

Fig. S1. Discriminant function analysis (DFA) of morphological characteristics of clingfish, lumpsucker, and snailfish adhesive discs. Arrows show the direction of loading and relative ability of the measured traits at discriminating the three species.

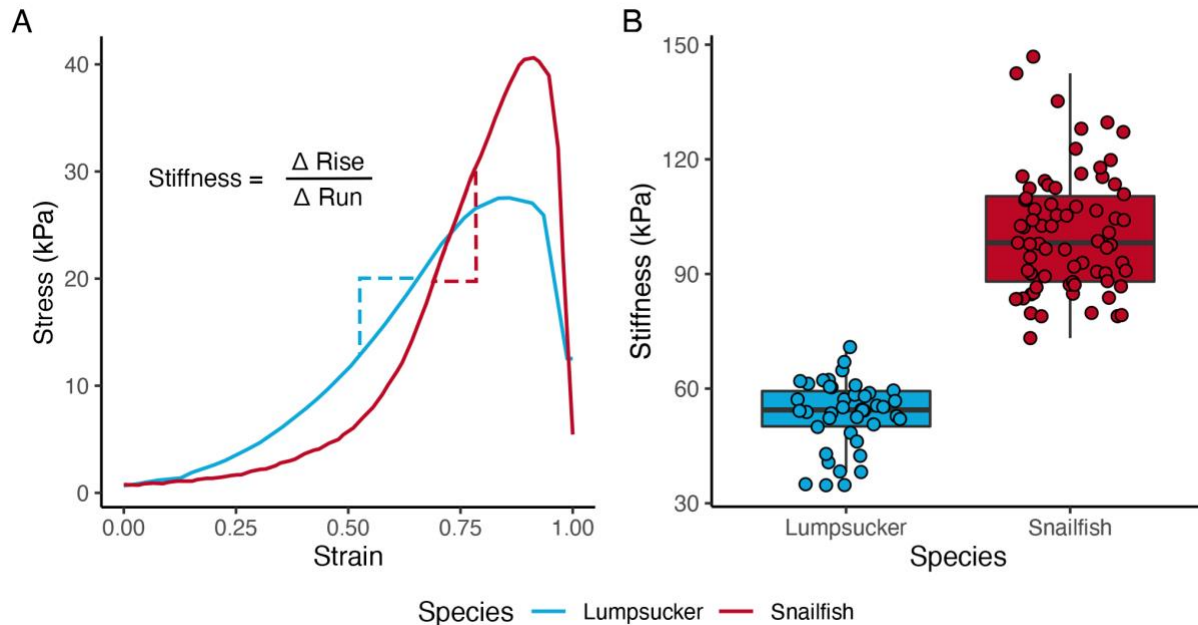


Fig. S2. Example A) stress-strain curves from the lumpsucker and snailfish pull-off trials and B) estimates of stiffness. The stiffness of the lumpsucker and snailfish adhesive discs were estimated by calculating the slope of stress-strain curves at the linear portion.

Table S1. Adhesion performance data for the clingfish, lumpsucker, and snailfish on surfaces of varying roughness.

[Click here to download Table S1](#)

Table S2. Load extension curve data for the best clingfish, lumpsucker, and snailfish suction adhesion trails.

[Click here to download Table S2](#)

Table S3. Adhesive disc morphometric data measured from micro-CT scans for clingfish, lumpsucker, and snailfish.

[Click here to download Table S3](#)

Table S4. Body mass and force per body weight estimates for *Liparis dennyi* using different mass~SL relationships.

		<i>Liparis pulchellus</i>	<i>Liparis otochensis</i>	FishBase
Body Mass (g)	Range	10.93 – 42.40	13.43 – 47.74	6.85 – 25.41
	Mean \pm s.d.	23.37 \pm 9.75	27.20 \pm 10.63	14.25 \pm 5.93
Force x BW	Range	15.11 – 38.96	13.42 – 32.31	25.22 – 62.73
	Mean \pm s.d.	27.54 \pm 5.51	23.33 \pm 4.21	44.83 \pm 8.50

