
Supplementary File S4

Title: A Global Meta-Analysis Reveals the Toxicity of Plastics on Insect Health

Muzamil Abbas[1], Muhammad Jafir[1], Talha Nazir[2], Shan Hussain[3], Nadia Sarwar[1], Liyan Song[1] and Xia Wan[1]*

Correspondence to: wanxia@ahu.edu.cn

[1]*Department of Ecology, School of Resources and Environmental Engineering, Anhui University, Hefei, 230601, Anhui, China*

[2] *Centre for Agriculture and Bioscience International (CABI), Rawalpindi 46000, Punjab, Pakistan*

[3]*Department of Entomology, Faculty of Agriculture, University of Agriculture, Faisalabad, Pakistan*

```{r Environment Setup, message = FALSE, warning = FALSE}
library(metafor)

library(MuMIn)
library(tidyverse)
library(multcomp)
library(emmeans)
library(ggeffects)
library(forcats)
library(stringr)
library(clubSandwich)
library(cowplot)

##Make ggplot theme to use throughout
theme_JR <- function (base_size = 12, base_family = "Times New Roman")
```

```
  {
  theme(
    panel.background = element_rect(fill = NA),
    panel.grid = element_blank(),
    panel.border = element_rect(color = "black", fill = NA),
    axis.line = element_line(color = "black"),
    legend.title = element_text(size = 18, color = "black", family =
base_family),
    legend.text = element_text(size = 16, color = "black", family =
base_family),
    legend.key = element_rect(fill = NA, color = NA),
    axis.text = element_text(size = 12, color = "black", family =
base_family),
    axis.title = element_text(size = 16, color = "black", family =
base_family),
    strip.background = element_rect(fill = NA, color = "black")
  )
}

```

```{r Read and clean data, echo = FALSE}
##Read in dataset
fulldat2 <- read_excel(file.choose(), sheet = 1, col_names = T)

```

Grand Mean effect of MPs & NPs on Insects

Analysis of grand mean indicates that overall, Plastic will effcet on insect.

```{r, Overall model}
mgrand <- rma.mv(SMDHyi, SMDHvi,
             random = list(~1|id2, ~1|Study),
             data = fulldat2)

mgrandro <- robust(mgrand, cluster=id2, clubSandwich=TRUE)
mgrandro

##I2 (heterogeneity statistic) calculation
W <- diag(1/mgrandro$vi)
X <- mgrandro$X
P <- W - W %*% X %*% solve(t(X) %*% W %*% X) %*% t(X) %*% W
100 * sum(mgrandro$sigma2) / (sum(mgrandro$sigma2) + (mgrandro$k-
mgrandro$p)/sum(diag(P)))

##I2 = 0.7414218

##51.49929 is due to between study variation; 22.64289 is due to within
study variation
100 * mgrand$sigma2 / (sum(mgrand$sigma2) + (mgrand$k-
mgrand$p)/sum(diag(P)))
```

```{r}
Hog <- rma.mv(SMDHyi, SMDHvic,
                mods = ~  index.effect - 1,
                random = list(~1|id2, ~1|Study),
                data = meta1)

HogE <- robust(Hog, cluster=id2, clubSandwich=TRUE)
HogE
```

Funnel plot to investigate publication biases in dataset.

```{r Funnel plot, echo = F}

## Funnel plot to show potential for bias and heterogeneity
# overallmodel
# estimate = 0.1078
# se = 0.0336

# GCD specific effects
estimate = c(-1.1869, -0.3885, -0.5496, -0.8562, -1.2183, 0.3081, -
1.7780)
se = c(0.3280, 0.2971, 0.2086, 0.2306, 0.3024, 0.2240, 0.6707)

# Assuming meta1 is already defined in the workspace
Maxobsse = sqrt(max(meta1$SMDHvic , na.rm = TRUE)) + 1

# Compute vectors of the lower-limit and upper-limit values for the 95%
CI region
ll95 = estimate - (1.96 * Maxobsse)
ul95 = estimate + (1.96 * Maxobsse)

# Put all calculated values into one data frame
dfCI = data.frame(
  x = c(ll95, ul95, estimate),
  y = c(rep(Maxobsse, times = 14), rep(0, times = 7)),
  index.effect2 = rep(c("Behavioral", "Development", "Fecundity",
"Feeding", "Growth", "Health", "Survival"), times = 3)
)
dfCI$index.effect2 = factor(dfCI$index.effect2)
dfCI$index.effect2 = factor(
  dfCI$index.effect2,
  levels = levels(dfCI$index.effect2)[c(1, 2, 3, 4, 5, 6, 7)]
)

dfEST = data.frame(
  Maxobsse = Maxobsse,
  estimate = estimate,
  index.effect2 = c("Behavioral", "Development", "Fecundity", "Feeding",
"Growth", "Health", "Survival")
)
dfEST$index.effect2 = factor(dfEST$index.effect2)
dfEST$index.effect2 = factor(
```

```
  dfEST$index.effect2,
  levels = levels(dfEST$index.effect2)[c(1, 2, 3, 4, 5, 6, 7)]
)

KBF <- c(
  Behavioral = "Behavioral Response",
  Development = "Development",
  Fecundity = "Fecundity",
  Feeding = "Feeding",
  Growth = "Growth",
  Health = "Health",
  Survival = "Survival Rate"
)

meta1$index.effect2 = factor(meta1$index.effect)
meta1$index.effect2 = factor(meta1$index.effect2,
                             levels = levels(meta1$index.effect2)[c(1, 2,
3, 4, 5, 6, 7)]
)

EDF5B <- ggplot(meta1, aes(x = SMDHyi, y = sqrt(SMDHvic))) +
  facet_grid(~index.effect2, labeller = labeller(index.effect2 = GCD)) +
  scale_y_reverse() +
  geom_polygon(
    data = dfCI, aes(x = x, y = y),
    fill = "white", color = "black", size = 1,
    linetype = "dashed"
  ) +
  geom_segment(
    data = dfEST, aes(x = estimate, y = 0, xend = estimate, yend =
Maxobsse),
    size = 1, linetype = "dashed"
  ) +
  geom_point(aes(color = index.effect2)) +
  scale_color_manual(values = c(
    "#5E976E", "#58355E", "#FFCA3A", "#EC0B43",
    "#63ADF2", "#df7838", "#bf6568"
  )) +
  coord_cartesian(ylim = c(Maxobsse - 1, 0)) +
  ylab("Standard Error") +
  xlab("Hedge's G") +
   theme_JR()+
  theme(legend.position = "none")

EDF5B


ggsave("./FunnelPlotKBF.tiff",
       dpi = 600,
       width = 12,
       height = 6,
       units = "in")

```

Conducting Egger's test on the funnel plots shown in your graph reveals important insights into potential small-study effects and funnel plot asymmetry. Egger's test, which regresses effect sizes (Hedges' G) on their standard errors (SE), identifies funnel asymmetry when the intercept (beta0) significantly differs from zero. This asymmetry indicates small-study effects, though it does not directly confirm publication bias. In the context of your data, the variance (beta1) is significant for the categories of Development, Fecundity, and Health, indicating that smaller studies in these areas exhibit larger or more variable effect sizes compared to larger studies. This suggests the presence of small-study effects in these outcome categories. Additionally, the intercept (beta0) is significantly different from zero for Development, Feeding, Growth, and Health, pointing to the presence of funnel plot asymmetry in these categories, with Growth displaying the most pronounced asymmetry. This significant intercept implies that the overall effect sizes may be downwardly biased in these areas. However, for Behavioral Response and Survival Rate, neither the variance nor the intercept is significantly different from zero, indicating no evidence of funnel plot asymmetry or small-study effects. In these cases, the estimates of effect sizes are likely unbiased, and the intercept provides a reliable estimate of the adjusted mean. Therefore, while categories such as Development, Feeding, and Growth show evidence of asymmetry and potential bias, Behavioral Response and Survival Rate present more robust and reliable results with no signs of small-study effects or publication bias.

```{r Eggers Test}

###Generate effective sample  size; if no sample size for control and treatment exist, assume equal sample size and multiply 1/sample_size by 4 (is equal to 1/n1 + 1/n2, when n1 = n2)

meta1$Sample_size = ifelse(is.na(meta1$Sample_size),
                           rowSums(meta1[,c("Tn",
                                            "Cn")],
                                   na.rm = T),
                           meta1$Sample_size)

meta12 <- meta1 %>%
  mutate(Sample_sizeEff = ifelse(Cn == 0 | is.na(Cn),
                                 (1 / Sample_size) * 4,
                                 (1 / Cn) + (1/ Tn)),
         Sample_sizeEff = ifelse(is.infinite(Sample_sizeEff),
                                 NA,
                                 Sample_sizeEff),
         Sample_sizeEffsqrt = sqrt(Sample_sizeEff),
         Year.c = as.vector(scale(year, scale = F))
         )



###Publication bias analysis
```

```
##Egger's test

mpbias2n <- rma.mv(SMDHyi, SMDHvic, mods = ~ Sample_sizeEffsqrt *
index.effect + Year.c - 1,
                 random = list(~1|id2, ~1|Study),
                 data = meta12)

mpbias2nro <- robust(mpbias2n, cluster=id2, clubSandwich=TRUE)
mpbias2nro

##note that, per Nakagawa 2022, if slope of sqrt(effective sample size)
is significant,
##use effective sample size (it is significant, so use model below)

mpbias2n1 <- rma.mv(SMDHyi, SMDHvic, mods = ~ Sample_sizeEff *
index.effect + Year.c - 1,
                 random = list(~1|id2, ~1|Study),
                 data = meta12)


mpbias2nro1 <- robust(mpbias2n1, cluster=id2, clubSandwich=TRUE)
mpbias2nro1
##time is non-significant

mpbias2nem1 <- qdrg(object = mpbias2nro1, data = meta12, at =
list(Sample_sizeEff = 0, Year.c = 0 ))

##Significant beta0
emmmn <- emmeans(mpbias2nem1,  ~ index.effect, nesting = NULL)
test(emmmn)

#Sample_sizeEff
##Significant positive beta1
mx <- mean(meta12$Sample_sizeEff, na.rm = T)
rg <- qdrg(object = mpbias2nro1, data = meta12, nesting = NULL, at =
list(Sample_sizeEff = mx + c(0, 1)))
emt <- update(contrast(rg, "consec", simple = "Sample_sizeEff"), by =
NULL)
emt

#Plot
#Get trend slines from Egger's test and time regression
eggerstestlines <- data.frame(index.effect2 =
factor(unique(meta1$index.effect)),
           intercept = summary(emmmn)$emmean[c(7,6,5,4,3,2,1)],
           slope = summary(emt)$estimate[c(7,6,5,4,3,2,1)])

eggerstestlines$index.effect = factor(eggerstestlines$index.effect,
                                                 levels =
levels(eggerstestlines$index.effect)[c(1,2,3,4,5,6,7)])

EDF5C <- ggplot(meta12, aes(y = SMDHyi, x = Sample_sizeEff)) +
```

```
  facet_grid(~index.effect, labeller = labeller(index.effect = KBF)) +  #
Use 'index.effect' if no 'index.effect2'
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_point(aes(color = index.effect)) +  # Change 'index.effect2' to
'index.effect' if necessary
  scale_color_manual(values = c("#5E976E", "#58355E",
                                "#FFCA3A", "#EC0B43",
                                "#63ADF2", "#df7838",
                                "#bf6568")) +
  geom_segment(data = eggerstestlines, aes(x = 0, xend = 2, y =
intercept, yend = intercept + slope*10),
               size = 1, color = "black") +
  ylab("Hedge's G") +
  xlab("Inverse of Effective Sample Size") +
  theme_JR() +
  theme(legend.position = "none")
EDF5C

ggsave("./EggersTestKBF.tiff",
       dpi = 600,
       width = 11,
       height = 7,
       units = "in")
```

