TCBD Results - Draft V2

Ciara Hovis, CSIS Lab, Michigan State University

14 October 2020

Table of Contents

# 1. Summary

Synthesis of results for telacoupling/biodiversity systematic review. It consists of response summaries and visualizations of the accepted papers.

# 2. R Setup

The script presented here was done using R (version 4.0.2; R Core Team 2020) and its packages.

Load libraries, directories, and custom functions from source file.

# 3. Load data

# surveys  
 ab\_screen <- read.csv(paste0(dat.dir, 'absScreening\\AbsScreen1\_Merge.csv'))  
 ab\_maybes <- read.csv(paste0(dat.dir, 'absScreening\\AbsScreen2\_maybes\_to\_keep.csv'))  
 ab\_yeses <- read.csv(paste0(dat.dir, 'absScreening\\final\_sample\_for\_coding.csv'))  
 assignments <- read.csv(paste0(dat.dir,'absScreening\\Coding\_MergeAssign\_R1.csv'))  
  
  
 survey1 <- read.csv(paste0(dat.dir,'survey1\_cleaned.csv'))  
 s2 <- read.csv(paste0(dat.dir,'survey2\_cleaned.csv'))  
 survey3 <- read.csv(paste0(dat.dir,'survey3\_cleaned.csv'))  
 s23 <- read.csv(paste0(dat.dir,'s23.csv'))  
   
 taxEff<- read.csv(paste0(dat.dir,'tax\_eff.csv'))  
   
# factor format  
 s23$maj\_eff<-factor(s23$maj\_eff, c("Beneficial","Harmful","Changed", "Mixed","Not significant or unclear"))  
 s23\_maj <- s23[!duplicated(s23$paper\_id),]  
   
 taxEff$maj\_eff\_taxa<-factor(taxEff$maj\_eff\_taxa, c("Beneficial","Harmful","Changed", "Mixed","Not significant or unclear"))  
 taxEff\_maj <- taxEff[!duplicated(taxEff$tax\_ID),]

# 4. Survey 1 & 2 Results

## 4.1 Sample Results

Accepted/rejected published papers over time

Total number of papers at the coding stage = 591

# get total numbers but also ensure both values here match  
 nrow(survey1)  
 length(unique(survey1$paper\_id))

## [1] 591  
## [1] 591

Number of accepted and rejected papers.

Accepted = 131 (22%) , Rejected = 460 (78%)

# get summary  
 ddply(survey1, .(status), summarize,  
 # total count of entries   
 count=length(status))  
sum(survey1$status == 'accept')/nrow(survey1)  
sum(survey1$status == 'reject')/nrow(survey1)

## status count  
## 1 accept 131  
## 2 reject 460  
## [1] 0.2216582  
## [1] 0.7783418

Visualizing accepted papers over time, including the ones from the abstract screening process.

# all abstract screened papers (n=7306)  
 ab\_screen <- ab\_screen[!ab\_screen$PY==2020,] #remove 2020 studies  
 ab\_all <- ab\_screen[!duplicated(ab\_screen$TITLE),];nrow(ab\_all)  
  
# remove the papers accepted in screening from the abstract screening list (n=6665)  
 ab\_reject <- anti\_join(ab\_all,ab\_maybes,by=c("TITLE"))  
 ab\_reject <- anti\_join(ab\_reject,ab\_yeses,by=c("TITLE"))  
 ab\_reject <- ab\_reject[!duplicated(ab\_reject$TITLE,ab\_reject$PY),];nrow(ab\_reject)  
   
# all papers that were accepted from abstract screening (n=641)  
 ab\_accept <- ab\_yeses[!duplicated(ab\_yeses$TITLE),]  
  
   
# accepted papers from the assignment list for full review (n=595)  
 papers\_reviewed <- assignments[!duplicated(assignments$TITLE),]  
  
