

Variant mapping with EvoTrace

2023-04-25

This tutorial explains how to use functions in the EvoTrace package to plot variants on to lollipop plots and AlphaFold protein structures. Currently both functions only support human (ENSP id) and E. coli proteins (b number).

1. Installation

I recommend running R code with Rstudio. If you don't want to install Rstudio on you local computer, you can use Rstudio server on Taco with your BCM logins. <http://mhgcp-r00.grid.bcm.edu:8787/>, <http://mhgcp-r01.grid.bcm.edu:8787/>

EvoTrace library can be directly installed through Github. To do so, you need to setup Github token for R. But setting this up allows easy installation for future updates. Check this link for details.

```
if (!"remote" %in% installed.packages()) {  
  install.packages("remote")  
}  
remotes::install_github("LichtargeLab/EvoTrace", build_vignettes = FALSE)
```

The package can also be installed through downloaded zip file of the repo.

```
if (!"devtools" %in% installed.packages()) {  
  install.packages("devtools")  
}  
devtools::install_local("EvoTrace-master.zip")
```

2. Prepare input file

The input files for both lollipop plots and structure mappings are R dataframe. For lollipop plots “SUB”, “EA” and “AC” columns are required. For mapping to protein structures, “SUB” and “EA” columns are required. Single lettere amino acids should be used in the SUB column, eg, G18T. “AC” stands for allele count. Mutations from basS are used as example. Your own data can be loaded with `readr::read_csv()` or `readr::read_tsv()` functions.

```
# attach library  
library(EvoTrace)
```

```
# protein id use full ENSP for human protein and b number for E coli protein  
prot_id <- "b4112"
```

```
# load mutations  
basS_muts <- read_tsv(file.path(system.file("extdata", package = "EvoTrace"),  
                                     "basS_muts.tsv"),
```

```

show_col_types = FALSE)
# generate random allele counts
set.seed(566)
mut_case <- basS_muts %>%
  mutate(AC = runif(11, 0, 1000)) %>%
  mutate(AC = round(AC))

mut_controls <- basS_muts %>%
  mutate(AC = runif(11, 0, 300)) %>%
  mutate(AC = round(AC))

mut_case

```

```

## # A tibble: 11 x 3
##   SUB      EA    AC
##   <chr> <dbl> <dbl>
## 1 D312N  31.8  179
## 2 R105H  78.8  438
## 3 C84R   73.9  909
## 4 L10P   48.8  381
## 5 P94L   94.9  323
## 6 L14P   85.2  977
## 7 D149V  95.1  778
## 8 D149Y  89.4  603
## 9 V88E   91.9  552
## 10 A20D  27.2  415
## 11 L10R  37.8  852

```

```
mut_controls
```

```

## # A tibble: 11 x 3
##   SUB      EA    AC
##   <chr> <dbl> <dbl>
## 1 D312N  31.8  281
## 2 R105H  78.8   37
## 3 C84R   73.9   84
## 4 L10P   48.8   85
## 5 P94L   94.9  224
## 6 L14P   85.2  140
## 7 D149V  95.1  136
## 8 D149Y  89.4  210
## 9 V88E   91.9  142
## 10 A20D  27.2  167
## 11 L10R  37.8  218