Conduct Fail-safe N analyses for each outcome, which returns the "File Drawer Number," representing the number of statistically non-significant unpublished results needed to make the overall effect non-significant. Apply the Rosenthal, Orwin, and Rosenberg methods, setting the "target" for Orwin's method to 0.1 for all outcomes. Feeding and Growth may be more vulnerable to bias, with Orwinâ€™s method potentially showing values below the threshold (5*Nstudies + 10). However, for Behavioral Response, Development, Health, and Survival Rate, all fail-safe N values, particularly from Rosenthal and Rosenberg, are expected to far exceed the threshold, suggesting robustness against publication bias. Thus, even though some asymmetry is present, it is unlikely that enough unpublished null results exist to negate the observed effects.

```{r Fail-safe N}

# Filter the data
subdat <- meta1 %>%
  filter(!is.na(SMDHyi),
         !is.na(SMDHvic))

# Split the data by 'index.effect'
subdat <- split(subdat, subdat$index.effect)

# Initialize a dataframe for results
hldn <- data.frame(KBF = as.character(paste(rep(0, times = 7))),
```

```
                    Rosenthal = rep(0, times = 7),
                    Orwin = rep(0, times = 7),
                    Rosenberg = rep(0, times = 7),
                    ThreasholdLargeStudyN = rep(0, times = 7))

# Loop through each subset of data
for(i in 1:length(subdat)) {
  # Store the current effect name
  hldn$KBF[i] = names(subdat)[i]

  # Fit an 'rma' model for each subset of data
  rma_model <- rma(yi = subdat[[i]]$SMDHyi, vi = subdat[[i]]$SMDHvic)

  # Calculate the fail-safe numbers
  hldn$Rosenthal[i] = fsn(rma_model, type = "Rosenthal")$fsnum
  hldn$Orwin[i] = fsn(rma_model, type = "Orwin", target = 0.1)$fsnum
  hldn$Rosenberg[i] = fsn(rma_model, type = "Rosenberg")$fsnum

  # Calculate the threshold for large study N
  hldn$ThreasholdLargeStudyN[i] = 5 * length(unique(subdat[[i]]$id2)) +
10
}

hldn
```

```{r}
str(subdat)
```

Leave one out analysis to test for robustness of our analyses. For loop is not run due to time constraints, but code is presented. Leave one out analyses suggests that our results are robust to removal of individual studies.

```{r}
#
subdat = subgranmod = NULL
hld = data.frame(beta = 0,
                 se = 0)

 for(i in 1:length(unique(meta1$id2))) {
   subdat = subgranmod = NULL
  subdat <- meta1 %>%
     filter(id2 != unique(meta1$id2)[i])
   subgranmod <- rma.mv(SMDHyi, SMDHvic,
                        random = list(~1|id2, ~1|Study),
                        data = subdat)
  hld[i,1] = subgranmod$beta[1]
  hld[i,2] = subgranmod$se
 }
 saveRDS(hld, "./Leave1Outdat.RDS")

hld <- readRDS ("K:/Abbas/Meta-Analysis/Meta-Analysis/Leave1Outdat.RDS")
```

```
l1o <- ggplot(hld, aes(x = 1:nrow(hld), y = beta))+
  geom_errorbar(aes(ymin = beta - se, ymax = beta + se), width = 0)+
  geom_point(color = "light grey", size = 2)+
  geom_hline(yintercept = -1.428, color = "#EA2B1F", linewidth = 1)+
  geom_hline(yintercept = -1.428 + 0.474, color = "#EA2B1F", linewidth =
1)+
  geom_hline(yintercept = -1.428 - 0.474, color = "#EA2B1F", linewidth =
1)+
  scale_x_continuous(limits = c(0,83), expand = c(0,0))+
  labs(x = "Study",
       y = "Hedge's G")+
  theme_JR()+
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank())
l1o
ggsave("./Leave1out.tiff",
       dpi = 300,
       width = 5,
       height = 4.65,
       units = "in")
```

```

Forest plots to show distribution of effect sizes and variances among key biological factor.

```{r}
data2 = meta1 %>%
  filter(! is.na(SMDHyi)) %>%
  filter(! is.na(SMDHvic))

##Weights as percentages
data2$weights = rowSums(weights(mgrandro, type = "matrix"))/100
data2 = data2[order(data2$SMDHyi),]

data2$x = seq(1,nrow(data2)*2,2)
set.seed(20)
data2$x2 = sample(data2$x, nrow(data2))

data2$index.effect = as.factor(data2$index.effect)
data2$index.effect = factor(data2$index.effect,
                        levels(data2$index.effect)[c(1,2,3,4,5,6,7)])

EDF5A <- ggplot(data2, aes(y = SMDHyi, x = x2, color = index.effect))+
  # scale_y_continuous(limits = c(-7.5,7.5),
  #                    expand = c(0,0))+
```
```

```
  facet_grid(~index.effect, labeller = labeller(index.effect = GCD))+
  scale_x_reverse()+
  geom_hline(yintercept = 0, linetype = "dashed")+
  geom_errorbar(aes(x = x2,
                    ymax = SMDHyi + sqrt(SMDHvic),
                    ymin = SMDHyi - sqrt(SMDHvic)),
                width = 0)+
  geom_point(aes(size = weights), shape = 21)+
  # geom_errorbar(aes(x = max(x), y = 0.1875,
  #                   ymin = 0.095,
  #                   ymax = 0.28), width = 10)+
  # geom_point(aes(x = max(x), y = 0.1875), shape = 23, fill = NA, size =
4)+
  scale_color_manual(values = c("#5E976E", "#58355E",
                                "#FFCA3A", "#EC0B43",
                                "#63ADF2", "#df7838",
                                "#bf6568")) +
  # coord_flip(ylim = c(-10,10))+
  ylab("Hedge's G")+
  coord_flip()+
  theme_JR()+
  theme(axis.text.y = element_blank(),
        axis.title.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.line.y = element_blank(),
        legend.position = "none")
EDF5A
ggsave("./ForestPlot.tiff",
       dpi = 600,
        height = 5.5,
        width = 7.5,
        units = "in")

```

```{r}
#Generate Extended data figure 5
EDF5_WHOLE = cowplot::align_plots(EDF5A,
                                  EDF5B,
                                  EDF5C,
                                  l1o,
                                  align = 'hv', axis = 'l')


EDF5_fig = cowplot::plot_grid(EDF5_WHOLE[[1]],
                                 EDF5_WHOLE[[2]],
                                 EDF5_WHOLE[[3]],
                               EDF5_WHOLE[[4]],
                              labels = c("A)","B)","C)","D)"),
                              ncol = 1,
                              label_x = 0,
                              label_y = 0.975)
EDF5_fig
ggsave("./EDF5.tiff",
```

```
       dpi = 600,
       height = 11.3,
       width = 12,
       units = "in")

```

Analyses

Step 1: Test for effects on key biological factors

```{r General Effects of GCDs, warning = FALSE}

options(contrasts = c("contr.treatment","contr.poly"))

m2 <- rma.mv(SMDHyi, SMDHvic,
             mods = ~  index.effect - 1,
             random = list(~1|id2, ~1|Study),
             data = meta1)

m2ro <- robust(m2, cluster=id2, clubSandwich=TRUE)
m2ro

##I2 (heterogeneity statistic) calculation
W <- diag(1/m2ro$vi)
X <- m2ro$X
P <- W - W %*% X %*% solve(t(X) %*% W %*% X) %*% t(X) %*% W
100 * sum(m2ro$sigma2) / (sum(m2ro$sigma2) + (m2ro$k-
m2ro$p)/sum(diag(P)))

##I2 = 98.26079

##33.68871 is due to between study variation; 64.57208 is due to within
study variation
100 * m2ro$sigma2 / (sum(m2ro$sigma2) + (m2ro$k-m2ro$p)/sum(diag(P)))

##Set up emmeans reference grid
m2g <- qdrg(object = m2ro, data = meta1)


m2em <- emmeans(m2g, ~ index.effect)
emmeans:::cld.emmGrid(m2em)

