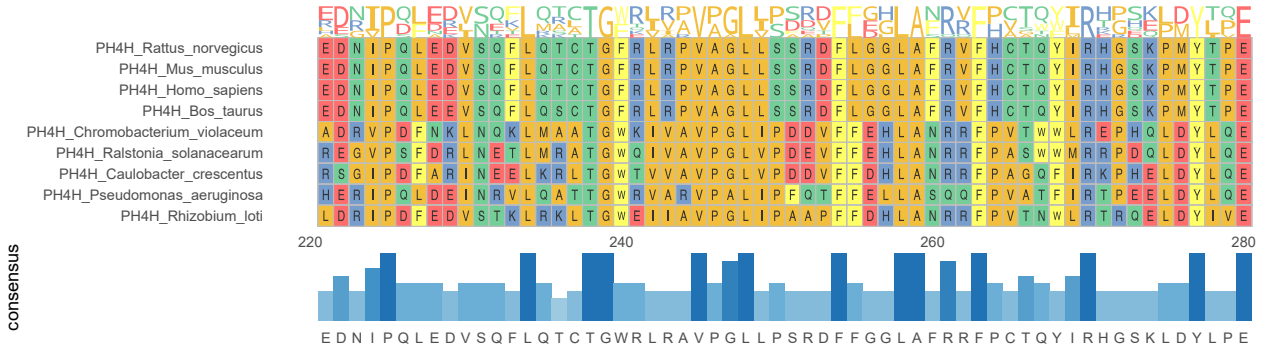

Fig. 3A|A local visualization of the sequence alignment of the phenylalanine hydroxylase protein (PH4H) within nine species. The center panel is the main MSA plot with residues in the alignment colored according to the Chemistry color scheme (amino acids are colored according to their sidechain chemistry). 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. 3B: The external data genome locus in conjunction with the MSA-tree plot (R code)

```
library(phangorn)
#tree
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))
#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
pb <- p_tp53  + xlim_tree(4) +
    geom_facet(geom = geom_msa, data = data_53,
               panel = 'msa', 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 = 'genes',  on = 'TP53',
               arrowhead_height = unit(3, "mm"),
               arrowhead_width = unit(1, "mm"))

facet_widths(pb, c(Tree = 0.35, genes = 0.3))
```

Fig. 3B|The external data, genome locus, in conjunction with the MSA-tree plot. Comparative genome locus structure (genes panel), sequence alignment of TP53 protein (msa panel) and the corresponding phylogenetic tree (Tree panel) among six organisms. The local genome map shows the 30000 sites around the TP53 gene, and the phylogenetic tree that represents evolutionary relationships of TP53 protein was inferred using the Neighbor-Joining method based on the evolutionary distances of JTT matrix-based method. The Danio rerio shows strong inconsistent in the alignment and remote evolutionary relationships in the tree comparing other organisms. And the inconsistent is also discovered in local genome map panel (e.g. linkage disequilibrium of TP53-WRAP53 locus deletions, the direction of transcription and the variety of neighboring genes). This combined visualization interprets the variant concordance of the TP53 molecule between molecular sequences and genome locus structures.

Fig. 4A: Standalone sequence logo module (R code)

```
negative <-  system.file("extdata", "Gram-negative_AKL.fasta", package = "ggmsa")
seqlogo(negative, color = "Chemistry_AA", font = "DroidSansMono")
```

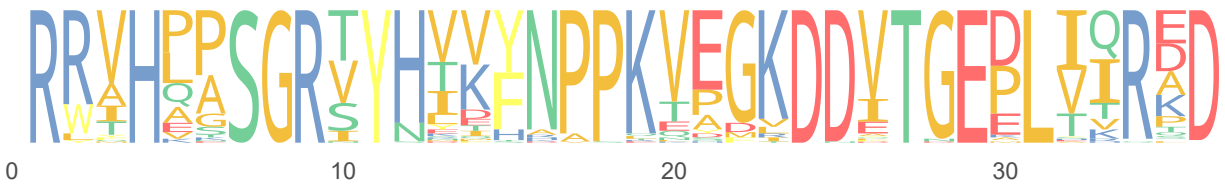

Fig. 4A| An example of protein sequence logos showing the Gram-negative AKL domain.

