Supplementary Information for

Spatial patterns of trematode-induced pits on bivalve skeletons: Challenges and prospects for research on parasite-host dynamics

Alexis Rojas[1*], John Warren Huntley[2], Daniele Scarponi[1]

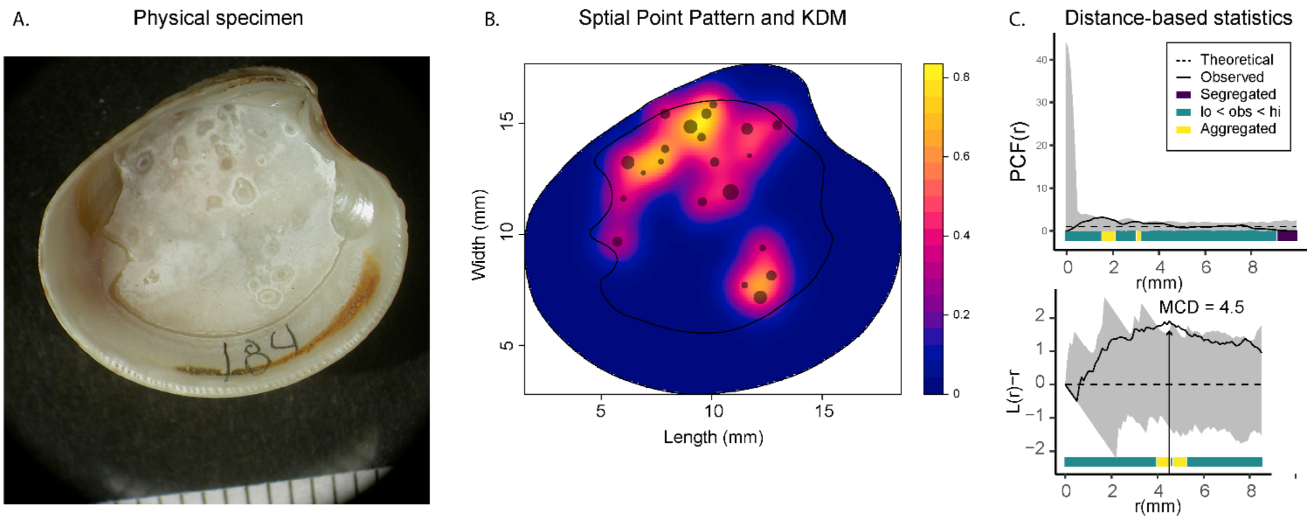
[1]Department of Biological, Geological, and Environmental Sciences, University of Bologna, Italy

[2]Department of Geological Sciences, University of Missouri, Columbia, Missouri 65211, USA

*** Correspondence:**
Corresponding Author
alexis.rojasbriceno@unibo.it

This PDF file includes:

- Supplementary Figure S1
- Supplementary Table S1
- Supplementary Legends for Supplementary Data S1 to S3.
- R script



Supplementary Figure S1. Univariate analysis. The assembled datasets give the locations of trematode-induced pits in a single specimen. **A**. Valve 240S8 13.10m 184 L; 21 pits. **B**. Point pattern and kernel density map in standard units (millimeters). B. L-function. The black arrow indicates the point of maximum clustering distance (MCD). C. Pair correlation function (PCF) and L-function. The dark gray area is the simulation envelope for 999 Monte Carlo simulations of Complete Spatial Randomness CSR (Theoretical). The empirical curves of the distance-based statistical functions (i.e., observed data) are compared to the Monte Carlo envelopes generated through simulations of the null model. A departure from the null model is indicated when the empirical curves fall outside the simulation envelopes. The maximum clustering distance (MCD = 4.5 mm) represents the scale at which pits are most strongly clustered (i.e., the distance r at which the observed value of the L-function deviates the most from the expected value under CSR). The position of the empirical curve in relation to the simulation envelope of the null model is indicated by a color bar at the bottom. Intuitively, the kernel density map indicates an aggregated patterns with two clusters. However, given the limited reduced number of traces, the graphical outputs of the distance-based statistics show marginally significant clustering.