```

```{r Figure for Main KBF, echo = FALSE}

figdat_KBF <- data.frame(m2em)
figdat_KBF$grouping = c("*A", "*A", "*A", "*B", "*A", "*AB", "*A")
figdat_KBF$EffLab = c(7,6,5,4,3,2,1)

#create a new column for KBF labels used in figure
figdat_KBF$index.effect2 <- c("Behaviour",
                                        "Development",
```

```
                                        "Fecundity",
                                        "Feeding",
                                        "Growth",
                                        "Health",
                                        "Survival")


figdat_KBF$k = (meta1 %>%
  group_by(index.effect) %>%
  summarize(count = n()))$count

figdat_KBF$n = (meta1 %>%
  group_by(index.effect) %>%
  summarize(count = length(unique(id2))))$count


figdat_KBF$KBF_xlab <- paste(figdat_KBF$index.effect2, "\n(n = ",
                             figdat_KBF$n, ", k = ",
                             figdat_KBF$k, ")", sep = "")

#make the plot
ggplot(figdat_KBF, aes(x = KBF_xlab,
                       y = emmean,
                       color = index.effect2)) +
    geom_hline(yintercept = 0,                   #make a reference line
at 0
             linetype = "dashed",
             color = "black",
             linewidth = 1) +
  geom_point(size = 3) +                                 #plot points
  geom_errorbar(aes(ymin = lower.CL,        #include error bars
                    ymax = upper.CL),
                width=0.25,
                size = 1) +                    #adjust width of bar
  geom_text(aes(label = grouping,
                x = EffLab),
            color = "black",
            position = position_nudge(x = 0.32,y= -0.04))+
  scale_y_continuous(limits = c(-12.5,2),      #change limits of Y axis
                     breaks = seq(-12.5,1.5,2))+      #set Y axis breaks
  scale_color_manual(values = c("#5E976E","#cad76c",
                                "#976543","#58355E",
                                "#FFCA3A","#EC0B43",
                                "#63ADF2"))+
  xlab("Key biological Factors") +               #relabel X and Y axes
  ylab("Hedge's G") +
  scale_x_discrete(limits = rev(levels(as.factor(figdat_KBF$KBF_xlab))))+
  coord_flip()+
  theme_JR()+                                        #call your theme
  theme(legend.position = "none",
        axis.title.y = element_blank())

ggsave("./KBFMetaFig2.tiff",
       dpi = 800,
```

```
       width = 5.85,
       height = 4.5,
       units = "in")

```

Step 2: Test for Subfactors: i.e., subgroupings of ALL KBF

```{r Subfactor analyses}

variable_counts <- meta1 %>%
  group_by(End.Point) %>%
  tally(name = "count")

# Filter out the variables with fewer than 10 occurrences
filtered_data <- meta1 %>%
  semi_join(variable_counts %>% filter(count >= 10), by = "End.Point")

sort(unique(filtered_data$End.Point))
sort(unique(filtered_data$index.effect))



mKBFactor_sub <- rma.mv(SMDHyi, SMDHvic,
             mods = ~  End.Point - 1,
             random = list(~1|id2, ~1|Study),
             data = filtered_data)

mKBFactor_subro <- robust(mKBFactor_sub, cluster=id2, clubSandwich=TRUE)
mKBFactor_subro

##I2 (heterogeneity statistic) calculation
W <- diag(1/mKBFactor_subro$vi)
X <- mKBFactor_subro$X
P <- W - W %*% X %*% solve(t(X) %*% W %*% X) %*% t(X) %*% W
100 * sum(mKBFactor_subro$sigma2) / (sum(mKBFactor_subro$sigma2) +
(mKBFactor_subro$k-mKBFactor_subro$p)/sum(diag(P)))

##I2 = 99.86

##10.60 is due to between study variation; 89.26 is due to within study
variation
100 * mKBFactor_subro$sigma2 / (sum(mKBFactor_subro$sigma2) +
(mKBFactor_subro$k-mKBFactor_subro$p)/sum(diag(P)))


##Set up emmeans reference grid
mKBFactor_subrog <- qdrg(object = mKBFactor_subro, data = filtered_data)


mKBFactor_subroem <- emmeans(mKBFactor_subrog, ~ End.Point)
emmeans:::cld.emmGrid(mKBFactor_subroem)

```

```{r}
#end.points
###Need to remove some variables, due to lack of replication
## Drop Fungicide & Sulfur containing
variable_counts <- meta1 %>%
  group_by(End.Point) %>%
  tally(name = "count")
```

```{r}
# Filter out the variables with fewer than 10 occurrences
filtered_data1 <- meta1 %>%
  semi_join(variable_counts %>% filter(count >= 10), by = "End.Point")

sort(unique(filtered_data1$End.Point))
sort(unique(filtered_data1$index.effect))

mKBFactor_sub <- rma.mv(LRRyi, LRRvic,
                        mods = ~  End.Point - 1,
                        random = list(~1|id2, ~1|Study),
                        data = filtered_data)

mKBFactor_subro <- robust(mKBFactor_sub, cluster=id2, clubSandwich=TRUE)
mKBFactor_subro

```

```{r Subfactor figure, echo = FALSE, warning = FALSE}
figdat_sub <- data.frame(est = mKBFactor_subro$beta,
                         ci.low = mKBFactor_subro$ci.lb,
                         ci.up = mKBFactor_subro$ci.ub,
                         KBF = row.names(mKBFactor_subro$beta))

figdat_sub$End.Point = as.factor(gsub('End.Point', '', figdat_sub$KBF))

figdat_sub$KBF =
fct_rev(filtered_data$index.effect[match(figdat_sub$End.Point,

filtered_data$End.Point)])

#
figdat_sub$KBF = fct_rev(figdat_sub$KBF)

figdat_sub$End.Point <- str_to_sentence(figdat_sub$End.Point)

figdat_sub$End.Point[1] = "Anti-Oxidant"
figdat_sub$End.Point[2] = "Body length"
figdat_sub$End.Point[3] = "Body weight"
figdat_sub$End.Point[4] = "Climbing Activity"
figdat_sub$End.Point[5] = "D-Glucose content"
figdat_sub$End.Point[6] = "Emergence ratio"
figdat_sub$End.Point[7] = "Emerging time"
figdat_sub$End.Point[8] = "Food intake"
```

```
figdat_sub$End.Point[9] = "GSH"
figdat_sub$End.Point[10] = "Locomotory activity"
figdat_sub$End.Point[11] = "MDA"
figdat_sub$End.Point[12] = "Number of eggs"
figdat_sub$End.Point[13] = "Protein Content"
figdat_sub$End.Point[14] = "ROS"
figdat_sub$End.Point[15] = "Shannon Diversity"
figdat_sub$End.Point[16] = "Simphson Diversity"
figdat_sub$End.Point[17] = "Sleep time"
figdat_sub$End.Point[18] = "SOD"
figdat_sub$End.Point[19] = "Stage Duration"
figdat_sub$End.Point[20] = "Survival"
figdat_sub$End.Point[21] = "TG Content"


figdat_sub$End.Point = as.factor(figdat_sub$End.Point)

subkn = (filtered_data %>%
           group_by(End.Point) %>%
           summarize(kcount = n(),
                     ncount = length(unique(id2))))


figdat_sub$n = subkn$ncount[c(1,2,4,5,3,6:19,21,20)]
figdat_sub$k = subkn$kcount[c(1,2,4,5,3,6:19,21,20)]

figdat_sub$KBF_xlab <- as.factor(paste(figdat_sub$End.Point, "\n(n = ",
                                       figdat_sub$n, ", k = ",
                                       figdat_sub$k, ")", sep = ""))
#


figdat_sub$KBF_xlab = factor(figdat_sub$KBF_xlab,

levels(figdat_sub$KBF_xlab)[c(2,3,4,10,17,6,7,19,8,12,20,1,5,9,11,13,14,1
8,21,15,16)])


figdat_sub$KBF = factor(figdat_sub$KBF,
                        levels(figdat_sub$KBF)[c(1,2,3,4,5,7,6)])


#make the plot
ggplot(figdat_sub, aes(x = KBF_xlab,
                       y=est,
                       color = KBF)) +
  geom_hline(yintercept = 0,                    #make a reference line at
0
             linetype = "dashed",
             color = "black",
             size = 1) +
  geom_vline(xintercept = c(2.5,10.5,11.5, 13.5,16.6,19.5),
             color = "grey",
             linetype = "dotted",
```

```
            linewidth = 1)+
  geom_point(size = 2.5) +                                  #plot points
  geom_errorbar(aes(ymin = ci.low,        #include error bars
                    ymax = ci.up),
                width=0.33,
                linewidth = 0.6) +                    #adjust width of bar
  scale_y_continuous(limits = c(-7.02,9),      #change limits of Y axis
                     breaks = seq(-6,8,2))+      #set Y axis breaks
  xlab("") +               #relabel X and Y axes
  ylab("Hedge's G") +
  scale_color_manual(values = c("#5E976E","#222081",
                                "#b64873","#FFCA3A",
                                "#58355E","#EC0B43",
                                "#63ADF2"))+
  coord_flip()+#ylim = c(-2,2))+
  scale_x_discrete(limits = rev(levels(figdat_sub$KBF_xlab)))+
  theme_JR()+                                        #call your theme
  theme(legend.position = "none",
        axis.text.y = element_text(size = 9),
        axis.title.y = element_blank())

ggsave("./KBFFig3NEWwhole.tiff",
       dpi = 300,
       width = 7,
       height = 9,
       units = "in")


```

Step 3: Test for Interactions between KBF and other factors:

Checking for generalities of the patterns

Checking for main and interactive effects between study, country,Plastic Types and insect family and the main biological Factors.

```{r Interaction Models - Generality of Findings Setup, echo = FALSE}


#Regression models that include two-way interactions between Key biological factors and each other factor
#Returns to deviations from a reference level
options(contrasts = c("contr.treatment","contr.poly"))


##Country
#
contdat <- meta1 %>%
  filter(!is.na(Country))

##Insect Species

datSpec = meta1 %>%
```

```
filter(!is.na(Species))

##Insect Families

datFamil = meta1 %>%
filter(!is.na(Family))

##Insect Order

datOrd <- meta1 %>%
filter(!is.na(Order))



##Plastic Types
datPlast <- meta1 %>%
  filter(!(Type %in% c("other")))



##Insect Sex
datSex = meta1 %>%
filter(!is.na(Sex)) %>%
filter(!str_detect(Sex, "-"))


##Insect Stage
datStag =meta1 %>%
filter(!(Stage %in% c("Pre-Pupae", "Brood", "Cocoon")))


##Plastic Concentration
datConcen = meta1 %>%
  filter(!is.na(Concentration))


###Plastic Size
datSize = meta1 %>%
  filter(!is.na(Plastic.Size))

###Polymer Type
datPoly = meta1 %>%
 filter(!is.na(Polymer.types))



###Exposure Duration
datExpos = meta1 %>%
 filter(!is.na(Exposure.duration))