# papers with PDF not available or rejected because of language (n=46)  
 paper\_unavail <- anti\_join(ab\_accept,papers\_reviewed,by=c("TITLE"))  
 paper\_unavail <- paper\_unavail[!duplicated(paper\_unavail$TITLE),]  
   
# rejected papers or not yet reviewed (n=460)  
   
 keep <- subset(survey1, status=='accept')  
 reject <- anti\_join(survey1,keep)  
  
 papers\_rejected <- anti\_join(papers\_reviewed,keep,by=c("STUDY\_ID.1"="paper\_id"))  
 papers\_rejected <- papers\_rejected[!duplicated(papers\_rejected$TITLE),]

## [1] 7306  
## [1] 6665

# show numbers:  
 paste('original number of papers in WOS search:',nrow(ab\_screen))  
 paste('number of duplicates in WOS search:',nrow(ab\_screen)-nrow(ab\_all))  
 paste('number of abstracts screened:',nrow(ab\_all))  
 paste('number of accepted abstracts:',nrow(ab\_accept))  
 paste('number of rejected abstracts:',nrow(ab\_reject))  
 paste('number of accepted papers unavailable or non-English:',  
 nrow(paper\_unavail))  
 paste('number of papers available for full review:',  
 nrow(papers\_reviewed))  
 paste('number of full papers accepted for synthesis:',  
 nrow(keep))   
 paste('number of full papers rejected from synthesis:',nrow(reject))  
 paste('number of full papers not yet reviewed for synthesis:',  
 nrow(papers\_rejected)-nrow(reject)-5)  
   
# num of full papers not yet reviewed is -1, idk why. Maybe due to common papers? The check below is True so I'm not worried.   
   
   
# test if they add up correctly  
 paste('COUNT TESTS:')  
 paste('total full papers:',  
 nrow(papers\_reviewed) == nrow(ab\_accept) - nrow(paper\_unavail))  
 paste('total abstract screen:',  
 nrow(ab\_all) == nrow(ab\_reject) + nrow(ab\_accept))

## [1] "original number of papers in WOS search: 7318"  
## [1] "number of duplicates in WOS search: 12"  
## [1] "number of abstracts screened: 7306"  
## [1] "number of accepted abstracts: 641"  
## [1] "number of rejected abstracts: 6665"  
## [1] "number of accepted papers unavailable or non-English: 46"  
## [1] "number of papers available for full review: 595"  
## [1] "number of full papers accepted for synthesis: 131"  
## [1] "number of full papers rejected from synthesis: 460"  
## [1] "number of full papers not yet reviewed for synthesis: -1"  
## [1] "COUNT TESTS:"  
## [1] "total full papers: TRUE"  
## [1] "total abstract screen: TRUE"

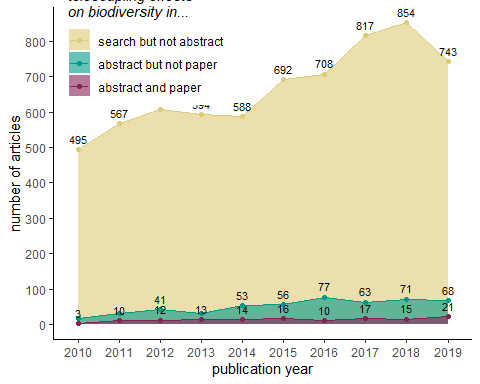
Next, simplify columns and combine for three tiers in the plot.

# select columns  
 # papers found in WOS but don't meet criteria  
 reject\_abstract\_level <- subset(ab\_reject,select=c("STUDY\_ID","PY"))  
 # WOS papers that meet criteria but don't directly look at coupling effects on biodiv  
 reject\_paper\_level <- subset(papers\_rejected,select=c("STUDY\_ID.1","PY"))  
 reject\_paper\_level2 <- subset(paper\_unavail,select=c("STUDY\_ID","PY"))  
 # papers that do look at coupling effects on biodiv  
 accept\_paper\_level <- subset(keep,select=c("paper\_id","year\_pub"))  
   