# pits per valve	# valves	MCD	u*	rank	p.value	bandwidth	mc x	mc y	std x	std y
1	62	0.42	0.0188	1	0.001	0.04	0.0541	0.2565	0.1877	0.1842
2	12	0.4	0.0137	1	0.001	0.05	0.0839	0.2243	0.1752	0.2314
3	22	0.41	0.0177	1	0.001	0.04	0.1194	0.2540	0.2112	0.1807
4	17	0.35	0.0088	1	0.001	0.05	0.0351	0.2134	0.2027	0.2604
5	14	0.35	0.0272	1	0.001	0.04	0.1277	0.2844	0.1557	0.1665
6	2	0.26	0.0178	7	0.007	0.06	0.1683	0.2210	0.1070	0.2555
7	4	0.4	0.0265	1	0.001	0.05	0.1679	0.2858	0.1468	0.1815
6	2	0.26	0.0178	14	0.014	0.06	0.1683	0.2210	0.1070	0.2555
8	4	0.37	0.0138	1	0.001	0.05	0.1305	0.1982	0.2094	0.1846
9	5	0.47	0.0142	1	0.001	0.05	0.0995	0.2073	0.1677	0.2330
10	2	0.33	0.0152	1	0.001	0.06	0.1201	0.1710	0.1852	0.1913
11	6	0.44	0.0154	1	0.001	0.04	0.1346	0.2246	0.1875	0.2101
13	2	0.47	0.0126	1	0.001	0.06	0.1666	0.1708	0.2118	0.1985
15	1	0.51	0.0236	1	0.001	0.06	0.0969	0.3216	0.1770	0.1812
16	2	0.38	0.0092	1	0.001	0.06	0.1827	0.1042	0.1462	0.2813
19	1	0.38	0.0223	1	0.001	0.05	0.1708	0.2449	0.1592	0.1827
21	1	0.39	0.0115	1	0.001	0.07	-0.0329	0.2269	0.2077	0.2423
26	2	0.39	0.0119	1	0.001	0.05	0.1196	0.1421	0.1646	0.2269
29	1	0.36	0.0195	1	0.001	0.05	0.2310	0.2157	0.1778	0.2018
32	1	0.45	0.0082	1	0.001	0.06	0.1032	0.1558	0.2350	0.2102
34	1	0.41	0.0194	1	0.001	0.05	0.1487	0.2317	0.1833	0.1760
43	1	0.49	0.0097	1	0.001	0.05	0.0778	0.1779	0.2029	0.2307
48	1	0.35	0.0102	1	0.001	0.05	0.1833	0.1093	0.2000	0.2414
63	1	0.49	0.0045	1	0.001	0.06	-0.0609	0.1081	0.2551	0.2392

mean center (mc); unweigted standard distance (sd); *Diggle-Cressie-Loosmore-Ford test of CSR.

Supplementary Table S1. Descriptive statistics of the point patterns combining data from hosts with multiple traces (i.e., multi-parasitized shells), as illustrated in Figure 8. The assembled datasets represent the locations of trematode-induced pits from all valves exhibiting the same number of traces in core 240S8. The maximum clustering distance (MCD) indicates the scale at which pits are most strongly clustered. The mean center (mc) is the average (centroid) location of all pits in the dataset. The unweighted standard distance (sd) measures the average dispersion of points around the mean center, reflecting how spread out the points are and summarizing the spatial variability of the distribution.

Supplementary Data captions

Supplementary Data S1. Assembled point data on trematode-induced pits on *C. gallina*. This dataset provides the locations and sizes of trematode-induced pits in the examined valves. It includes the original landmark coordinates as well as the rotated, scaled, and translated coordinates obtained through Bookstein baseline registration using landmarks 3 and 6.

Supplementary Data S2. Outer area. This dataset provides rotated, scaled, and translated coordinates of the observation window obtained through Bookstein baseline registration using landmarks 3 and 6.