```

Country

```{r}

##Number of replicates
nrow(contdat)

##Main effect of endpoint
mCountry <- rma.mv(SMDHyi, SMDHvic,
                   mods = ~ Country + index.effect,
                   random = list(~1|id2, ~1|Study),
                   data=contdat, method = "ML")

mCountryIntML <- rma.mv(SMDHyi, SMDHvic,
                   mods = ~ Country * index.effect,
                   random = list(~1|id2, ~1|Study),
                   data=contdat, method = "ML")


##different; interaction is significant
anova(mCountry, mCountryIntML)


mCountryInt <- rma.mv(SMDHyi, SMDHvic,
       mods = ~ Country * index.effect - 1,
       random = list(~1|id2, ~1|Study),
       data=contdat)

mCountryIntro <- robust(mCountryInt, cluster=id2, clubSandwich=TRUE)
mCountryIntro

##Currently, testing whether the effect of GCDs are similar across
endpoints
##Set up emmeans reference grid
mContg <- qdrg(object = mCountryIntro, data = contdat)

mContgem <- emmeans(mContg, ~ Country|index.effect)


##Differences between Europe and North America for HLC
##Differences between Europe and Asia for BC
pairs( (mContgem))

```

```{r}
##Data
datfigCont <- data.frame(mContgem)
datfigCont$index.effect = as.factor(datfigCont$index.effect)
datfigCont$index.effect = factor(datfigCont$index.effect,

levels(datfigCont$index.effect)[c(1,2,3,4,5,6,7)])

figCont <- ggplot(datfigCont, aes(x = Country,
                  y=emmean,
```

```
                color = Country)) +            #subtract 2 to adjust
range of response
  facet_grid(~index.effect, labeller = labeller(index.effect = KBF)) +
  geom_point(position = position_dodge(width = 0.2),
             size = 2.5) +                                  #plot points
  geom_errorbar(aes(ymin = lower.CL,        #include error bars
                    ymax = upper.CL),
                width=0.7,
                position = position_dodge(width = 0.2)) +
#adjust width of bar
  geom_hline(yintercept = 0,                    #make a reference line at
0
             linetype = "dashed",
             color = "black",
             size = 1) +
  xlab("Country") +               #relabel X and Y axes
  ylab("Hedge's G") +
  coord_flip()+
  theme_JR()  +
  theme(legend.position = "none")

figCont

ggsave("./KBFFigSCountry.tiff",
       dpi = 600,
       width = 12,
       height = 8,
       units = "in")

```

Insects Order

Some differences among Insects Order. Interaction model suggests only differences in responses among Insects Order are within key biologicatl triats

```{r}
##Number of replicates
nrow(datOrd)


mInsects_Order <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Order + index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datOrd, method = "ML")

mInsects_OrderIntML <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Order * index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datOrd, method = "ML")
```

```
##very different
anova(mInsects_Order, mInsects_OrderIntML)

mInsects_OrderInt <- rma.mv(SMDHyi, SMDHvic,
                      mods = ~ Order * index.effect - 1,
                      random = list(~1|id2, ~1|Study),
                      method = "REML", data=datOrd)



mInsects_OrderIntro <- robust(mInsects_OrderInt, cluster=id2,
clubSandwich=TRUE)
mInsects_OrderIntro

##Set up emmeans reference grid
mOrderg <- qdrg(object = mInsects_OrderIntro, data = datOrd)

mOrdergem <- emmeans(mOrderg, ~ Order|index.effect)

#
pairs((mOrdergem))

```

```{r Order figure, echo = FALSE}

datfig_Order <- data.frame(mOrdergem)
datfig_Order $index.effect = as.factor(datfig_Order $index.effect)
datfig_Order $index.effect = factor(datfig_Order $index.effect,
                         levels(datfig_Order
$index.effect)[c(1,3,2,4,5,6,7)])
#make the plot
Orderfig <- ggplot(datfig_Order , aes(x = Order,
                 y=emmean,
                 color = Order)) +            #subtract 2 to adjust range
of response
  facet_grid(~index.effect, labeller = labeller(index.effect = KBF)) +
  geom_point(position = position_dodge(width = 0.2),
             size = 2.5) +                                #plot points
  geom_errorbar(aes(ymin = lower.CL,        #include error bars
                    ymax = upper.CL),
                width=0.7,
                position = position_dodge(width = 0.2)) +
#adjust width of bar
  geom_hline(yintercept = 0,                   #make a reference line at
0
             linetype = "dashed",
             color = "black",
             size = 1) +
  xlab("Insect Order") +               #relabel X and Y axes
  ylab("Hedge's G") +
  coord_flip()+
  theme_JR()  +
  theme(legend.position = "none")
```

```
Orderfig

ggsave("./KBFFigS7.tiff",
       dpi = 600,
       width = 12.5,
       height = 6.65,
       units = "in")
```

Insect Species

Some differences among Insect Species. Interaction model suggests only differences in responses among Insect Species are within key biologicatl triats

```{r Speceies}


##Number of replicates
nrow(datSpec)

mInsect_Species <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Species + index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datSpec, method = "ML")



mInsect_SpeciesIntML <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Species * index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datSpec, method = "ML")


##different
anova(mInsect_Species, mInsect_SpeciesIntML)


mInsect_SpeciesInt <- rma.mv(SMDHyi, SMDHvic,
       mods = ~ Species * index.effect - 1,
       random = list(~1|id2, ~1|Study),
       data=datSpec)


mInsect_SpeciesIntro <- robust(mInsect_SpeciesInt, cluster=id2,
clubSandwich=TRUE)
mInsect_SpeciesIntro

##Currently, testing whether the effect of KBFs are similar across
endpoints
##Set up emmeans reference grid
mInsect_Speciesg <- qdrg(object = mInsect_SpeciesIntro, data = datSpec)
```

```
mInsect_Speciesgem <- emmeans(mInsect_Speciesg, ~ Species|index.effect)

##Only different in CC
pairs(mInsect_Speciesgem)

```

```{r Insect Species Figure, echo = FALSE}

datfig_Specie <- data.frame(mInsect_Speciesgem)
datfig_Specie$index.effect = as.factor(datfig_Specie$index.effect)
datfig_Specie$index.effect = factor(datfig_Specie$index.effect,

levels(datfig_Specie$index.effect)[c(1,2,3,4,5,6,7)])
#make the plot
Speciefig <- ggplot(datfig_Specie, aes(x = Species,
                 y=emmean,
                 color = Species)) +            #subtract 2 to adjust
range of response
  facet_grid(~index.effect, labeller = labeller(index.effect = KBF)) +
  geom_point(position = position_dodge(width = 0.2),
             size = 2.5) +                                #plot points
  geom_errorbar(aes(ymin = lower.CL,        #include error bars
                    ymax = upper.CL),
                width=0.7,
                position = position_dodge(width = 0.2)) +
#adjust width of bar
  geom_hline(yintercept = 0,                   #make a reference line at
0
             linetype = "dashed",
             color = "black",
             size = 1) +
  xlab("Insect Species") +               #relabel X and Y axes
  ylab("Hedge's G") +
  coord_flip()+
  theme_JR()  +
  theme(legend.position = "none")

Speciefig
ggsave("./KBFFigSHTax.tiff",
       dpi = 600,
       width = 12.65,
       height = 8.65,
       units = "in")
```

Insects Families

Some differences among Insects Families. Interaction model suggests only differences in responses among Insects Families are within key biological triats

```{r}
```

```
##Number of replicates
nrow(datFamil)


mInsect_Family <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Family + index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datFamil, method = "ML")

mInsect_FamilyIntML <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Family * index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datFamil, method = "ML")



##very different
anova(mInsect_Family, mInsect_FamilyIntML)

mInsect_FamilyInt <- rma.mv(SMDHyi, SMDHvic,
                      mods = ~ Family * index.effect - 1,
                      random = list(~1|id2, ~1|Study),
                      method = "REML", data=datFamil)



mInsect_FamilyIntro <- robust(mInsect_FamilyInt, cluster=id2,
clubSandwich=TRUE)
mInsect_FamilyIntro

##Set up emmeans reference grid
mFamilyg <- qdrg(object = mInsect_FamilyIntro, data = datFamil)

mFamilygem <- emmeans(mFamilyg, ~ Family|index.effect)

#
pairs((mFamilygem))