# rename columns  
 colnames(reject\_abstract\_level) <- c('paper\_id','year\_pub')  
 colnames(reject\_paper\_level) <- c('paper\_id','year\_pub')  
 colnames(reject\_paper\_level2) <- c('paper\_id','year\_pub')  
   
# add columns  
 #papers that discuss tc effects on biodiv in...  
 reject\_abstract\_level$type <- 'search but not abstract'  
 reject\_paper\_level$type <- 'abstract but not paper'  
 reject\_paper\_level2$type <- 'abstract but not paper'  
 accept\_paper\_level$type <- 'abstract and paper'  
   
# rbind (should be 7307 rows)  
 papers <- rbind(reject\_abstract\_level,reject\_paper\_level,  
 reject\_paper\_level2, # <-- NOT SURE IF UNAVAILABLE PAPERS BELONG HERE  
 accept\_paper\_level)  
  
# check  
 nrow(papers)

## [1] 7306

Plot image.

# Get a count of records per year  
 paper\_ct <- ddply(papers, .(year\_pub,type), summarize, count=length(year\_pub))  
  
# Change to factors  
 paper\_ct$year\_pub <- as.factor(paper\_ct$year\_pub)  
 paper\_ct$type <- factor(paper\_ct$type,  
 levels = c('search but not abstract',  
 'abstract but not paper',  
 'abstract and paper'))  
   
 papers.fig <- ggplot(paper\_ct, aes(x=year\_pub, y=count)) +   
 geom\_area(position="identity",  
 aes(y=count, fill = type, group = type), alpha=0.6) +  
 geom\_point(aes(y =count, color = type, group = type))+  
 geom\_line(aes(y =count, color = type, group = type))+  
 geom\_text(aes(label=count),size=3,  
 position=position\_dodge(.7),  
 vjust=-.7,  
 check\_overlap=TRUE)+  
 scale\_color\_manual(  
 name=expression(italic("telecoupling effects\non biodiversity in...")),  
 values=paper\_col)+  
 scale\_fill\_manual(  
 name=expression(italic("telecoupling effects\non biodiversity in...")),  
 values=paper\_col)+  
 xlab("publication year")+  
 ylab("number of articles") + ylim(0,860)+  
 scale\_y\_continuous(breaks=seq(0,850,100)) +  
 theme\_classic()+  
 theme(legend.position = c(0.22,0.87))  
   
# show here  
 papers.fig



Acceptance rate for abstract screening.

# get total abstract acceptance rate  
 sum(ab\_all$INCLUDE=='YES')/nrow(ab\_all)

## [1] 0.08007117

Acceptance rate for papers that were reviewed (**NOTE:** this includes the 5 common surveys).

# get total paper acceptance rate  
 sum(survey1$status=='accept')/nrow(survey1)

## [1] 0.2216582

Acceptance rate per person (**NOTE:** this excludes the 5 common surveys per person, but has a rate for it on its own as “all”).

# get summary  
 acr <- ddply(survey1, .(coder\_id), summarize,  
 # total count of entries   
 count=length(coder\_id),  
 accept=sum(status=='accept'),  
 reject=sum(status=='reject'),  
 # percent acceptance  
 accept\_rate=sum(status=='accept')/length(coder\_id))  
 acr

## coder\_id count accept reject accept\_rate  
## 1 A\_Herzberger 63 20 43 0.3174603  
## 2 A\_Torres 51 7 44 0.1372549  
## 3 all 5 4 1 0.8000000  
## 4 C\_Hovis 92 19 73 0.2065217  
## 5 E\_Dean 43 7 36 0.1627907  
## 6 E\_Xing 43 9 34 0.2093023  
## 7 K\_Kapsar 52 11 41 0.2115385  
## 8 M\_Lei 43 15 28 0.3488372  
## 9 MG\_Chung 43 5 38 0.1162791  
## 10 X\_Wu 67 18 49 0.2686567  
## 11 Y\_Dou 5 0 5 0.0000000  
## 12 Y\_Li 32 2 30 0.0625000  
## 13 Y\_Zhang 52 14 38 0.2692308

Average per-person acceptance rates

# mean per-person acceptance rate   
 mean(acr$accept\_rate)

## [1] 0.2392594

## 4.2 General Results

Counts of papers that were inferred (accepted papers only)

Total rate of **inferring** across papers.