Supplementary Data S3. Internal area. This dataset provides rotated, scaled, and translated coordinates of the area used as a reference, obtained through Bookstein baseline registration using landmarks 3 and 6.

```
# Supplementary R Script

# Load libraries.............................................................

library(spatstat)
library(onpoint)
library(ggplot2)

 # Define paths..............................................................

path.data <- "C:/Users/data/"
path.output <- "C:/Users/output/"

# Load data .................................................................

taxa  <- "C_gallina"
internal_shell <- read.csv(paste(path.data,taxa,"_internal_area.csv", sep=""),
header=TRUE)
shell <- read.csv(paste(path.data,taxa,"_external_area.csv", sep=""), header=TRUE)
holes <- read.csv(paste(path.data,taxa,"_data.csv", sep=""), header=TRUE)

################################################################################
######                    Select the point data                           ######
################################################################################

# Filter data based on specified criteria ......................................
# OPTION 1: Select valve(s) ....................................................

#selected_Valve <- "13.1.332"# "13.1.197"#"13.1.193" #"13.1.342" #)
#experiment <- paste("Valve",selected_Valve, sep="_")
#holes <- holes[holes$Valve_Number %in% selected_Valve,]
#pits <- unique(holes$Num_Traces)
#pits <- pits[order(pits)]

# OPTION 2: Select valves with a given number of pits...........................
num_pits <- 63
experiment <- paste("shells_with", num_pits, "pits", sep="_") # Create a experiment name
holes <- holes[holes$Locality =="240S8" & holes$Num_Traces == num_pits,]
holes <- holes[complete.cases(holes$Trace_Max_Length & holes$Trace_Min_Length),]

# Define size classes
# Threshold pit size=0.5 mm following  Fitzgerald et al. 2024)..................
holes$sizeClass <- (holes$Trace_Max_Length+holes$Trace_Min_Length)/2
holes$sizeClass <- cut(holes$sizeClass, breaks=c(0,0.55, max(holes$sizeClass)),
labels=c("small","large"))
sizeClass <- c(levels(holes$sizeClass))
holes$sizeID <- as.character(holes$sizeClass)
holes$sizeID[holes$sizeID==sizeClass[1]] <- "small"
holes$sizeID[holes$sizeID==sizeClass[2]] <- "large"
size_l <- table(holes$sizeID)[1]
size_s <- table(holes$sizeID)[2]
selected_size <-  "small"# c("small","large")# "large" #"all" #
holes <- holes[holes$sizeClass %in% selected_size, ]
size_s <- length(which(holes$sizeClass=="small"))
size_l <- length(which(holes$sizeClass=="large"))

################################################################################
######                        Create Point Pattern                        ######
################################################################################

# create area..................................................................
pol <- data.frame(x=shell[,1], y=shell[,2])
pol <- as.matrix(pol)
pol <- owin(poly=pol,unitname="XY Coordinates")
```

```
# select pits..................................................................

xy <- holes[,c("BTrace_x","BTrace_y")]

# Create point pattern of traces................................................

ppp <- ppp(x=xy[,1],y=xy[,2],window=pol)
ppp <- unique(ppp, warn=T)
xy <- cbind(ppp$x, ppp$y) # rotated holes

# Plot results..................................................................

#pdf(file=paste(path.output," ",selected_size, " ", taxa, " ", paste("full_data_",
experiment, sep=""), " PPP.pdf", sep=""),height=5, width=5, pointsize=12,
useDingbats=FALSE)
plot(shell,type="l", asp=1, col="grey50", xlab="Shape (Bookstein) coordinate X",
ylab="Shape (Bookstein) coordinate Y")
#polygon(shell[,1],shell[,2], col = "grey90", border = "grey80")
polygon(internal_shell[,1],internal_shell[,2], col = "grey95", border = "grey90")
points(xy, pch=16, col=rgb(0,0,0, 0.6), cex=1.5*holes$Trace_Max_Length)
lines(internal_shell, col="grey80")
#dev.off()

# Change the scale to millimeters ..............................................
#par(mar = c(3,3,2,2))
#base_line <- 11.061 #  use distance in mm measured in the elected photograph
#shell[,1] <- shell[,1]*(base_line)+10
#shell[,2] <- shell[,2]*(base_line)+10
#internal_shell[,1] <- internal_shell[,1]*(base_line)+10
#internal_shell[,2] <- internal_shell[,2]*(base_line)+10
#xy <- cbind(holes$BTrace_x*base_line+10, holes$BTrace_y*base_line+10)

#pdf(file=paste(path.output," ",selected_size, " ", taxa, " ", paste("full_data_",
experiment, sep=""), " PPP.pdf", sep=""),height=5, width=5, pointsize=12,
useDingbats=FALSE)
plot(shell,type="l", asp=1, col="grey50", xlab="Shape (Bookstein) coordinate X",
ylab="Shape (Bookstein) coordinate Y")
#polygon(shell[,1],shell[,2], col = "grey90", border = "grey80")