```

```{r Families figure, echo = FALSE}

datfig_Family <- data.frame(mFamilygem)
datfig_Family$index.effect = as.factor(datfig_Family$index.effect)
datfig_Family$index.effect = factor(datfig_Family$index.effect,

levels(datfig_Family$index.effect)[c(1,3,2,4,5,6,7)])
#make the plot
Familyfig <- ggplot(datfig_Family, aes(x = Family,
                  y=emmean,
                  color = Family)) +            #subtract 2 to adjust
range of response
  facet_grid(~index.effect, labeller = labeller(index.effect = KBF)) +
  geom_point(position = position_dodge(width = 0.2),
```

```
            size = 2.5) +                                  #plot points
  geom_errorbar(aes(ymin = lower.CL,        #include error bars
                    ymax = upper.CL),
                width=0.7,
                position = position_dodge(width = 0.2)) +
#adjust width of bar
  geom_hline(yintercept = 0,                    #make a reference line at
0
             linetype = "dashed",
             color = "black",
             size = 1) +
  xlab("Insect Families") +               #relabel X and Y axes
  ylab("Hedge's G") +
  coord_flip()+
  theme_JR()  +
  theme(legend.position = "none")


Familyfig

ggsave("./KBFFigS6.tiff",
       dpi = 600,
       width = 12.5,
       height = 6.65,
       units = "in")
```

```{r polmers Types}
##Number of replicates
nrow(datPoly)


mpolmers <- rma.mv(SMDHyi, SMDHvic,
                   mods = ~ Polymer.types + index.effect,
                   random = list(~1|id2, ~1|Study),
                   data=datPoly, method = "ML")

mpolmersIntML <- rma.mv(SMDHyi, SMDHvic,
                   mods = ~ Polymer.types * index.effect,
                   random = list(~1|id2, ~1|Study),
                   data=datPoly, method = "ML")



##very different
anova(mpolmers, mpolmersIntML)

mpolmersInt <- rma.mv(SMDHyi, SMDHvic,
                      mods = ~ Polymer.types * index.effect - 1,
                      random = list(~1|id2, ~1|Study),
                      method = "REML", data=datPoly)
```

```
mpolmersIntro <- robust(mpolmersInt, cluster=id2, clubSandwich=TRUE)
mpolmersIntro

##Set up emmeans reference grid
mpolmersg <- qdrg(object = mpolmersIntro, data = datPoly)

mpolmersgem <- emmeans(mpolmersg, ~ Polymer.types|index.effect)

#
pairs((mpolmersgem))

```

```{r Polymer Types figure, echo = FALSE}

datfig_polmers <- data.frame(mpolmersgem)
datfig_polmers $index.effect = as.factor(datfig_polmers $index.effect)
datfig_polmers $index.effect = factor(datfig_polmers $index.effect,
                        levels(datfig_polmers
$index.effect)[c(1,3,2,4,5,6,7)])
#make the plot
Polymerfig <- ggplot(datfig_polmers , aes(x = Polymer.types,
                 y=emmean,
                 color = Polymer.types)) +            #subtract 2 to
adjust range of response
  facet_grid(~index.effect, labeller = labeller(index.effect = KBF)) +
  geom_point(position = position_dodge(width = 0.2),
             size = 2.5) +                                  #plot points
  geom_errorbar(aes(ymin = lower.CL,        #include error bars
                    ymax = upper.CL),
                width=0.7,
                position = position_dodge(width = 0.2)) +
#adjust width of bar
  geom_hline(yintercept = 0,                    #make a reference line at
0
             linetype = "dashed",
             color = "black",
             size = 1) +
  xlab("Polymer Types") +               #relabel X and Y axes
  ylab("Hedge's G") +
  coord_flip()+
  theme_JR()  +
  theme(legend.position = "none")


Polymerfig

ggsave("./KBFFigSPolymer.tiff",
       dpi = 600,
       width = 12.5,
       height = 6.65,
       units = "in")
```

```{r, echo = FALSE, eval = FALSE}
#Generate Extended data figure 8
EDF8_WHOLE = cowplot::align_plots(figCont,
                                  Orderfig,
                                  Speciefig,
                                  Familyfig,
                                  Polymerfig,
                                  align = 'hv', axis = 'l')
EDF8_fig = cowplot::plot_grid(EDF8_WHOLE[[1]],
                                 EDF8_WHOLE[[2]],
                                 EDF8_WHOLE[[3]],
                                 EDF8_WHOLE[[4]],
                                 EDF8_WHOLE[[5]],
                               labels = c("A)","B)","C)","D)","E)"),
                               ncol = 1,
                               label_x = 0,
                               label_y = 0.975)
EDF8_fig
ggsave("./EDF8.tiff",
       dpi = 300,
       height = 25,
       width = 22,
       units = "in")
```

Plastic Types

There are differences between Plastic Types and Biological Traits .

```{r Plastic Types}

##Number of replicates
nrow(datPlast)

##Main effect of Plastic Types


mPlastic_Types <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Type + index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datPlast, method = "ML")

mPlastic_TypesIntML <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Type * index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datPlast, method = "ML")


##different; interaction is significant
anova(mPlastic_Types , mPlastic_TypesIntML)


mPlastic_TypesInt <- rma.mv(SMDHyi, SMDHvic,
       mods = ~ Type * index.effect - 1,
```

```
       random = list(~1|id2, ~1|Study),
       data=datPlast)



mPlastic_TypesIntro <- robust(mPlastic_TypesInt, cluster=id2,
clubSandwich=TRUE)
mPlastic_TypesIntro


##Currently, testing whether the effect of KBFs are similar across
endpoints
##Set up emmeans reference grid
mTypeg <- qdrg(object = mPlastic_TypesIntro, data = datPlast)

mTypegem <- emmeans(mTypeg, ~ Type|index.effect)



pairs(mTypegem)
```

```{r Plastic_Types figure, echo = FALSE}
##Data
dat1 <- data.frame(mTypegem)
dat1$index.effect <- as.factor(dat1$index.effect)

dat1$index.effect <- factor(dat1$index.effect,

levels(dat1$index.effect)[c(7,6,5,4,3,2,1)])

#make the plot
Plastic_Types <- ggplot(dat1, aes(x = index.effect,
                 y = emmean,
                 shape = Type,
                 color = index.effect)) +            #subtract 2 to
adjust range of response
  geom_point(position = position_dodge(width = 0.2),
             size = 2.5) +                                #plot points
  geom_errorbar(aes(ymin = lower.CL,        #include error bars
                    ymax = upper.CL),
                width=0.25,
                position = position_dodge(width = 0.2)) +
#adjust width of bar
  # scale_y_continuous(limits = c(0,1.05),      #change limits of Y axis
  #                    breaks = c(0,0.5,1))+      #set Y axis breaks
  geom_hline(yintercept = 0,                   #make a reference line at
0
             linetype = "dashed",
             color = "black",
             size = 1) +
  scale_color_manual(values = rev(c("#5E976E","#222081",
                                    "#b64873","#FFCA3A",
                                    "#58355E","#EC0B43",
```

```
                                  "#63ADF2")),
                      name = "index.effect",
                      guide = "none")+
  scale_shape_manual(values = c(16,17),
                      name = "Plastic Types")+
  xlab("Biological Traits") +               #relabel X and Y axes
  ylab("Hedge's G") +
  coord_flip()+
  theme_JR() +
  theme(legend.position = c(0.4, 0.6))

Plastic_Types
ggsave("./KBFFigS12.tiff",
       dpi = 300,
       width = 6.5,
       height = 4.65,
       units = "in")
```

###Exposure duration

```{r Exposure duration }
##Number of replicates
nrow(datExpos)


mExposureduration <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Exposure.duration + index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datExpos, method = "ML")

mExposuredurationIntML <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Exposure.duration * index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datExpos, method = "ML")


#Diff
anova(mExposureduration, mExposuredurationIntML)

mExposuredurationInt <- rma.mv(SMDHyi, SMDHvic,
                       mods = ~ Exposure.duration * index.effect - 1,
                       random = list(~1|id2, ~1|Study), data=datExpos)


mExposuredurationIntro <- robust(mExposuredurationInt, cluster=id2,
clubSandwich=TRUE)
mExposuredurationIntro

##Set up emmeans reference grid
mExposuredurationg <- qdrg(object = mExposuredurationIntro, data =
datExpos)
```

```
mExposuredurationgem <- emmeans(mExposuredurationg, ~ Exposure.duration |
index.effect)


pairs((mExposuredurationgem))
```

```{r, echo = FALSE}

figdat_Exposur <- data.frame(mExposuredurationgem)
figdat_Exposur$index.effect = as.factor(figdat_Exposur$index.effect)

figdat_Exposur$index.effect = factor(figdat_Exposur$index.effect,

levels(figdat_Exposur$index.effect)[c(7,6,5,4,2,3,1)])

fig_Exposur <- ggplot(figdat_Exposur, aes(x = index.effect,
                        y=emmean,
                        shape = Exposure.duration,
                        color = index.effect)) +            #subtract 2
to adjust range of response
  geom_point(position = position_dodge(width = 0.2),
             size = 2.5) +                                  #plot points
  geom_errorbar(aes(ymin = lower.CL,        #include error bars
                    ymax = upper.CL),
                width=0.25,
                position = position_dodge(width = 0.2)) +
#adjust width of bar
  # scale_y_continuous(limits = c(0,1.05),      #change limits of Y axis
  #                    breaks = c(0,0.5,1))+      #set Y axis breaks
  geom_hline(yintercept = 0,                    #make a reference line at
0
             linetype = "dashed",
             color = "black",
             size = 1) +
  scale_color_manual(values = rev(c("#5E976E","#222081",
                                    "#b64873","#FFCA3A",
                                    "#58355E","#EC0B43",
                                    "#63ADF2")),
                     guide = "none")+
  scale_shape_manual(values = c(16,17,18),
                     name = "Exposure duration")+
  xlab("Biological Traits") +               #relabel X and Y axes
  ylab("Hedge's G") +
  coord_flip()+
  theme_JR() +
  theme(legend.position = c(0.3, 0.5))
fig_Exposur
ggsave("./KBFFigS11.tiff",
       dpi = 300,
       width = 8.5,
       height = 6.65,
       units = "in")
```

Insect Sex

```{r Insect Sex}

##Number of replicates
nrow(datSex)

mInsect_Sex <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Sex + index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datSex, method = "ML")
mInsect_SexIntML <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Sex * index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datSex, method = "ML")

#Difference
anova(mInsect_Sex, mInsect_SexIntML)


mInsect_SexIntro <- rma.mv(SMDHyi, SMDHvic,
                      mods = ~ Sex * index.effect - 1,
                      random = list(~1|id2, ~1|Study), data=datSex)

mInsect_SexIntro <- robust(mInsect_SexIntro, cluster=id2,
clubSandwich=TRUE)
mInsect_SexIntro

##Set up emmeans reference grid
mInsect_SexIntg <- qdrg(object = mInsect_SexIntro, data = datSex)

mInsect_SexIntgem <- emmeans(mInsect_SexIntg, ~ Sex | index.effect)


pairs((mInsect_SexIntgem))