# get total papers for which the tc relationships were inferred  
 paste('number inferred papers:',sum(keep$explicit\_distant\_impacts=='No'))  
 paste('number explicit papers:',sum(keep$explicit\_distant\_impacts=='Yes'))  
   
# get total inferring rate  
 paste('inferred percent:',sum(keep$explicit\_distant\_impacts=='No')/nrow(keep))  
 paste('explicit percent:',sum(keep$explicit\_distant\_impacts=='Yes')/nrow(keep))

## [1] "number inferred papers: 46"  
## [1] "number explicit papers: 76"  
## [1] "inferred percent: 0.351145038167939"  
## [1] "explicit percent: 0.580152671755725"

Percent of papers inferred per person per person

# get summary  
 ddply(keep, .(coder\_id), summarize,  
 # total count of entries   
 count=length(coder\_id),  
 inferred=sum(explicit\_distant\_impacts=='No'),  
 explicit=sum(explicit\_distant\_impacts=='Yes'),  
 # percent inferred  
 inferred\_rate=sum(explicit\_distant\_impacts=='No')/length(coder\_id))

## coder\_id count inferred explicit inferred\_rate  
## 1 A\_Herzberger 20 9 11 0.4500000  
## 2 A\_Torres 7 2 5 0.2857143  
## 3 all 4 2 2 0.5000000  
## 4 C\_Hovis 19 5 14 0.2631579  
## 5 E\_Dean 7 2 5 0.2857143  
## 6 E\_Xing 9 0 0 0.0000000  
## 7 K\_Kapsar 11 7 4 0.6363636  
## 8 M\_Lei 15 6 9 0.4000000  
## 9 MG\_Chung 5 1 4 0.2000000  
## 10 X\_Wu 18 2 16 0.1111111  
## 11 Y\_Li 2 0 2 0.0000000  
## 12 Y\_Zhang 14 10 4 0.7142857

Counts of study types (accepted papers only)

# get summary  
 ddply(keep, .(C4\_study), summarize,  
 # total count of entries   
 count=length(C4\_study),  
 percent=length(C4\_study)/nrow(keep))

## C4\_study count percent  
## 1 Experimental study 3 0.022900763  
## 2 Modeling study (using empirical data and not projected) 11 0.083969466  
## 3 Observational study 116 0.885496183  
## 4 Observational/Modeling study 1 0.007633588

Habitat Frequency (study level, n= 131)

habitat.study <- count(s2, 'habitat')  
print(habitat.study)  
prop.table(table(s2$habitat))

## habitat freq  
## 1 Freshwater 22  
## 2 Freshwater;Marine 1  
## 3 Marine 25  
## 4 Not specified/Global study 2  
## 5 Terrestrial 75  
## 6 Terrestrial;Freshwater 4  
## 7 Terrestrial;Marine 2  
##   
## Freshwater Freshwater;Marine   
## 0.167938931 0.007633588   
## Marine Not specified/Global study   
## 0.190839695 0.015267176   
## Terrestrial Terrestrial;Freshwater   
## 0.572519084 0.030534351   
## Terrestrial;Marine   
## 0.015267176