Fig. 4B: Visualizing MSAs as Sequence Bundles (R code)

```
ggSeqBundle(negative)
```

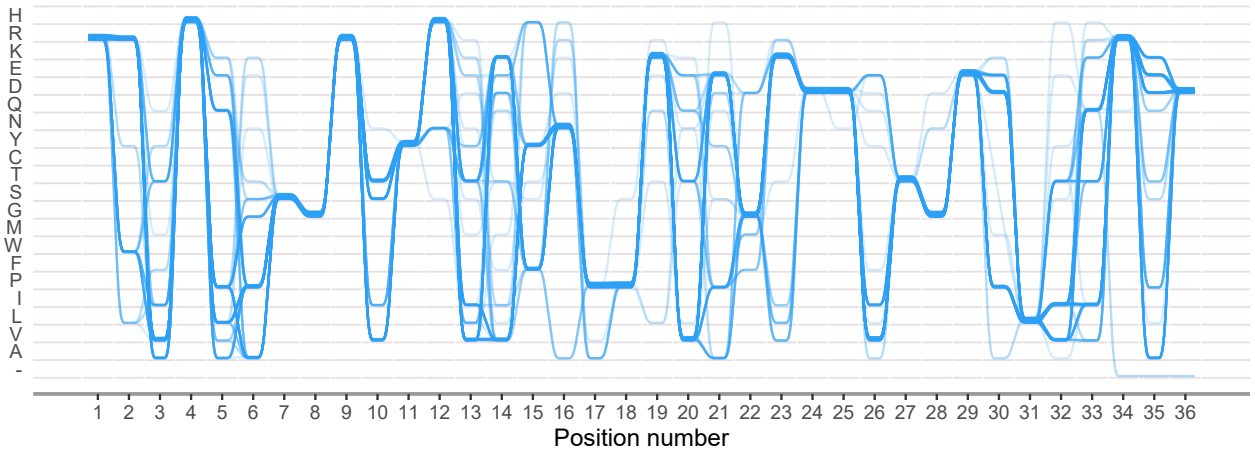

Fig. 4B| ggmsa used seqBundle function to visualize the MSA of AKL family as the new visualization technique: Sequence Bundles ("Sequence Bundles: A Novel Method for Visualising, Discovering and Exploring Sequence Motifs" 2014). It can show amino acid distribution and correlation. Symbols of the Y-axis are arranged on a scale representing amino acid hydrophobicity. The lines' curved paths expose the conservation of residues by converging at matched positions and symbols place in the Y-axis reveals patterns in functionality.

Fig. 4C: RNA secondary structure annotation for a MSA (R code)

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

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

ggmsa(RF03120_msa,
      font = NULL,
      color = "Chemistry_NT",
      seq_name = F,
      show.legend = T,
      border = NA) +
  geom_helix(helix_data = known) +
  theme(axis.text.y = element_blank())
```

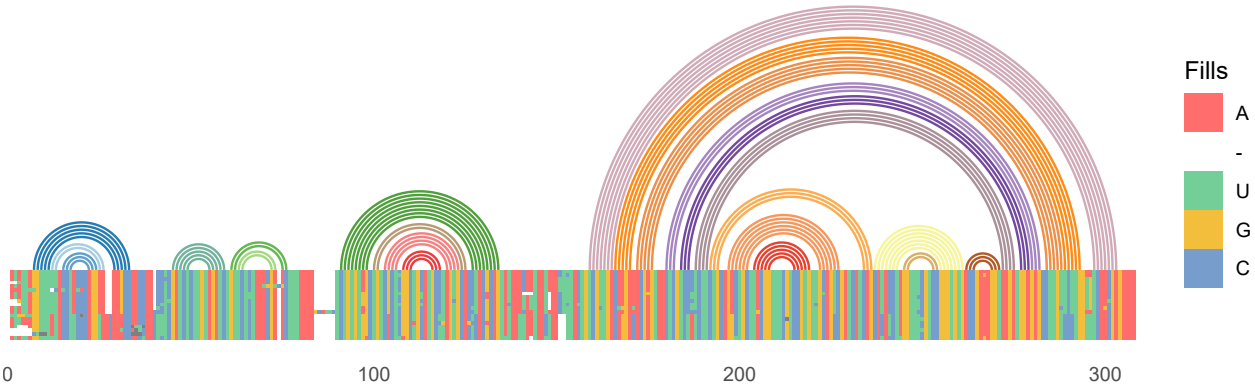

Fig. 4C|An example of visualizing MSAs in conjunction with RNA secondary structure. The data from

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. The extra RNA secondary structure data integrated with MSA plot as arc diagrams and helices were colored according to the base-pair group. The 5 UTR within the Sarbecovirus is responsible for important biological functions, such as viral replication, transcription and packaging. And it has a conserved RNA secondary structure on common Coronavirus genera. The graphical combination of the alignment and secondary structures reveals the co-variation of Sarbecovirus 5'UTR that retains the base-pairing ability, but changes the base-pairing nucleotides, and is strong evidence for RNA structure conservation.

Fig. 5: Examples of detecting sequence recombination signals (R code)