```

```{r Sex Figure, echo = FALSE}

datfig_Sex  <- data.frame(mInsect_SexIntgem)
datfig_Sex$index.effect = as.factor(datfig_Sex$index.effect)

datfig_Sex$index.effect = factor(datfig_Sex$index.effect,

levels(datfig_Sex$index.effect)[c(7,6,5,4,2,3,1)])

#make the plot
fig_Sex <- ggplot(datfig_Sex, aes(x = index.effect,
                        y=emmean,
                        shape = Sex,
                        color = index.effect)) +            #subtract 2
to adjust range of response
```

```
  geom_point(position = position_dodge(width = 0.2),
             size = 2.5) +                                  #plot points
  geom_errorbar(aes(ymin = lower.CL,        #include error bars
                    ymax = upper.CL),
                width=0.7,
                position = position_dodge(width = 0.2)) +
#adjust width of bar
  # scale_y_continuous(limits = c(0,1.05),      #change limits of Y axis
  #                    breaks = c(0,0.5,1))+      #set Y axis breaks
  geom_hline(yintercept = 0,                   #make a reference line at
0
             linetype = "dashed",
             color = "black",
             size = 1) +
  scale_color_manual(values = rev(c("#5E976E","#222081",
                                    "#b64873","#FFCA3A",
                                    "#58355E","#EC0B43",
                                    "#63ADF2")),
                     guide = "none")+
  scale_shape_manual(values = c(16,17),
                     name = "SEX")+
  xlab("Biological Traits") +               #relabel X and Y axes
  ylab("Hedge's G") +
  coord_flip()+
  theme_JR()+
  theme(legend.position = c(0.7, 0.5),
        axis.text.y = element_blank(),
       axis.title.y = element_blank())
fig_Sex
 ggsave("./KBFFigSex.tiff",
        dpi = 300,
        width = 8.5,
        height = 6.65,
        units = "in")

```

Plastic Concentration

```{r Plastic Concentration}

##Number of replicates
nrow(datConcen)

mConcentration <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Concentration + index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datConcen, method = "ML")
mConcentrationIntML <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Concentration * index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datConcen, method = "ML")

#Difference
```

```
anova(mConcentration, mConcentrationIntML)


mConcentrationIntro <- rma.mv(SMDHyi, SMDHvic,
                      mods = ~ Concentration * index.effect - 1,
                      random = list(~1|id2, ~1|Study), data=datConcen)

mConcentrationIntro <- robust(mConcentrationIntro, cluster=id2,
clubSandwich=TRUE)
mConcentrationIntro

##Set up emmeans reference grid
mConcentrationIntg <- qdrg(object = mConcentrationIntro, data =
datConcen)

mConcentrationIntgem <- emmeans(mConcentrationIntg, ~ Concentration |
index.effect)

#Differences in BC and IS
pairs((mConcentrationIntgem))

```

```{r plastic Concentration Figure, echo = FALSE}

datfig_Concentration  <- data.frame(mConcentrationIntgem)
datfig_Concentration$index.effect =
as.factor(datfig_Concentration$index.effect)

datfig_Concentration$index.effect =
factor(datfig_Concentration$index.effect,

levels(datfig_Concentration$index.effect)[c(7,6,5,4,2,3,1)])

#make the plot
fig_Concentration <- ggplot(datfig_Concentration, aes(x = index.effect,
                        y=emmean,
                        shape = Concentration,
                        color = index.effect)) +            #subtract 2
to adjust range of response
  geom_point(position = position_dodge(width = 0.2),
             size = 2.5) +                                  #plot points
  geom_errorbar(aes(ymin = lower.CL,        #include error bars
                    ymax = upper.CL),
                width=0.7,
                position = position_dodge(width = 0.2)) +
#adjust width of bar
  # scale_y_continuous(limits = c(0,1.05),      #change limits of Y axis
  #                    breaks = c(0,0.5,1))+      #set Y axis breaks
  geom_hline(yintercept = 0,                   #make a reference line at
0
             linetype = "dashed",
             color = "black",
             size = 1) +
```

```
  scale_color_manual(values = rev(c("#5E976E","#222081",
                                    "#b64873","#FFCA3A",
                                    "#58355E","#EC0B43",
                                    "#63ADF2")),
                     guide = "none")+
  scale_shape_manual(values = c(16,17,18),
                     name = "Concentration")+
  xlab("Biological Traits") +               #relabel X and Y axes
  ylab("Hedge's G") +
  coord_flip()+
  theme_JR()+
  theme(legend.position = c(0.3, 0.5),
        axis.text.y = element_blank(),
       axis.title.y = element_blank())
fig_Concentration
 ggsave("./KBFFigConcentration.tiff",
        dpi = 300,
        width = 8.5,
        height = 6.65,
        units = "in")

```

Plastic Size

```{r Plastic Size }

##Number of replicates
nrow(datSize)

mSize <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Plastic.Size + index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datSize, method = "ML")
mSizeIntML <- rma.mv(SMDHyi, SMDHvic,
                  mods = ~ Plastic.Size * index.effect,
                  random = list(~1|id2, ~1|Study),
                  data=datSize, method = "ML")

#Difference
anova(mSize, mSizeIntML)


mSizeIntro <- rma.mv(SMDHyi, SMDHvic,
                     mods = ~ Plastic.Size * index.effect - 1,
                     random = list(~1|id2, ~1|Study), data=datSize)

mSizeIntro <- robust(mSizeIntro, cluster=id2, clubSandwich=TRUE)
mSizeIntro

##Set up emmeans reference grid
mSizeIntg <- qdrg(object = mSizeIntro, data = datSize)

mSizeIntgem <- emmeans(mSizeIntg, ~ Plastic.Size | index.effect)
```

```
pairs((mSizeIntgem))

```

```{r  Plastic Size  Figure, echo = FALSE}

datfig_Size  <- data.frame(mSizeIntgem)
datfig_Size $index.effect = as.factor(datfig_Size $index.effect)

datfig_Size $index.effect = factor(datfig_Size $index.effect,
                                   levels(datfig_Size
$index.effect)[c(7,6,5,4,2,3,1)])

#make the plot
fig_Size <- ggplot(datfig_Size , aes(x = index.effect,
                        y=emmean,
                        shape = Plastic.Size,
                        color = index.effect)) +            #subtract 2
to adjust range of response
  geom_point(position = position_dodge(width = 0.2),
             size = 2.5) +                                  #plot points
  geom_errorbar(aes(ymin = lower.CL,        #include error bars
                    ymax = upper.CL),
                width=0.7,
                position = position_dodge(width = 0.2)) +
#adjust width of bar
  # scale_y_continuous(limits = c(0,1.05),      #change limits of Y axis
  #                    breaks = c(0,0.5,1))+      #set Y axis breaks
  geom_hline(yintercept = 0,                    #make a reference line at
0
             linetype = "dashed",
             color = "black",
             size = 1) +
  scale_color_manual(values = rev(c("#5E976E","#222081",
                                    "#b64873","#FFCA3A",
                                    "#58355E","#EC0B43",
                                    "#63ADF2")),
                     guide = "none")+
  scale_shape_manual(values = c(16,17,18),
                     name = "Plastic Size")+
  xlab("Biological Traits") +               #relabel X and Y axes
  ylab("Hedge's G") +
  coord_flip()+
  theme_JR() +
theme(legend.position = c(0.3, 0.5),
        axis.text.y = element_blank(),
       axis.title.y = element_blank())  # Position legend inside plot (x,
y coordinates)

fig_Size
 ggsave("./KBFFigSize.tiff",
        dpi = 300,
```

```
        width = 8.5,
        height = 6.65,
        units = "in")


```

###Generate Extended data figure 9

```{r, echo = FALSE, eval = FALSE}
#Generate Extended data figure 9
EDF9_WHOLE = cowplot::align_plots(fig_Exposur,
                                  fig_Concentration,
                                  fig_Size,
                                  align = 'hv', axis = 'l')
EDF9_fig = cowplot::plot_grid(EDF9_WHOLE[[1]],
                                 EDF9_WHOLE[[2]],
                                 EDF9_WHOLE[[3]],
                                 labels = c("A)","B)","C)"),
                              ncol = 3,
                              label_x = 0,
                              label_y = 0.975)
EDF9_fig
ggsave("./EDF9.tiff",
       dpi = 300,
       height = 5,
       width = 16,
       units = "in")
```

###Generate Extended data figure 10

```{r, echo = FALSE, eval = FALSE}
#Generate Extended data figure 10
EDF10_WHOLE = cowplot::align_plots(Plastic_Types,
                                  fig_Sex,
                                  align = 'hv', axis = 'l')
EDF10_fig = cowplot::plot_grid(EDF10_WHOLE[[1]],
                                 EDF10_WHOLE[[2]],
                                 labels = c("A)","B)"),
                              ncol = 2,
                              label_x = 0,
                              label_y = 0.975)
EDF10_fig
ggsave("./EDF10.tiff",
       dpi = 300,
       height = 6,
       width = 15,
       units = "in")
```

Step 4: Model Selection

```{r, echo = FALSE, eval = FALSE}
```

Step 4: Model Selection:

What factors best explain effc of plastic on inscet?