polygon(internal_shell[,1],internal_shell[,2], col = "grey95", border = "grey90")
points(xy, pch=16, col=rgb(0,0,0, 0.6), cex=1.5*holes$Trace_Max_Length)
lines(internal_shell, col="grey80")
#dev.off()

# Define the values of r at which to evaluate the distance-based functions.......
# (Change it for analysis in mm)

r <- seq(0, 1, 0.01) # Shape (Bookstein) units

# Compute PCF ...................................................................
PCF_envelope <- envelope(ppp, fun=pcf, nsim=999,  verbose=TRUE,  clipdata=TRUE,global =
FALSE,
nrank=1, funargs = list(correction = "Ripley", divisor = "d"), r=r)
#pdf(file=paste(path.output,selected_size, " ", taxa, " ", paste("full_data_", experiment,
sep=""), " PCF.pdf", sep=""),height=5, width=5, pointsize=12, useDingbats=FALSE)
#plot(PCF_envelope, xlim=c(0,0.8), ylim=c(0,15), main=paste(selected_size, taxa, "PCF",
sep=" "))
plot_quantums(PCF_envelope,ylab="PCF(r)", xlab="r(Shape Bookstein units")
#dev.off()

# Compute L function.............................................................
r <- seq(0, 1, 0.01)
L_envelope <- envelope(ppp, fun=Lest, r=r,nsim=999, var.approx=TRUE, correction="best")
L <- center_l_function(L_envelope)
#pdf(file=paste(path.output, selected_size, " ", taxa, " 2", experiment, " L-
```

```
function.pdf", sep=""),height=5, width=5, pointsize=12, useDingbats=FALSE)
p <- plot_quantums(L, ylab="L(r)-r", xlab="r(Shape Bookstein units")
# Plot maximum clustering distance
MCD <- data.frame(mcd=abs(L$obs-L$theo), r=r)
MCD <-  MCD[!is.na(MCD["mcd"]),]
r_mcd <- MCD[which(MCD$mcd ==  max(MCD$mcd)), 2]
p + geom_vline(xintercept=r_mcd) + geom_text(data=NULL, label=r_mcd, x=0.4, y=0.1)
#dev.off()

# Compute Oring function.................................................>.......
o_ring_envelope <- envelope(ppp, r=r, fun=estimate_o_ring, nsim=999)
#pdf(file=paste(path.output,selected_size, " ", taxa, " ", experiment, " o_ring.pdf",
sep=""),height=5, width=5, pointsize=12, useDingbats=FALSE)
#plot(o_ring_envelope, xlim=c(0, 0.8), ylim=c(0, 200), main=paste(selected_size, taxa, "O-
ring statistic",sep=" "))
plot_quantums(o_ring_envelope, ylab="g(r)λ", xlab="r(Shape Bookstein units)")
#dev.off()

################################################################################
######                                  Descriptive statistics
#######
################################################################################

# unweigthed mean center (mc)..................................................

mc   <- apply(xy, 2, mean)
mc.x <- as.numeric(mc[1])
mc.y <- as.numeric(mc[2])

# standard deviation in x and y..................................................

std.x <- sd(xy[,1])
std.y <- sd(xy[,2])

# unweigted standard distance (sdc)..............................................

sdn <- sqrt(sum((xy[,1] - mc[1])^2 + (xy[,2] - mc[2])^2) / nrow(xy))

# unweigted bandwidth.............................................

SD1 <- sdn
H1 <- ((2/(3*(dim(xy)[1])))^(1/4))*SD1
h1 <- H1/2                                                      # sigma

###############################################################################
######                                                       Plots
######
###############################################################################

# # Kernel Density Estimation...................................................

h1 <- round(h1,2)
#pdf(file=paste(path.output, selected_size, " ", taxa, " ", paste("full_data_",
experiment, sep=""), "(h=", h1, ") KDE.pdf", sep=""),height=5, width=5, pointsize=6,
useDingbats=FALSE)
par(mar = c(2,2,2,2))
den <- density.ppp(ppp, sigma=h1, at="pixels", edge=F, eps=0.01)
plot.im(den, main=paste(selected_size, taxa, "h:", " ", h1, "", "KDE",sep=" "), ribn=20,
        ribargs=list(las=1, nint=3, riblab="Density",ribscale=1))
points(xy, pch=16, col=rgb(0,0,0, alpha=0.5), cex=2.5*holes$Trace_Max_Length)
lines(internal_shell)
lines(shell)
#dev.off()

# Explore r obs theo and max...........................................
```

```
L$MAX <- L$obs-L$theo
sum <-data.frame(r=round(L$r,4), obs=round(L$obs,4), theo=round(L$theo,4),
max=round(L$MAX,4))
sum <- na.omit(sum)
sum[sum$max==max(sum$max),]

# goodness-of-fit test...............................................
# sensu Loosemore and Ford (2006)

gof <- dclf.test(ppp, r=r, Lest, nsim=999)
gof

# Descriptive statistics ...........................................

sum_data <- NULL
sum_data_i <- data.frame(num_pits=num_pits, obs_p=ppp$n, size_s=size_s, size_l=size_l,
MCD=r_mcd,      u=gof[[1]]$u,
           rank=gof[[1]]$rank, p_value=gof[2],bandwidth=h1, mc_x=mc.x, mc_y=mc.y,
std_x=std.x, std_y=std.y)
sum_data <- rbind(sum_data, sum_data_i)
#write.csv(sum_data, file=paste(path.output,experiment,"_sum_data.csv",sep=""))

################################################################################
```