Package ‘koolmaps3d’

December 6, 2019

Type Package

Title Makes your heatmap 3D!

Version 0.1.0

Author The Hell Boy George Michael Jordan Creeds

Description Takes your data makes 3d heatmap comparing two populations.
Use four spaces when indenting paragraphs within the Description.

License GPL-3

Encoding UTF-8

LazyData true

Imports rayshader, ggplot2, devtools (>= 2.0.0), dplyr, Hmisc, tibble,
tidyr, magrittr, reshape

Suggests rmarkdown, knitr, testthat

VignetteBuilder knitr

NeedsCompilation no

Maintainer Jordan Creed `<Jordan.H.Creed@moffitt.org>`

R topics documented:

`dual_matrix` *Creates matrix to plot.*

Description

Takes two long format data sets and combines into a single matrix for plotting in a 3D KOOL MAP.

Usage

```
dual_matrix(dataset1, dataset2, snp1, snp2, pos1, pos2, ld)
```

Arguments

`dataset1`	Data set with position 1 and 2 with rs numbers 1 and 2 and some measure of ld to plot.
`dataset2`	Same as data set 1 but different population
`snp1`	Col name of snp1
`snp2`	Col name of snp1
`pos1`	Col name of position 1
`pos2`	Col name of position 2
`ld`	Col name of ld measure

Value

Returns a matrix to plot

Note

no more notes

Author(s)

THe Hell Boy George Michael Jordan Creeds

References

None

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (dataset1, dataset2, snp1, snp2, pos1, pos2, ld)
{
    dataset1 <- dataset1 %>% dplyr::mutate(rsnum = paste(dataset1[[snp1]],
        "|", dataset1[[snp2]], sep = ""), pos = paste(dataset1[[pos1]],
        "|", dataset1[[pos2]], sep = ""))
    dataset2 <- dataset2 %>% dplyr::mutate(rsnum = paste(dataset2[[snp1]],
        "|", dataset2[[snp2]], sep = ""), pos = paste(dataset2[[pos1]],
        "|", dataset2[[pos2]], sep = ""))
    rsnum <- setdiff(dataset1[[rsnum]], dataset2[[rsnum]])
    pos <- setdiff(dataset1[[pos]], dataset2[[pos]])
    rowstoadd <- data.frame(rsnum, pos, stringsAsFactors = FALSE)
    dftest <- bind_rows(dataset1, rowstoadd)
    dftestv2 <- bind_rows(dataset2, rowstoadd)
    wtf <- strsplit(dftest$rsnum, "|", fixed = TRUE)
    rsdf <- do.call(rbind.data.frame, wtf)
    names(rsdf) <- c("rs1", "rs2")
    rsdf$rs1 <- as.character(rsdf$rs1)
    rsdf$rs2 <- as.character(rsdf$rs2)
    pos2 <- strsplit(dftest$pos, "|", fixed = TRUE)
    posdf <- do.call(rbind.data.frame, pos2)
    names(posdf) <- c("pos1", "pos2")
    rsdf$pos1 <- as.character(posdf$pos1)
    rsdf$pos2 <- as.character(posdf$pos2)
    dftest2 <- dftest2 %>% dplyr::mutate(X1 = ifelse(is.na(X1),
        pos1, X1), X2 = ifelse(is.na(X2), pos1, X2), X4 = ifelse(is.na(X4),
        rs1, X4), X5 = ifelse(is.na(X5), rs2, X5))
    bad_snp <- c(setdiff(dftest2$pos1, dftest2$pos2), setdiff(dftest2$pos2,
        dftest2$pos1))
    dftest3 <- dftest2 %>% dplyr::mutate(pos1 = as.numeric(pos1)) %>%
        dplyr::mutate(pos2 = as.numeric(pos2)) %>% left_join(dftestv2 %>%
        select(pos, other_ld = {
            {
                ld
            }
        }), by = c("pos")) %>% dplyr::mutate(ld1 = {
        {
            ld
        }
    }) %>% dplyr::mutate(ld1 = replace_na(ld1, 0)) %>% dplyr::mutate(other_ld = replace_r
        0)) %>% arrange(pos1, pos2) %>% dplyr::mutate(pos1 = as_factor(pos1)) %>%
        dplyr::mutate(pos2 = as_factor(pos2))
    data1 <- dftest3 %>% dplyr::select(pos1, pos2, ld1) %>% dplyr::pivot_wider(names_from
        values_from = ld1, values_fill = list(ld1 = 0)) %>% magrittr::set_rownames(.$pos1
        dplryr::select(-pos1) %>% as.matrix
    data2 <- dftest3 %>% dplyr::select(pos1, pos2, other_ld) %>%
        dplyr::pivot_wider(names_from = pos2, values_from = other_ld,
            values_fill = list(other_ld = 0)) %>% magrittr::set_rownames(.$pos1) %>%
        dplyr::select(-pos1) %>% as.matrix
    data1 <- data1[!(row.names(data1) %in% bad_snp), !(colnames(data1) %in%
        bad_snp)]
    data2 <- data2[!(row.names(data2) %in% bad_snp), !(colnames(data2) %in%
        bad_snp)]
    plot_data <- matrix(NA, nrow = nrow(data1), ncol = ncol(data1))
    plot_data[upper.tri(plot_data)] <- data1[upper.tri(data1,
        diag = FALSE)]
    plot_data[lower.tri(plot_data)] <- data2[upper.tri(data2,
        diag = FALSE)]
    row.names(plot_data) <- row.names(data1)
    colnames(plot_data) <- colnames(data1)
  }
```

`kool_plot` *Creates a KOOL MAP! (Creates 3D heat map and movie)*

Description

Function that takes output from dual_matrix function and returns a static 3D plot or movie

Usage

```
kool_plot(data_matrix, movie)
```

Arguments

`data_matrix`	Matrix from dual_matrix function.
`movie`	Location to save movie leave blank for static image = ""

Details

Function

Value

A KOOL MAP movie or image

Author(s)

The HELL BOY GEORGE MICHAEL JORDAN CREEDS

References

None

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (data_matrix, movie)
{
    df <- reshape::melt(data_matrix)
    df$value <- as.numeric(df$value)
    p <- ggplot2::ggplot(df, aes(x = X1, y = X2)) + ggplot2::geom_tile(aes(fill = value),
        color = "white") + ggplot2::coord_equal() + ggplot2::scale_fill_viridis_c(NULL,
        option = "plasma") + ggplot2::theme_minimal() + ggplot2::geom_abline(intercept =
        slope = 1, color = "white", size = 2) + ggplot2::theme(axis.title = element_blank
        legend.position = "bottom", axis.ticks = element_blank(),
        axis.text.x = element_text(angle = 60, hjust = 1))
    if (movie == "") {
        return(p)
    }
    else {
        rayshader::plot_gg(p, width = 5, height = 5)
        rayshader::render_movie(movie, frames = 600)
    }
  }
```

`plot_data` *plot_data*

Description

LD from HapMap CEU and JPT populations

Format

The format is: num [1:103, 1:103] NA 0.007 0.015 0.449 0.007 1 0.454 0.002 0.425 0.786 ... - attr(*, "dimnames")=List of 2 ..$: chr [1:103] "17766858" "17773336" "17773458" "17778418" ... ..$: chr [1:103] "17766858" "17773336" "17773458" "17778418" ...

Details

103 SNPS

Source

HapMap

References

HapMap

Examples

```
data(plot_data)
## maybe str(plot_data) ; plot(plot_data) ...
```