- Model selection process for each individual KBF
- Only using citation here to stay within the bounds of the meta-analysis framework

Data prep

- For each individual KBF, removed all NA values for predictors. *Cchecked for missing cells .
- As such, replicates for each KBF are below.

```
##Behavioral
##Remove all rows with NA values in ANY of the categories
dataBE2 = meta1 %>%
  filter(index.effect == "Behavioral" &
           complete.cases(Species) &
           complete.cases(Family) &
           complete.cases(Order) &
           complete.cases(Type) &
           complete.cases(Sex) &
           Sex != "-" &
           complete.cases(Stage) &
           complete.cases(End.Point)&
           complete.cases(Polymer.types) &
           complete.cases(Concentration)&
           complete.cases(Exposure.duration)&
           complete.cases(Plastic.Size)&
           complete.cases(SMDHyi) &
           complete.cases(SMDHvic))

##Replicates for Behavioral (44)
nrow(dataBE2)

##Development
##Remove all rows with NA values in ANY of the categories
dataDE2 = meta1 %>%
  filter(index.effect == "Development" &
           complete.cases(Species) &
           complete.cases(Family) &
           complete.cases(Order) &
           complete.cases(Type) &
           complete.cases(Sex) &
           Sex != "-" &
           complete.cases(Stage) &
           complete.cases(End.Point)&
           complete.cases(Polymer.types) &
           complete.cases(Concentration)&
```

```
            complete.cases(Exposure.duration)&
            complete.cases(Plastic.Size)&
            complete.cases(SMDHyi) &
            complete.cases(SMDHvic))

##Replicates for Development (31)
nrow(dataDE2)

#Fecundity
##Remove all rows with NA values in ANY of the categories
dataFEC2 = meta1 %>%
  filter(index.effect == "Fecundity",
         complete.cases(Species) &
           complete.cases(Family) &
           complete.cases(Country) &
           complete.cases(Order) &
           complete.cases(Type) &
           complete.cases(Sex) &
           Sex != "-" &
           complete.cases(Stage) &
           complete.cases(End.Point)&
           complete.cases(Polymer.types)&
           complete.cases(Concentration)&
           complete.cases(Exposure.duration)&
           complete.cases(Plastic.Size)&
           complete.cases(SMDHyi) &
           complete.cases(SMDHvic))


##Replicates for Fecundity (21)
nrow(dataFEC2)

##Feeding
##Remove all rows with NA values in ANY of the categories
dataFE2 = meta1 %>%
  filter(index.effect == "Feeding" &
           complete.cases(Species) &
           complete.cases(Family) &
           complete.cases(Order) &
           complete.cases(Type) &
           complete.cases(Sex) &
           Sex != "-" &
           complete.cases(Stage) &
           complete.cases(End.Point)&
           complete.cases(Polymer.types) &
           complete.cases(Concentration)&
           complete.cases(Exposure.duration)&
           complete.cases(Plastic.Size)&
           complete.cases(SMDHyi) &
           complete.cases(SMDHvic))

##Replicates for Feeding (26)
nrow(dataFE2)
```

```
##Growth
##Remove all rows with NA values in ANY of the categories
dataGR2 = meta1 %>%
  filter(index.effect == "Growth" &
         complete.cases(Species) &
           complete.cases(Family) &
           complete.cases(Order) &
           complete.cases(Type) &
           complete.cases(Sex) &
           Sex != "-" &
           complete.cases(Stage) &
           complete.cases(End.Point)&
           complete.cases(Polymer.types) &
           complete.cases(Concentration)&
           complete.cases(Exposure.duration)&
           complete.cases(Plastic.Size)&
           complete.cases(SMDHyi) &
           complete.cases(SMDHvic))

##Replicates for Growth (37)
nrow(dataGR2)



##Health
##Remove all rows with NA values in ANY of the categories
dataHE2 = meta1 %>%
  filter(index.effect == "Health" &
         complete.cases(Species) &
           complete.cases(Family) &
           complete.cases(Order) &
           complete.cases(Type) &
           complete.cases(Sex) &
           Sex != "-" &
           complete.cases(Stage) &
           complete.cases(End.Point)&
           complete.cases(Polymer.types) &
           complete.cases(Concentration)&
           complete.cases(Exposure.duration)&
           complete.cases(Plastic.Size)&
           complete.cases(SMDHyi) &
           complete.cases(SMDHvic))

##Replicates for Health (126)
nrow(dataHE2)


##Survival
##Remove all rows with NA values in ANY of the categories
dataSU2 = meta1 %>%
  filter(index.effect == "Survival" &
         complete.cases(Species) &
           complete.cases(Family) &
           complete.cases(Order) &
```

```
            complete.cases(Type) &
            complete.cases(Sex) &
            Sex != "-" &
            complete.cases(Stage) &
            complete.cases(End.Point)&
            complete.cases(Polymer.types) &
            complete.cases(Concentration)&
            complete.cases(Exposure.duration)&
            complete.cases(Plastic.Size)&
            complete.cases(SMDHyi) &
            complete.cases(SMDHvic))

##Replicates for Survival (6)
nrow(dataSU2)
```

Data Analysis

-   Model selection output is for Δ AIC \< 2.
-   Best models are AICc indicated best models, not including parsimony in best model.

```{r}

##rma() wrapper for dredge
makeArgs.rma <- function(obj, termNames, comb, opt,
                           ...) {
  ret <- MuMIn:::makeArgs.default(obj, termNames, comb, opt)
  names(ret)[1L] <- "mods"
  ret
}

##rma() wrapper for dredge
coefTable.rma <- function(model, ...) {
  MuMIn:::.makeCoefTable(model$b, model$se, coefNames =
rownames(model$b))
}


```

Behavioral

```{r Behavioral, echo = FALSE,  warning = FALSE}
##Do this once
options(na.action = "na.fail")

 Beha_Whole <- rma.mv(SMDHyi, SMDHvic,
                         mods = ~ Species +
                           Family +
                           Sex + End.Point +
                           Polymer.types + Concentration +
                           Exposure.duration + Plastic.Size,
                         random = list(~1|id2, ~1|Study),
```

```
                        data = dataBE2, method = "ML")

 Beha_Dredge = dredge(Beha_Whole)
 saveRDS(Beha_Dredge, "K:/Abbas/Meta-Analysis/Meta-
Analysis/Data/Beha_Dredge.RDS")



BioDiv_Dredge <- readRDS("K:/Abbas/Meta-Analysis/Meta-
Analysis/Data/Beha_Dredge.RDS")


summary(model.avg(BioDiv_Dredge, delta < 2))
sw(subset(BioDiv_Dredge, delta < 4))

```

Development

```{r Climate Change, echo = FALSE,  warning = FALSE}
 Devel_Whole <- rma.mv(SMDHyi, SMDHvic,
                        mods = ~ Species +
                          Family +
                          Type +
                          Sex + End.Point +
                          Polymer.types + Concentration +
                          Exposure.duration + Plastic.Size,
                        random = list(~1|id2, ~1|Study),
                        data = dataDE2, method = "ML")

 Devel_Dredge = dredge(Devel_Whole)

 saveRDS(Devel_Dredge, "K:/Abbas/Meta-Analysis/Meta-
Analysis/Data/Devel_Dredge.RDS")
Devel_Dredge <- readRDS("K:/Abbas/Meta-Analysis/Meta-
Analysis/Data/Devel_Dredge.RDS")

summary(model.avg(Devel_Dredge, delta < 2))
sw(subset(Devel_Dredge, delta < 4))

```

Fecundity

```{r Fecundity, echo = FALSE,  warning = FALSE}
  Fecun_Whole <- rma.mv(SMDHyi, SMDHvic,
                        mods = ~ Species +
                          Family +
                          End.Point +
                          Polymer.types + Concentration +
                          Exposure.duration + Plastic.Size,
                        random = list(~1|id2, ~1|Study),
                        data = dataFEC2, method = "ML")
```

```
 Fecun_Dredge = dredge(Fecun_Whole, trace = 2)

 saveRDS(Fecun_Dredge, "K:/Abbas/Meta-Analysis/Meta-
Analysis/Data/Fecun_Dredge.RDS")

HabLoss_Dredge <- readRDS("K:/Abbas/Meta-Analysis/Meta-
Analysis/Data/Fecun_Dredge.RDS")

summary(model.avg(Fecun_Dredge, delta < 2))
sw(subset(Fecun_Dredge, delta < 4))
```

Feeding

```{r Feeding, echo = FALSE,  warning = FALSE}

 Feeding_Whole <- rma.mv(SMDHyi, SMDHvic,
                         mods = ~ Sex + Polymer.types + Concentration +
                           Exposure.duration + Plastic.Size,
                         random = list(~1|id2, ~1|Study),
                         data = dataFE2, method = "ML")

 Feeding_Dredge = dredge(Feeding_Whole)
 saveRDS(Feeding_Dredge, "K:/Abbas/Meta-Analysis/Meta-
Analysis/Data/Feeding_Dredge.RDS")

IntroSpec_Dredge <- readRDS("K:/Abbas/Meta-Analysis/Meta-
Analysis/Data/Feeding_Dredge.RDS")


summary(model.avg(Feeding_Dredge, delta < 2))
sw(subset(Feeding_Dredge, delta < 4))

```

Growth

```{r Growth, echo = FALSE, warning = FALSE}
 Growth_Whole <- rma.mv(SMDHyi, SMDHvic,
                         mods = ~ Species +
                           Family +
                           Sex + End.Point +
                           Polymer.types + Concentration +
                           Exposure.duration + Plastic.Size,
                         random = list(~1|id2, ~1|Study),
                         data = dataGR2, method = "ML")

 Growth_Dredge = dredge(Growth_Whole)
```

```
 saveRDS(Growth_Dredge, "K:/Abbas/Meta-Analysis/Meta-
Analysis/Data/Growth_Dredge.RDS")

Growth_Dredge <- readRDS("K:/Abbas/Meta-Analysis/Meta-
Analysis/Data/Growth_Dredge.RDS")


summary(model.avg(Growth_Dredge, delta < 2))
sw(subset(Growth_Dredge, delta < 4))
```

Health

```{r Health, echo = FALSE, warning = FALSE}
 Health_Whole <- rma.mv(SMDHyi, SMDHvic,
                        mods = ~ Species +
                          Family +
                          Type +
                          Sex + End.Point +
                          Polymer.types + Concentration +
                          Exposure.duration + Plastic.Size,
                        random = list(~1|id2, ~1|Study),
                        data = dataHE2, method = "ML")

 Health_Dredge = dredge(Health_Whole)
 saveRDS(Health_Dredge, "K:/Abbas/Meta-Analysis/Meta-
Analysis/Data/Health_Dredge.RDS")

Health_Dredge <- readRDS("K:/Abbas/Meta-Analysis/Meta-
Analysis/Data/Health_Dredge.RDS")


summary(model.avg(Health_Dredge, delta < 2))
sw(subset(Health_Dredge, delta < 4))
```