Supplementary Materials and Methods

R script file with code to perform the analyses and plot the figures used in this study.

```
# Code written by: JMH
# Last updated: October 2022

# Set Up -----
# clear previous environment
rm(list = ls(all=TRUE))

# load in necessary packages
library(dplyr)
library(ggplot2)
library(patchwork)
library(lme4)
library(emmeans)
library(performance)
library(lmodel2)
library(MASS)

# Load Data -----

AllData <- read.csv("Table S1.csv", na.strings = "")
colnames(AllData) <- c("Species", "Specimen", "SL", "Mass", "DA", "Trial", "Surface",
  "Force", "x_BW", "Stress", "Work", "Stiffness")
AllData$Species <- factor(AllData$Species, level = c("EO", "LD", "GM"))
col = setNames(c("#0da8db", "#bd0825", "#FE9F6D"), c("EO", "LD", "GM"))

#estimate snailfish mass
#with Liparis pulchellus
AllData$Mass[AllData$Species == "LD"] <- exp(log(AllData$SL[which(AllData$Species ==
"LD"])*10)*3.26+(-12.54))

# with Liparis ochotensis
#AllData$Mass[AllData$Species == "LD"] <- (as.numeric(AllData$SL[which(AllData$Species ==
"LD"]))^3.05)*0.01288*1.00260

# with FishBase
#AllData$Mass[AllData$Species == "LD"] <- (as.numeric(AllData$SL[which(AllData$Species ==
"LD"]))^3.15)*0.00525

AllData$Mass <- as.numeric(AllData$Mass)
AllData$x_BW <- AllData$Force/(AllData$Mass/1000*9.8)

# Calculate Work -----
```

```

extension <- read.csv("Table S2.csv")
colnames(extension) <- c("Species", "Specimen", "SL", "Mass", "DA", "Trial", "Surface",
                        "Peak_Force", "Load", "Time", "Extension", "P1", "P2")

AllData$Work <- NA
for (i in unique(extension$Specimen)) {
  specimen <- AllData[which(AllData$Specimen == i),]
  for (y in specimen$Trial) {
    tmp.data <- extension %>% dplyr::filter(Specimen == i & Trial == y)
    tmp.data$Load <- as.numeric(tmp.data$Load)
    tmp.data$Extension <- as.numeric(tmp.data$Extension) * 0.001
    # convert extension from mm to m

    # remove all the points prior to the zero extension point and calibrate time
    tmp.data <- tmp.data %>% dplyr::filter(Load >= mean(as.numeric(tmp.data$Load[1:15]))+0.1)
    tmp.data$Time <- as.numeric(tmp.data$Time) - min(as.numeric(tmp.data$Time))
    tmp.data$Extension <- as.numeric(tmp.data$Extension) - min(as.numeric(tmp.data$Extension))

    # find when/what extension = peak load
    max.point <- which(specimen[specimen$Trial == y,"Force"] == tmp.data$Load)
    max.point <- max.point[length(max.point)]

    dropoff <- max.point
    for (t in max.point:nrow(tmp.data)){
      if (tmp.data$Load[dropoff]-tmp.data$Load[t] > tmp.data$Load[dropoff]*0.5) {
        dropoff = t
      }
    }

    #calculate stress and strain
    tmp.data$Stress <- tmp.data$Load/tmp.data$DA*10
    tmp.data$Strain <- tmp.data$Extension/tmp.data$Extension[length(tmp.data$Extension)]

    start <- unique(tmp.data$P1)
    stop <- unique(tmp.data$P2)
    stiff <- ((tmp.data$Stress[stop] - tmp.data$Stress[start])/(tmp.data$Strain[stop] - tmp.data$Strain[start]))

    # keep only part of the curve until peak load
    tmp.data <- tmp.data[1:max.point,]
    # work under the whole curve
    work <- sum((tmp.data$Load[2:length(tmp.data$Load)] + tmp.data$Load[-length(tmp.data$Load)])/2 *
              (tmp.data$Extension[2:length(tmp.data$Extension)] - tmp.data$Extension[-
length(tmp.data$Extension)]))

    AllData[which(AllData$Specimen == i & AllData$Trial == y), "Work"] <- as.numeric(work)
    AllData[which(AllData$Specimen == i & AllData$Trial == y), "Stiffness"] <- as.numeric(stiff)
  }
}