Taxa Frequency (study level, n=131)

desired\_length <- 8   
empty\_list <- vector(mode = "list", length = desired\_length)  
  
taxa <- (s2$taxa)  
table(taxa)  
taxa.list <- c('Mammals', 'Plants', 'Reptiles', 'Amphibians', 'Birds', 'Fish', 'Invertebrates', 'Other')  
taxa.count <- empty\_list  
  
for (i in 1:length(taxa.list)) {  
   
 taxa.count[i] <- sum(str\_count(taxa, taxa.list[i]))  
 }  
  
taxa.count <- rbind(taxa.list,taxa.count)  
taxa.count  
  
multiple <- str\_count(taxa, ';')  
multiple <- multiple != 0  
multiple <- table(multiple) ["TRUE"]  
multiple  
  
# Others? Not a big deal since taxa is clarified in survey 3.Not a big deal.  
  
othTaxa <- s2[which(s2$taxa == 'Other'),]

## taxa  
## Amphibians; Plants   
## 1   
## Birds   
## 36   
## Birds; Mammals   
## 1   
## Birds; Mammals; Amphibians; Plants   
## 1   
## Birds; Mammals; Reptiles; Invertebrates; Plants   
## 1   
## Birds; Reptiles   
## 1   
## Fish   
## 14   
## Fish; Invertebrates   
## 1   
## Invertebrates   
## 23   
## Invertebrates; Plants   
## 2   
## Invertebrates;Plants   
## 1   
## Mammals   
## 14   
## Mammals; Reptiles; Amphibians; Invertebrates   
## 1   
## Other   
## 3   
## Plants   
## 26   
## Plants; habitat pollution   
## 1   
## Reptiles   
## 3   
## Reptiles;Amphibians   
## 1   
## [,1] [,2] [,3] [,4] [,5] [,6]   
## taxa.list "Mammals" "Plants" "Reptiles" "Amphibians" "Birds" "Fish"  
## taxa.count 18 33 7 4 40 15   
## [,7] [,8]   
## taxa.list "Invertebrates" "Other"  
## taxa.count 29 3   
## TRUE   
## 12

Flow frequency (study level, n=131)

desired\_length <- 10   
empty\_list <- vector(mode = 'list', length = desired\_length)  
  
tc <- s2$tele\_cat  
tc.list <-c("Energy Transfer","Investment","Knowledge Transfer",  
 "Migration (human)","Migration (non-human)",  
 "Species Dispersal","Tourism","Trade","Waste Transfer","Water Transfer")  
tc.count <- empty\_list  
  
for (i in 1:length(tc.list)) {  
   
 tc.count[i] <- sum(str\_count(tc, tc.list[i]))  
 }  
  
tc.count <- rbind(tc.list,tc.count)  
tc.count  
  
  
# Can't get loop to count the migration values for some reason, manual input code below  
# Migration (human) n=4 # Migration (non-human) n=0  
tc.table <- table(s2$tele\_cat)  
  
tc.count[2,4] <- tc.table[5] + tc.table[13] # Migration (human) n=6 # Migration (non-human) n-7  
  
tc.count  
  
#gather(tc.count)  
  
multiple <- str\_count(tc, ';')  
multiple <- multiple != 0  
multiple <- table(multiple) ["TRUE"]  
multiple

## [,1] [,2] [,3]   
## tc.list "Energy Transfer" "Investment" "Knowledge Transfer"  
## tc.count 9 8 13   
## [,4] [,5] [,6]   
## tc.list "Migration (human)" "Migration (non-human)" "Species Dispersal"  
## tc.count 0 0 17   
## [,7] [,8] [,9] [,10]   
## tc.list "Tourism" "Trade" "Waste Transfer" "Water Transfer"  
## tc.count 57 34 2 4   
## [,1] [,2] [,3]   
## tc.list "Energy Transfer" "Investment" "Knowledge Transfer"  
## tc.count 9 8 13   
## [,4] [,5] [,6]   
## tc.list "Migration (human)" "Migration (non-human)" "Species Dispersal"  
## tc.count 4 0 17   
## [,7] [,8] [,9] [,10]   
## tc.list "Tourism" "Trade" "Waste Transfer" "Water Transfer"  
## tc.count 57 34 2 4   
## TRUE   
## 16