```
library(aplot)
library(ggplotify)
library(patchwork)

fas <- list.files(system.file("extdata", "GVariation", package="ggmsa"),
                  pattern="fas", full.names=TRUE)
fas <- fas[-3]

xx <- lapply(fas, seqdiff)
plts <- lapply(xx, plot)
plts[[3]] <- simplot(fas, 'CF_YL21')

plot_list(lapply(plts, function(i)as.ggplot(i)), ncol = 1) +
  plot_annotation(tag_levels = "A")
```

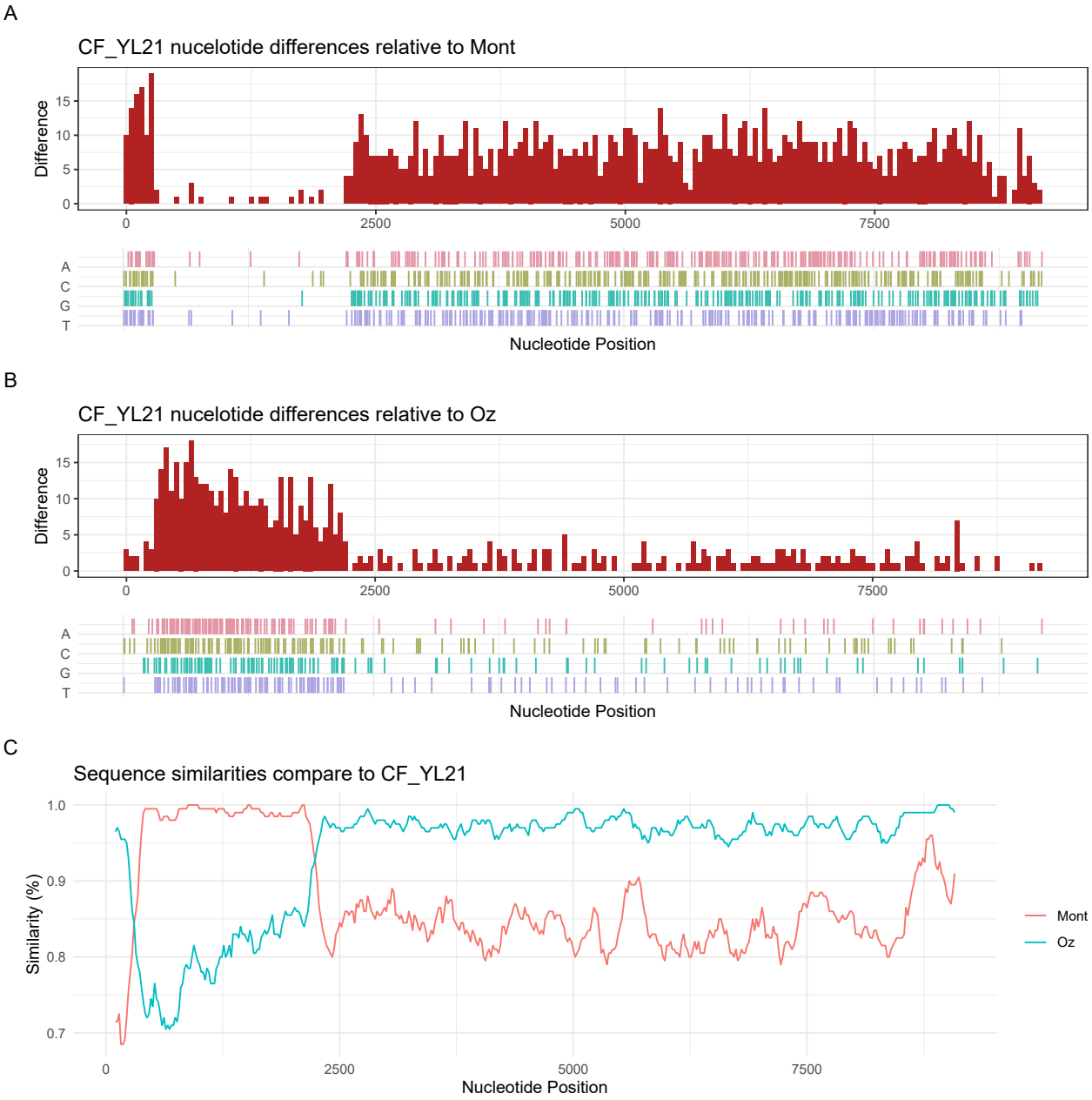

Fig. 5| Examples of detecting sequence recombination signals. CF_YL21 is a Potato virus Y (PVY) isolate from Solanum tuberosum in China. PVYN (Mont, AY884983) and PVYO (Oz, EF026074) were chosen as potential parents. (A) The comparison of CF_YL21 sequence with Mont. The x-axis corresponds to nucleotide position along the CF_YL21 genome, and the y-axis is the number of nucleotide differences between the genomes. Bar charts indicate number of base differences in every 50 sites (B) The comparison of CF_YL21 sequence with Oz (C) Detection and verification of recombination breakpoints in PVY CF_YL21 using PVYN and PVYO as reference strains. Two recombination signals (The intersection of line, site of 309 and 2224) were detected in the CF_YL21 genome.

Session Info

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