```

```

}
}
AllData$Work <- AllData$Work * 1000 # convert J to mJ

# LMER -----

# perform linear mixed effect models
# calculate point estimates and R^2
AllData$Surface <- as.factor(AllData$Surface)
Force.lme <- lmer(Force ~ Species * Surface + (1|Specimen) , data = AllData)
emmeans(Force.lme, ~ Surface | Species)
r2_nakagawa(Force.lme)

x_BW.lme <- lmer(x_BW ~ Species * Surface + (1|Specimen) , data = AllData)
emmeans(x_BW.lme, ~ Surface | Species)
r2_nakagawa(x_BW.lme)

Stress.lme <- lmer(Stress ~ Species * Surface + (1|Specimen) , data = AllData)
emmeans(Stress.lme, ~ Surface | Species)
r2_nakagawa(Stress.lme)

Work.lme <- lmer(Work ~ Species * Surface + (1|Specimen) , data = AllData)
emmeans(Work.lme, ~ Surface | Species)
r2_nakagawa(Work.lme)

# Plot Fig 2 -----

# code for plotting Figure 2
a <- ggplot(AllData, aes(x = as.factor(Surface), y=Stress, fill = Species)) +geom_boxplot(lwd=0.3,outlier.size =
1, outlier.shape = 21, color = "black", show.legend = T) +
  labs( x = bquote('Surface Roughness'~(mu*m)), y = "Stress (kPa)") +
  scale_fill_manual(values = col[1:3], labels = c("LD" = "Snailfish", "EO" = "Lumpsucker", "GM" =
"Clingfish"))+
  theme_classic()+
  theme(legend.position="bottom")
b <- ggplot(AllData, aes(x = as.factor(Surface), y=Force, fill = Species)) +geom_boxplot(lwd=0.3,outlier.size =
1, outlier.shape = 21, color = "black", show.legend = FALSE) +
  labs( x = bquote('Surface Roughness'~(mu*m)), y = "Adhesive Force (N)") +
  scale_fill_manual(values = col[1:3], labels = c("LD" = "Snailfish", "EO" = "Lumpsucker", "GM" =
"Clingfish"))+
  theme_classic()+
  theme(legend.position="none")
c <- ggplot(AllData, aes(x = as.factor(Surface), y=((Work)), fill = Species))
+geom_boxplot(lwd=0.3,outlier.size = 1, outlier.shape = 21, color = "black", show.legend = FALSE) +
  labs( x = bquote('Surface Roughness'~(mu*m)), y = "Work to Detach (mJ)") +

```

```

scale_fill_manual(values = col[1:3], labels = c("LD" = "Snailfish", "EO" = "Lumpsucker", "GM" =
"Clingfish"))+
theme_classic()+
theme(legend.position="none")
d <- ggplot(AllData, aes(x = as.factor(Surface), y=Force/(Mass/1000*9.8), fill = Species))
+geom_boxplot(lwd=0.3,outlier.size = 1, outlier.shape = 21, color = "black", show.legend = FALSE) +
labs( x = bquote('Surface Roughness'~(mu*m)), y = "Force x BW") +
scale_fill_manual(values = col[1:3], labels = c("LD" = "Snailfish", "EO" = "Lumpsucker", "GM" =
"Clingfish"))+
theme_classic()+
theme(legend.position="none")

b + c + d + a + plot_annotation(tag_levels = "A") + plot_layout(guides = 'collect') & theme(legend.position =
'bottom')

```

Plot Fig 3 -----

code for plotting Figure 3

```

scaleFUN <- function(x) sprintf("%.0f", x)
a <- ggplot(AllData, aes(x = Mass, y = DA, color = Species)) +geom_point(pch = 1, size = 1.7) +
labs( x = "Body Mass (g)", y = bquote('Disc Area'~(cm^2))) +
scale_colour_manual(values = col[1:3],labels = c("LD" = "Snailfish", "EO" = "Lumpsucker", "GM" =
"Clingfish"))+
scale_x_continuous(trans = "log", labels = scaleFUN) +
scale_y_continuous(trans = "log", labels = scaleFUN) +
theme_classic()+
geom_smooth(method=lm, se=F, formula = y ~ x) +
theme()
b <- ggplot(AllData, aes(x = Mass, y = Force, color = Species)) +geom_point(pch = 1, size = 1.7) +
labs( x = "Body Mass (g)", y = bquote('Adhesive Force (N)')) +
scale_colour_manual(values = col[1:3],labels = c("LD" = "Snailfish", "EO" = "Lumpsucker", "GM" =
"Clingfish"))+
scale_x_continuous(trans = "log", labels = scaleFUN) +
scale_y_continuous(trans = "log", labels = scaleFUN, lim = c(1,60)) +
theme_classic()+
geom_smooth(method=lm, se=F, formula = y ~ x) +
theme(legend.position = "none")
c <- ggplot(AllData, aes(x = DA, y = Force, color = Species)) +geom_point(pch = 1, size = 1.7) +
labs( x = bquote('Disc Area'~(cm^2)), y = "Adhesive Force (N)") +
geom_smooth(method=lm, se=F, formula = y ~ x) +
scale_colour_manual(values = col[1:3],labels = c("LD" = "Snailfish", "EO" = "Lumpsucker", "GM" =
"Clingfish"))+
scale_x_continuous(trans = "log", labels = scaleFUN) +
scale_y_continuous(trans = "log", labels = scaleFUN) +
theme_classic()+