Surviuval

```{r Surviuval, echo = FALSE, warning = FALSE}
 Surviuval_Whole <- rma.mv(SMDHyi, SMDHvic,
                        mods = ~ Sex,
                        random = list(~1|id2, ~1|Study),
                        data = dataGR2, method = "ML")

 Surviuval_Dredge = dredge(Surviuval_Whole)
 saveRDS(Surviuval_Dredge, "K:/Abbas/Meta-Analysis/Meta-
Analysis/Data/Surviuval_Dredge.RDS")

Surviuval_Dredge <- readRDS("K:/Abbas/Meta-Analysis/Meta-
Analysis/Data/Surviuval_Dredge.RDS")
```

```
summary(model.avg(Surviuval_Dredge))
sw(subset(Surviuval_Dredge, delta < 4))

```

```{r single Importplot, echo = FALSE, warning = FALSE}
#BRING ALL OF THE IMPORTANCE SCORE FIGURES TOGETHER IN 1
Beha_import <- data.frame(
  Variable = attr(sw(subset(Beha_Dredge, delta < 4)), "names"),
  Importance = c(sw(subset(Beha_Dredge, delta < 4))),
  Nmodel = attr(sw(subset(Beha_Dredge, delta < 4)), "n.models")
)

# Make sure there are 8 variable names, one for each row
Beha_import$Variable <- c("End Point",
                          "Polymer Types",
                          "Family",
                          "Species",
                          "Sex",
                          "Exposure Duration",
                          "Plastic Size",
                          "Concentration"  # Add another variable to
match the length
                          )

Beha_import$Variable = forcats::fct_reorder(Beha_import$Variable,
Beha_import$Importance, .desc = FALSE)
levels(Beha_import$Variable) =
rev(as.character(forcats::fct_reorder(Beha_import$Variable,
Beha_import$Importance, .desc = FALSE)))

importplot_Beha = ggplot(Beha_import, aes(x = Variable, y = Importance))+
  geom_segment( aes(x=Variable, xend=Variable, y=0, yend=Importance),
                color="#5E976E") +
  geom_point( size=5, color="#5E976E", shape = 21, fill = "#5E976E") +
  scale_y_continuous(expand = c(0,0),
                     limits = c(0, 1.05))+
  # annotate("text", x=1.6, y=0.85, label=bdlb1, parse=TRUE)+
  # annotate("text", x=1, y=0.85, label=bdlb2, parse=TRUE)+
  ylab("Importance Score")+
  ggtitle("Behavioral Response") +
  coord_flip()+
  theme_JR()+
  theme(
    axis.title.y = element_blank(),
    axis.text.y = element_text(size = 12),
    plot.title = element_text(hjust = 0.5),  # Center the title
    panel.border = element_blank(),  # Remove the border around the plot
    panel.background = element_blank(),  # Remove the background
    axis.line = element_line(color = "black")  # Keep axis lines
  )


importplot_Beha
```

```
Devel_import = data.frame(
  Variable = attr(sw(subset(Devel_Dredge, delta < 4)), "names"),
  Importance = c(sw(subset(Devel_Dredge, delta < 4))),
  Nmodel = attr(sw(subset(Devel_Dredge, delta < 4)), "n.models"))



Devel_import$Variable = c("Concentration",
                              "Exposure Duration",
                              "Family",
                              "Sex",
                              "End Point",
                              "Type",
                          "Plastic.Size",
                          "Polymer.types",
                          "Species")

Devel_import$Variable = forcats::fct_reorder(Devel_import$Variable,
Devel_import$Importance, .desc = FALSE)
levels(Devel_import$Variable) =
rev(as.character(forcats::fct_reorder(Devel_import$Variable,
Devel_import$Importance, .desc = FALSE)))

importplot_Devel = ggplot(Devel_import, aes(x = Variable, y =
Importance))+
  geom_segment( aes(x=Variable, xend=Variable, y=0, yend=Importance),
                color="#FFCA3A") +
  geom_point( size=5, color="#FFCA3A", shape = 21, fill = "#FFCA3A") +
  scale_y_continuous(expand = c(0,0),
                     limits = c(0, 1.05))+
  # annotate("text", x=1.6, y=0.85, label=cclb1, parse=TRUE)+
  # annotate("text", x=1, y=0.85, label=cclb2, parse=TRUE)+
  ylab("Importance Score")+
  ggtitle("Development") +
  coord_flip()+
  theme_JR()+
  theme(
    axis.title.y = element_blank(),
    axis.text.y = element_text(size = 12),
    plot.title = element_text(hjust = 0.5),  # Center the title
    panel.border = element_blank(),  # Remove the border around the plot
    panel.background = element_blank(),  # Remove the background
    axis.line = element_line(color = "black")  # Keep axis lines
  )

importplot_Devel


Feeding_import = data.frame(
  Variable = attr(sw(subset(Feeding_Dredge, delta < 4)), "names"),
  Importance = c(sw(subset(Feeding_Dredge, delta < 4))),
  Nmodel = attr(sw(subset(Feeding_Dredge, delta < 4)), "n.models"))
```

```
Feeding_import$Variable = c("Exposure Duration",
                            "Concentration",
                            "Polymer Types",
                            "Plastic Size",
                            "Sex"
)

Feeding_import$Variable = forcats::fct_reorder(Feeding_import$Variable,
Feeding_import$Importance, .desc = FALSE)
levels(Feeding_import$Variable) =
rev(as.character(forcats::fct_reorder(Feeding_import$Variable,
Feeding_import$Importance, .desc = FALSE)))

importplot_Feeding = ggplot(Feeding_import, aes(x = Variable, y =
Importance))+
  geom_segment( aes(x=Variable, xend=Variable, y=0, yend=Importance),
                color="#EC0B43") +
  geom_point( size=5, color="#EC0B43", shape = 21, fill = "#EC0B43") +
  scale_y_continuous(expand = c(0,0),
                     limits = c(0, 1.05))+
  ylab("Importance Score")+
  ggtitle("Feeding") +
  coord_flip()+
  theme_JR()+
  theme(
    axis.title.y = element_blank(),
    axis.text.y = element_text(size = 12),
    plot.title = element_text(hjust = 0.5),  # Center the title
    panel.border = element_blank(),  # Remove the border around the plot
    panel.background = element_blank(),  # Remove the background
    axis.line = element_line(color = "black")  # Keep axis lines
  )


importplot_Feeding



Growth_import = data.frame(
  Variable = attr(sw(subset(Growth_Dredge, delta < 4)), "names"),
  Importance = c(sw(subset(Growth_Dredge, delta < 4))),
  Nmodel = attr(sw(subset(Growth_Dredge, delta < 4)), "n.models"))


Growth_import$Variable = c("Concentration",
                           "Exposure Duration",
                           "Sex",
                           "Plastic Size",
                           "Family",
                           "Polymer.types",
                           "Species",
                           "End.Point")
```

```
Growth_import$Variable = forcats::fct_reorder(Growth_import$Variable,
Growth_import$Importance, .desc = FALSE)
levels(Growth_import$Variable) =
rev(as.character(forcats::fct_reorder(Growth_import$Variable,
Growth_import$Importance, .desc = FALSE)))

importplot_Growth = ggplot(Growth_import, aes(x = Variable, y =
Importance))+
  geom_segment( aes(x=Variable, xend=Variable, y=0, yend=Importance),
                color="#63ADF2") +
  geom_point( size=5, color="#63ADF2", shape = 21, fill = "#63ADF2") +
  scale_y_continuous(expand = c(0,0),
                     limits = c(0, 1.05))+
  ylab("Importance Score")+
  ggtitle("Growth") +
  coord_flip()+
  theme_JR()+
  theme(
    axis.title.y = element_blank(),
    axis.text.y = element_text(size = 12),
    plot.title = element_text(hjust = 0.5),  # Center the title
    panel.border = element_blank(),  # Remove the border around the plot
    panel.background = element_blank(),  # Remove the background
    axis.line = element_line(color = "black")  # Keep axis lines
  )


importplot_Growth



Health_import = data.frame(
  Variable = attr(sw(subset(Health_Dredge, delta < 4)), "names"),
  Importance = c(sw(subset(Health_Dredge, delta < 4))),
  Nmodel = attr(sw(subset(Health_Dredge, delta < 4)), "n.models"))


Health_import$Variable = c("Type",
                           "Concentration",
                           "Sex",
                           "Plastic Size",
                           "Exposure duration")


Health_import$Variable = forcats::fct_reorder(Health_import$Variable,
Health_import$Importance, .desc = FALSE)
levels(Health_import$Variable) =
rev(as.character(forcats::fct_reorder(Health_import$Variable,
Health_import$Importance, .desc = FALSE)))

importplot_Health = ggplot(Health_import, aes(x = Variable, y =
Importance))+
  geom_segment( aes(x=Variable, xend=Variable, y=0, yend=Importance),
```

```
                color="#58355E") +
  geom_point( size=5, color="#58355E", shape = 21, fill = "#58355E") +
  scale_y_continuous(expand = c(0,0),
                     limits = c(0, 1.05))+
  ylab("Importance Score")+
  ggtitle("Health") +
  coord_flip()+
  theme_JR()+
  theme(
    axis.title.y = element_blank(),
    axis.text.y = element_text(size = 12),
    plot.title = element_text(hjust = 0.5),  # Center the title
    panel.border = element_blank(),  # Remove the border around the plot
    panel.background = element_blank(),  # Remove the background
    axis.line = element_line(color = "black")  # Keep axis lines
  )



importplot_Health



import_whole = cowplot::align_plots(importplot_Beha, importplot_Devel,
                                    importplot_Feeding,
importplot_Growth,
                                    importplot_Health,
                                    align = 'hv', axis = 'l')

importplotplot <- cowplot::plot_grid(import_whole[[1]],
import_whole[[2]], import_whole[[3]],
                                     import_whole[[4]], import_whole[[5]],
                                     labels = c("A)","B)","C)","D)","E)"),
                                     ncol = 2,
                                     label_x = 0.9,
                                     label_y = 0.275)
importplotplot

ggsave("./ImportancePlotsKBF.tiff",importplotplot,
       dpi = 300,
       width = 4.75, ##Need to update this..
       height = 4.75,
       units = "in", scale = 2)


```
```
```