```

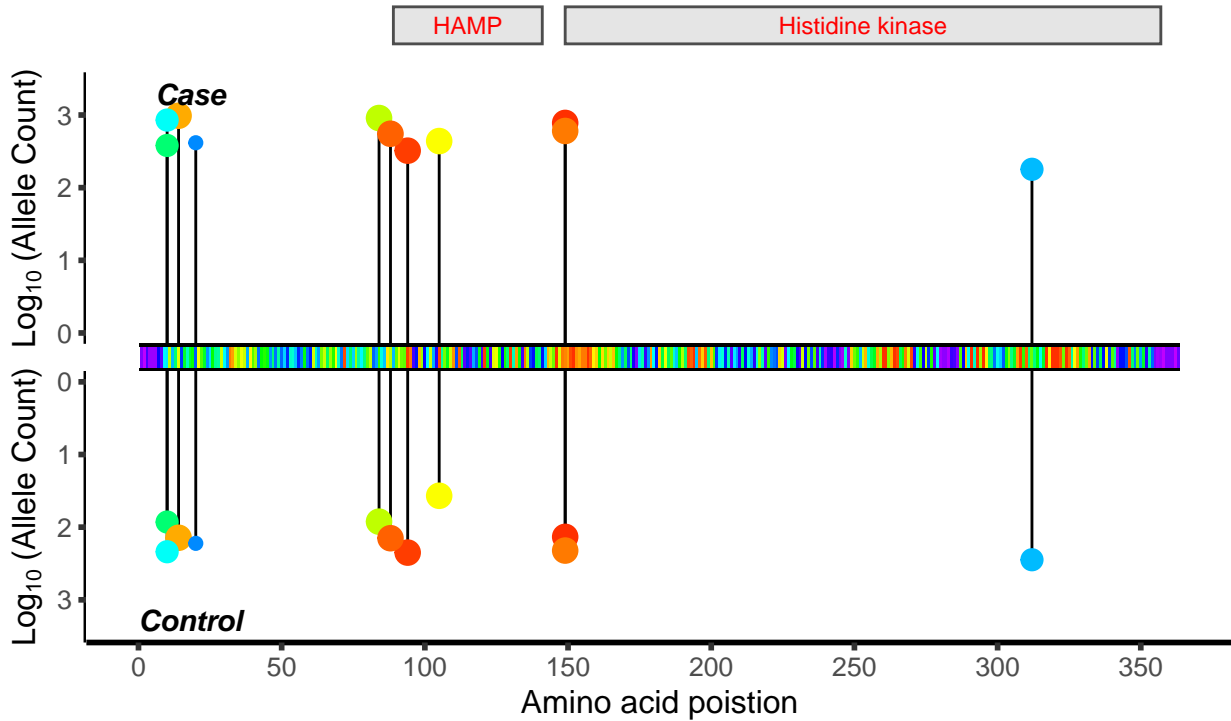
3. Lollipop plot

Lollipop plot for case vs control can be plotted with `LollipopPlot2()` Lollipop plot for only case can be plotted with `Lollipop()` Protein domains are extracted from Uniprot. The detailed parameters for each function can be accessed with:

```
?LollipopPlot2
?LollipopPlot
```

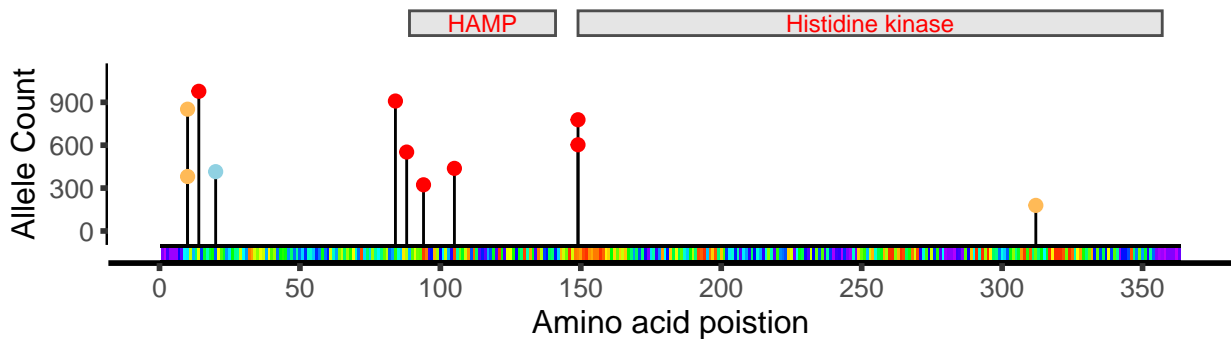
Case only lollipop plot. Here the circles on the lollipop plot are scaled with EA bins.

```
LollipopPlot2(variants_case = mut_case, variants_ctrl = mut_controls,
              prot_id = prot_id, plot_domain = TRUE, AC_scale = "log",
              show_EA_bin = TRUE, fix_scale = TRUE, EA_color = "prismatic")
```



Case vs control lollipop plot. Linear scale is used, and circles are colored with EA bins.

```
LollipopPlot(variants = mut_case, prot_id = prot_id, plot_domain = TRUE, AC_scale = "linear",
             show_EA_bin = FALSE, fix_scale = TRUE, EA_color = "EA_bin")
```



The plots can be save with ggsave() function or through the Plot-Export button in Rstudio.

4. Map variants and ET to AlphaFold structures.

The detailed parameters for the `Color_Variants_AlphaFold()` can be accessed through:

```
?Color_Variants_AlphaFold
```