```

```

theme(legend.position = "none")
d <- ggplot(AllData, aes(x = Mass, y = Work, color = Species)) + geom_point(pch = 1, size = 1.7) +
  labs( x = "Body Mass (g)", y = "Work to Detach (mJ)") +
  geom_smooth(method=lm, se=F, formula = y ~ x) +
  scale_colour_manual(values = col[1:3], labels = c("LD" = "Snailfish", "EO" = "Lumpsucker", "GM" =
"Clingfish"))+
  scale_x_continuous(trans = "log", labels = scaleFUN) +
  scale_y_continuous(trans = "log", labels = scaleFUN) +
  theme_classic()+
  theme(legend.position = "none")
b + c + d + a + plot_layout(guides = 'collect') + plot_annotation(tag_levels = "A") & theme(legend.position =
'bottom')

```

Scaling Analyses -----

```

# Perform scaling analyses
Allometry <- data.frame()
for (i in c("GM", "EO", "LD")) {
  data <- AllData %>% filter(Species == i & Surface != "0")
  sma <- lmodel2(log10(Force)~log10(as.numeric(Mass)), data = data %>% filter(Surface != "0" ))
  iso <- "I"
  if (isTRUE(sma$confidence.intervals[3,4] > 2/3 & sma$confidence.intervals[3,5] > 2/3) == TRUE) {
    iso <- "P"
  }
  if (isTRUE(sma$confidence.intervals[3,4] < 2/3 & sma$confidence.intervals[3,5] < 2/3) == TRUE) {
    iso <- "N"
  }
  Allometry <- rbind(Allometry, c("Force vs Mass", i,
    round(sma$rsquare,3),
    round(sma$regression.results[3,2],3),
    paste0(round(sma$confidence.intervals[3,2],3),",",
    round(sma$confidence.intervals[3,3],3)),
    round(sma$regression.results[3,3],3),
    paste0(round(sma$confidence.intervals[3,4],3),",",
    round(sma$confidence.intervals[3,5],3)),
    "0.66",
    iso))
  sma <- lmodel2(log10(Force)~log10(as.numeric(DA)), data = data %>% filter(Surface != "0" ))
  iso <- "I"
  if (isTRUE(sma$confidence.intervals[3,4] > 2/2 & sma$confidence.intervals[3,5] > 2/2) == TRUE) {
    iso <- "P"
  }
  if (isTRUE(sma$confidence.intervals[3,4] < 2/2 & sma$confidence.intervals[3,5] < 2/2) == TRUE) {
    iso <- "N"
  }
}

