visualise\_step\_detections.R

z3254626

2021-07-07

#variable - one of: n.samples, n.loci, n.pops  
  
visualise\_step\_sensitivities <- function(measure, variable, colour) {  
   
 if (variable == "n.samples") data <- readd(step\_sensitivities\_samples)  
   
 if (variable == "n.loci") data <- readd(step\_sensitivities\_loci)  
   
 if (variable == "n.pops") data <- readd(step\_sensitivities\_pops)  
  
 #Creates 4 plots for each value of step (0, 1, 5, 50)  
 # as a horizontal grid   
 #Takes inputs:  
 # measure - genetic diversity measure results to extract  
 # colour - colour of lines to be plotted  
 vis.beta.step.grid <- function(variable, p.start, p.end){  
   
 #Base properties of each plot  
 p <- ggplot()+  
 ylab("Step detections (out of 100)") +  
 ylim(0,100) +  
 theme\_classic()   
   
 #step == 0  
 p1 <- p +  
 geom\_line(data = filter(data, step == 0 & p.start == !!p.start & p.end == !!p.end),  
 aes(x= !!sym(variable), y = !!sym(measure)), colour = colour, linetype = "dashed")  
   
 #step == 1  
 p2 <- p +  
 geom\_line(data = filter(data, step == 1 & p.start == !!p.start & p.end == !!p.end),  
 aes(x= !!sym(variable), y = !!sym(measure)), colour = colour, linetype = "dashed") +  
 geom\_line(data = filter(data, step == 1 & p.start == !!p.start & p.end == !!p.end),  
 aes(x = !!sym(variable), y = !!sym(paste0(measure, "\_correct"))), colour = colour)  
   
 #step == 5  
 p3 <- p +  
 geom\_line(data = filter(data, step == 5 & p.start == !!p.start & p.end == !!p.end),  
 aes(x= !!sym(variable), y = !!sym(measure)), colour = colour, linetype = "dashed") +  
 geom\_line(data = filter(data, step == 5 & p.start == !!p.start & p.end == !!p.end),  
 aes(x = !!sym(variable), y = !!sym(paste0(measure, "\_correct"))), colour = colour)  
   
 #step == 50  
 p4 <- p +  
 geom\_line(data = filter(data, step == 50 & p.start == !!p.start & p.end == !!p.end),  
 aes(x= !!sym(variable), y = !!sym(measure)), colour = colour, linetype = "dashed") +  
 geom\_line(data = filter(data, step == 50 & p.start == !!p.start & p.end == !!p.end),  
 aes(x = !!sym(variable), y = !!sym(paste0(measure, "\_correct"))), colour = colour)  
  
 #Merge the four plots together  
 p\_grid <- plot\_grid(p1, p2, p3, p4, ncol = 4,  
 labels = c('Step = 0', 'Step = 1', 'Step = 5', 'Step = 50'),   
 label\_size = 12, hjust = -1, vjust= 5)  
   
   
 return(p\_grid)  
 }  
   
   
   
 final\_plot <- plot\_grid(  
 vis.beta.step.grid(variable, 0, 1),  
 vis.beta.step.grid(variable, 0.1, 0.9),  
 vis.beta.step.grid(variable, 0, 0.5),  
 vis.beta.step.grid(variable, 0, 0.2),  
 vis.beta.step.grid(variable, 0.3, 0.5),  
 ncol = 1,   
 labels = c("0 to 1", "0.1 to 0.9", "0 to 0.5", "0 to 0.2", "0.3 to 0.5"))  
   
   
 ggsave(final\_plot,   
 filename = paste0("./Outputs/", measure, "\_", variable, ".pdf"),  
 height = 297, width = 210, unit = "mm")  
  
}  
  
create\_measure\_pdfs <- function() {  
   
 beta\_measure\_names <- c("H0b.Jac", "H0b.Sor", "H1b.MI", "H1b.ShD",   
 "H2b.JOST", "H2b.GST", "D0b.A", "D0b.B",   
 "D1b.A", "D1b.B", "D2b.A", "D2b.B")  
   
 #Names of each beta measure including their by locus and global variant  
 beta\_measures <- c(paste0(beta\_measure\_names, ".locus"),  
 paste0(beta\_measure\_names, ".rel.locus"),   
 paste0(beta\_measure\_names, ".global"),   
 "BC.locus", "RBC.locus")  
   
 test\_variables <- c("n.samples", "n.loci", "n.pops")  
   
 sens\_vars <- cross\_df(list(beta\_measures = beta\_measures,   
 test\_variables = test\_variables))  
   
 map2(sens\_vars$beta\_measures, sens\_vars$test\_variables,   
 visualise\_step\_sensitivities, "black")  
   
}