`Color_Variants_AlphaFold()` will fetch AlphaFold structure and precalculated ET scores. A pymol script will be generated. The structure will be colored based on ET, AlphaFold prediction accuracy score pLDDT, EA score for mutations in cases, and EA score of mutations in controls. If multiple mutations occur at the same position, the max EA will be used. Prismatic colors are used for EA. EA = 100 is colored red, EA = 0 is colored purple. The color code and correspondent values (ET, pLDDT, EA) are returned as a dataframe.

```
color_df <- Color_Variants_AlphaFold(variants_case = mut_case, variants_ctrl = mut_controls,
                                     prot_id = prot_id, pml_output = "basS.pml")

# Examine ET scores for mutated positions
color_df_filt <- color_df %>%
  filter(!is.na(EA_case)) | (!is.na(EA_ctrl)))
color_df_filt
```

```
## # A tibble: 9 x 11
##   AA_POS AA      ET chain pLDDT EA_case EA_ctrl ET_co~1 pLDDT~2 EA_ca~3 EA_ct~4
##   <dbl> <chr> <dbl> <chr> <dbl> <dbl> <dbl> <chr> <chr> <chr> <chr>
## 1    10 L     64.9 A     76.1  48.8  48.8 0x00ec~ 0x65cb~ 0x00ff~ 0x00ff~
## 2    14 L     23.5 A     82.4  85.2  85.2 0xe4ff~ 0x65cb~ 0xffab~ 0xffab~
## 3    20 A     76.2 A     86.7  27.2  27.2 0x0059~ 0x65cb~ 0x008a~ 0x008a~
## 4    84 C     38.8 A      88    73.9  73.9 0x2cff~ 0x65cb~ 0xbfff~ 0xbfff~
## 5    88 V      28    A     84.5  91.9  91.9 0xa7ff~ 0x65cb~ 0xff61~ 0xff61~
## 6    94 P      5.69 A     77.0  94.9  94.9 0xff3d~ 0x65cb~ 0xff3d~ 0xff3d~
## 7   105 R      6.33 A     66.9  78.8  78.8 0xff49~ 0xfcdb~ 0xfcff~ 0xfcff~
## 8   149 D     11.7 A     51.5  95.1  95.1 0xff86~ 0xfcdb~ 0xff30~ 0xff30~
## 9   312 D     33.0 A     54.4  31.8  31.8 0x6aff~ 0xfcdb~ 0x00bb~ 0x00bb~
## # ... with abbreviated variable names 1: ET_color, 2: pLDDT_color,
## #   3: EA_case_color, 4: EA_ctrl_color
```

5. Legends

Currently the legends are not produced with the lollipop plots or pymol sessions. See Figure 2-4 for ET, EA (prismatic), and AlphaFold pLDDT legends.

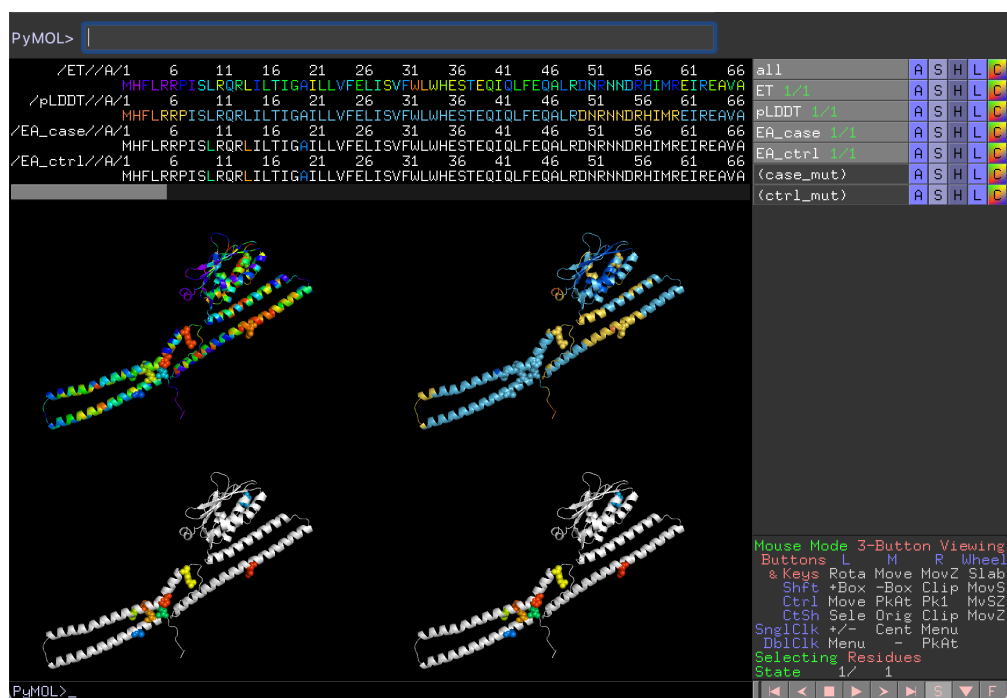


Figure 1: Pymol screenshot



Figure 2: ET legend



Figure 3: EA prismatic legend



Figure 4: pLDDT legend