```
## R version 4.1.0 (2021-05-18)
```

```
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19043)
##
## 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    parallel  stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] ggplotify_0.0.7     aplot_0.0.6         ggimage_0.2.8
##  [4] phangorn_2.7.1      ggtreeExtra_1.2.1   patchwork_1.1.1
##  [7] pals_1.7            RColorBrewer_1.1-2  knitr_1.33
## [10] dplyr_1.0.7         ggnewscale_0.4.5    Biostrings_2.60.1
## [13] GenomeInfoDb_1.28.1 XVector_0.32.0      IRanges_2.26.0
## [16] S4Vectors_0.30.0    BiocGenerics_0.38.0 ape_5.5
## [19] gggenes_0.4.1       ggtree_3.0.2        ggplot2_3.3.5
## [22] ggmsa_1.0.1
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-152           bitops_1.0-7           httr_1.4.2
##  [4] ash_1.0-15             tools_4.1.0            utf8_1.2.2
##  [7] R6_2.5.0               KernSmooth_2.23-20     lazyeval_0.2.2
## [10] colorspace_2.0-2       withr_2.4.2            tidyselect_1.1.1
## [13] ggalt_0.4.0            curl_4.3.2             compiler_4.1.0
## [16] extrafontdb_1.0        labeling_0.4.2         scales_1.1.1
## [19] proj4_1.0-10.1         quadprog_1.5-8         stringr_1.4.0
## [22] digest_0.6.27          R4RNA_1.20.0           rmarkdown_2.9
## [25] dichromat_2.0-0        pkgconfig_2.0.3        htmltools_0.5.1.1
## [28] extrafont_0.17         highr_0.9              maps_3.3.0
## [31] readxl_1.3.1           rlang_0.4.11           gridGraphics_0.5-1
## [34] farver_2.1.0           generics_0.1.0         jsonlite_1.7.2
## [37] RCurl_1.98-1.3         magrittr_2.0.1         GenomeInfoDbData_1.2.6
## [40] Matrix_1.3-4           Rcpp_1.0.7             munsell_0.5.0
## [43] fansi_0.5.0            ggfittext_0.9.1        lifecycle_1.0.0
## [46] stringi_1.7.3          yaml_2.2.1             seqmagick_0.1.5
## [49] MASS_7.3-54            zlibbioc_1.38.0        grid_4.1.0
## [52] crayon_1.4.1           lattice_0.20-44        mapproj_1.2.7
## [55] magick_2.7.2           pillar_1.6.1           igraph_1.2.6
## [58] codetools_0.2-18       fastmatch_1.1-3        glue_1.4.2
## [61] evaluate_0.14          BiocManager_1.30.16    vctrs_0.3.8
## [64] treeio_1.16.1          tweenr_1.0.2           cellranger_1.1.0
## [67] Rttf2pt1_1.3.9         gtable_0.3.0           purrr_0.3.4
## [70] polyclip_1.10-0        tidyr_1.1.3            xfun_0.24
## [73] ggforce_0.3.3          tidytree_0.3.4         tibble_3.1.3
## [76] rvcheck_0.1.8          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.

Lai, D., J. R. Proctor, Yaz Jing, and I. M. Meyer. 2012. "R-CHIE: A Web Server and r Package for Visualizing RNA Secondary Structures." *Nucleic Acids Research* 40 (12): e95.

Nawrocki, Eric P, Sarah W Burge, Alex Bateman, Jennifer Daub, Ruth Y Eberhardt, Sean R Eddy, Evan W Floden, et al. 2015. "Rfam 12.0: Updates to the RNA Families Database." *Nucleic Acids Research* 43 (D1): D130–37.

Schneider, Thomas D, and R Michael Stephens. 1990. "Sequence Logos: A New Way to Display Consensus Sequences." *Nucleic Acids Research* 18 (20): 6097–6100.

"Sequence Bundles: A Novel Method for Visualising, Discovering and Exploring Sequence Motifs." 2014. *Bmc Proceedings* 8 (Suppl 2 Proceedings of the 3rd Annual Symposium on Biologica): S8.

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.

Wiebe, Nicholas JP, and Irmtraud M Meyer. 2010. "Transat—a Method for Detecting the Conserved Helices of Functional RNA Structures, Including Transient, Pseudo-Knotted and Alternative Structures." *PLoS Computational Biology* 6 (6): e1000823.

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.