```



```

Allometry <- rbind(Allometry, c("Force vs DA", i,
                                round(sma$rsquare,3),
                                round(sma$regression.results[3,2],3),
                                paste0(round(sma$confidence.intervals[3,2],3),"
",round(sma$confidence.intervals[3,3],3)),
                                round(sma$regression.results[3,3],3),
                                paste0(round(sma$confidence.intervals[3,4],3),"
",round(sma$confidence.intervals[3,5],3)),
                                "1",
                                iso))
sma <- lmodel2(log10(Work)~log10(as.numeric(Mass)), data = data %>% filter(Surface != "0" ))
iso <- "I"
if (isTRUE(sma$confidence.intervals[3,4] > 3/3 & sma$confidence.intervals[3,5] > 3/3) == TRUE) {
  iso <- "P"
}
if (isTRUE(sma$confidence.intervals[3,4] < 3/3 & sma$confidence.intervals[3,5] < 3/3) == TRUE) {
  iso <- "N"
}
Allometry <- rbind(Allometry, c("Work vs Mass", i,
                                round(sma$rsquare,3),
                                round(sma$regression.results[3,2],3),
                                paste0(round(sma$confidence.intervals[3,2],3),"
",round(sma$confidence.intervals[3,3],3)),
                                round(sma$regression.results[3,3],3),
                                paste0(round(sma$confidence.intervals[3,4],3),"
",round(sma$confidence.intervals[3,5],3)),
                                "1",
                                iso))
data <- AllData %>% filter(Species == i & Surface != "0")
data <- data[which(!duplicated(data$Specimen)),]
sma <- lmodel2(log10(DA)~log10(Mass), data = data)
iso <- "I"
if (isTRUE(sma$confidence.intervals[3,4] > 2/3 & sma$confidence.intervals[3,5] > 2/3) == TRUE) {
  iso <- "P"
}
if (isTRUE(sma$confidence.intervals[3,4] < 2/3 & sma$confidence.intervals[3,5] < 2/3) == TRUE) {
  iso <- "N"
}
Allometry <- rbind(Allometry, c("DA vs Mass", i,
                                round(sma$rsquare,3),
                                round(sma$regression.results[3,2],3),
                                paste0(round(sma$confidence.intervals[3,2],3),"
",round(sma$confidence.intervals[3,3],3)),
                                round(sma$regression.results[3,3],3),
                                paste0(round(sma$confidence.intervals[3,4],3),"
",round(sma$confidence.intervals[3,5],3)),
                                "1",
                                iso))

```

```

        "0.66",
        iso))
    }
colnames(Allometry) <- c( "Variable", "Species", "R2", "Intercept", "Intercept CI", "Slope", "Slope CI",
,"Expected Slope", "Allometry")
Allometry <- Allometry[c(5,9,1,6,10,2,7,11,3,8,12,4),c(1,2,3,4,5,6,7,8,9)]
Allometry #print results

# CT Data -----

# Read in measurements from CT scans
CTData <- read.csv("Table S3.csv")
colnames(CTData) <- c("Species", "Number", "MorphoSource", "SL", "GirdleLength", "GirdleWidth",
"GirdleHeight", "MeanFinLength", "MeanContraFinDist", "MeanFinSpacing")

# size-correct morphodata using girdle length
CTData[,4:10] <- log(CTData[,4:10])
for (i in c(6:10)) {
  lm <- lm(CTData[,i]~CTData[,5])
  CTData[,i] <- lm$residuals
}

# perform LDA and plot it
lda <- lda(Species~., data = CTData[,c(1,6:10)])
lda
plot(predict(lda)$x[,1],predict(lda)$x[,2], col = col[CTData$Species], pch = 19,
      xlab = "DF1 (95% of trace)", ylab = "DF2 (5% of trace)")
legend("topright", col = col, c("Lumpsucker", "Snailfish", "Clingfish"), pch = 19, bty = "n")

lda.arrows <- function(x, myscale = 1, tex = 0.75, choices = c(1,2), ...){
  ## adds `biplot` arrows to an lda using the discriminant function values
  heads <- coef(x)
  arrows(x0 = 0, y0 = 0,
        x1 = myscale * heads[,choices[1]],
        y1 = myscale * heads[,choices[2]], length = .1,...)
  text(myscale * heads[,choices], labels = row.names(heads),
       cex = tex, pos = c(2,4,3,4,1))
}

lda.arrows(lda, myscale = 0.1)

# Stiffness -----

ggplot(AllData %>% filter(Species != "GM"), aes(x = as.factor(Species), y=Stiffness, fill = Species))+
  geom_boxplot(outlier.alpha = 0, show.legend = F)+
  geom_point(shape = 21,size = 2,aes(fill = Species, bg = "black"),

```

```
position = position_jitter(0.25), alpha = 1, show.legend = F)+  
labs( x = bquote('Species'), y = "Stiffness (kPa)" ) +  
scale_fill_manual(values = col[1:2], labels = c("LD" = "Snailfish", "EO" = "Lumpsucker"))+  
scale_x_discrete(labels = c("LD" = "Snailfish", "EO" = "Lumpsucker" ) )+  
theme_classic()+  
theme(legend.position="none")
```