plot\_betas.R

z3254626

2021-07-07

Take one\_locus\_data and return a plot of beta diversities

.. content for ..

@title

plot\_betas <- function(one\_locus\_data, measures, colours,  
 expected = T, errorbars = T, insert = T) {  
 #select data   
 data <- one\_locus\_data$data\_summary  
   
 #select variables to print at the end  
 variables <- one\_locus\_data$variables  
   
 #calculate n for standard error   
 n.se <- variables$n.loci  
   
 #create base plot  
 p <- ggplot(data) +  
 ylab("Adjacent Beta Diversity") +  
 xlab("Distance (0 to 1)") +  
 xlim(0, 1) +  
 theme\_classic()  
   
   
 if("all.AvLast" %in% measures) {measures <- c(measures,   
 "H0b.Jac.AvLast", "H0b.Sor.AvLast",  
 "H1b.MI.AvLast", "H1b.ShD.AvLast",  
 "H2b.JOST.AvLast", "H2b.GST.AvLast",  
 "D0b.A.AvLast", "D0b.B.AvLast",  
 "D1b.A.AvLast", "D1b.B.AvLast",  
 "D2b.A.AvLast", "D2b.B.AvLast", "BC.AvLast")}  
   
 if("all.H.AvLast" %in% measures) {measures <- c(measures,  
 "H0b.Jac.AvLast", "H0b.Sor.AvLast",  
 "H1b.MI.AvLast", "H1b.ShD.AvLast",  
 "H2b.JOST.AvLast", "H2b.GST.AvLast")}  
   
 if("all.D.AvLast" %in% measures) {measures <- c(measures,  
 "D0b.A.AvLast", "D0b.B.AvLast",  
 "D1b.A.AvLast", "D1b.B.AvLast",  
 "D2b.A.AvLast", "D2b.B.AvLast")}  
   
 if("all.AvFirst" %in% measures) {measures <- c(measures,   
 "H0b.Jac.AvFirst", "H0b.Sor.AvFirst",  
 "H1b.MI.AvFirst", "H1b.ShD.AvFirst",  
 "H2b.JOST.AvFirst", "H2b.GST.AvFirst",  
 "D0b.A.AvFirst", "D0b.B.AvFirst",  
 "D1b.A.AvFirst", "D1b.B.AvFirst",  
 "D2b.A.AvFirst", "D2b.B.AvFirst")}  
   
 if("all.H.AvFirst" %in% measures) {measures <- c(measures,  
 "H0b.Jac.AvFirst", "H0b.Sor.AvFirst",  
 "H1b.MI.AvFirst", "H1b.ShD.AvFirst",  
 "H2b.JOST.AvFirst", "H2b.GST.AvFirst")}  
   
 if("all.D.AvFirst" %in% measures) {measures <- c(measures,  
 "D0b.A.AvFirst", "D0b.B.AvFirst",  
 "D1b.A.AvFirst", "D1b.B.AvFirst",  
 "D2b.A.AvFirst", "D2b.B.AvFirst")}  
   
 #Function for adding error bars to plots  
 gg\_errorbars <- function(measure, colour){  
   
 D\_adjust <- ifelse(startsWith(measures[i], "D") == T, -1, 0)  
   
   
 # Calculate standard errors  
 data <- data %>%  
 mutate(measure\_se = !!sym(paste0(measure, "\_sd"))/sqrt(n.se),  
 measure\_mean = !!sym(paste0(measure, "\_mean")))  
   
 p <- p +  
 geom\_errorbar( data = data,  
 aes(  
 x = d + i\_mean / 2,  
 y = measure\_mean + D\_adjust,  
 ymin = ifelse(measure\_mean - measure\_se + D\_adjust < 0, 0,   
 measure\_mean - measure\_se + D\_adjust),  
 ymax = measure\_mean + measure\_se +D\_adjust  
 ),  
 colour = colour,  
 width = .05,  
 position = position\_dodge(10))  
   
 return(p)  
 }  
   
   
   
   
 for (i in seq(1, length(measures))) {  
   
 D\_adjust <- ifelse(startsWith(measures[i], "D") == T, -1, 0)  
  
 p <- p +   
 geom\_point(aes(d + i\_mean / 2, !!sym((paste0(measures[i], "\_mean"))) + !!D\_adjust), colour = colours[i]) +  
 geom\_line(aes(d + i\_mean / 2, !!sym((paste0(measures[i], "\_mean"))) + !!D\_adjust), colour = colours[i])  
   
 if(errorbars == T) p <- gg\_errorbars(measures[i], colours[i])  
   
 }  
   
 #Toggle of an iset plot of the underlying allele proportions  
 if(insert == T){  
   
 inset.plot <- ggplot(data) +  
 geom\_point(aes(d, p\_mean))+  
 theme(  
 panel.background = element\_rect(fill = "transparent"), # bg of the panel  
 plot.background = element\_rect(fill = "transparent", color = NA), # bg of the plot  
 panel.grid.major = element\_blank(), # get rid of major grid  
 panel.grid.minor = element\_blank(), # get rid of minor grid  
 legend.background = element\_rect(fill = "transparent"), # get rid of legend bg  
 legend.box.background = element\_rect(fill = "transparent") # get rid of legend panel bg  
 )  
   
 p <- ggdraw(p) +  
 draw\_plot(inset.plot, x = 0.7, y = .7, width = .3, height = .3)  
   
 }  
  
 return(list(beta\_plot = p, variables = variables))  
 }