Number of metrics reported by each paper

# get summary  
 ddply(survey3, .(paper\_id), summarize,  
 # total count of entries   
 num\_entries=length(entry\_id),  
 perc\_dataset=length(entry\_id)/nrow(survey3),  
 num\_taxa=length(unique(taxa))) %>%   
 arrange(desc(perc\_dataset))  
  
num.metric <- count(survey3$paper\_id)  
range(num.metric$freq)  
sum(num.metric$freq)

## paper\_id num\_entries perc\_dataset num\_taxa  
## 1 84 162 0.221614227 1  
## 2 4060 57 0.077975376 1  
## 3 1551 27 0.036935705 1  
## 4 3772 22 0.030095759 1  
## 5 6502 21 0.028727770 1  
## 6 134 20 0.027359781 1  
## 7 4155 20 0.027359781 1  
## 8 4552 15 0.020519836 1  
## 9 479 14 0.019151847 1  
## 10 3112 13 0.017783858 1  
## 11 3989 12 0.016415869 1  
## 12 4831 11 0.015047880 1  
## 13 1007 10 0.013679891 1  
## 14 2606 9 0.012311902 1  
## 15 5844 8 0.010943912 1  
## 16 6254 8 0.010943912 1  
## 17 1376 7 0.009575923 2  
## 18 5756 7 0.009575923 1  
## 19 6948 7 0.009575923 1  
## 20 227 6 0.008207934 1  
## 21 264 6 0.008207934 1  
## 22 1893 6 0.008207934 2  
## 23 3122 6 0.008207934 1  
## 24 3485 6 0.008207934 1  
## 25 3849 6 0.008207934 1  
## 26 4551 6 0.008207934 1  
## 27 5076 6 0.008207934 1  
## 28 5905 6 0.008207934 1  
## 29 6511 6 0.008207934 1  
## 30 357 5 0.006839945 1  
## 31 1366 5 0.006839945 1  
## 32 1743 5 0.006839945 1  
## 33 2050 5 0.006839945 1  
## 34 2642 5 0.006839945 1  
## 35 5926 5 0.006839945 1  
## 36 6396 5 0.006839945 1  
## 37 292 4 0.005471956 1  
## 38 302 4 0.005471956 1  
## 39 471 4 0.005471956 1  
## 40 826 4 0.005471956 1  
## 41 1299 4 0.005471956 1  
## 42 1590 4 0.005471956 1  
## 43 1616 4 0.005471956 1  
## 44 2401 4 0.005471956 1  
## 45 2546 4 0.005471956 1  
## 46 2998 4 0.005471956 1  
## 47 3484 4 0.005471956 1  
## 48 4020 4 0.005471956 1  
## 49 4063 4 0.005471956 1  
## 50 4257 4 0.005471956 1  
## 51 4293 4 0.005471956 1  
## 52 5746 4 0.005471956 1  
## 53 379 3 0.004103967 1  
## 54 483 3 0.004103967 1  
## 55 518 3 0.004103967 1  
## 56 1270 3 0.004103967 1  
## 57 2345 3 0.004103967 1  
## 58 3247 3 0.004103967 1  
## 59 3361 3 0.004103967 1  
## 60 4209 3 0.004103967 1  
## 61 4445 3 0.004103967 1  
## 62 5063 3 0.004103967 1  
## 63 6321 3 0.004103967 2  
## 64 80 2 0.002735978 2  
## 65 339 2 0.002735978 2  
## 66 396 2 0.002735978 1  
## 67 425 2 0.002735978 1  
## 68 466 2 0.002735978 1  
## 69 510 2 0.002735978 1  
## 70 708 2 0.002735978 1  
## 71 713 2 0.002735978 1  
## 72 1565 2 0.002735978 1  
## 73 1799 2 0.002735978 1  
## 74 2201 2 0.002735978 1  
## 75 2670 2 0.002735978 1  
## 76 3081 2 0.002735978 1  
## 77 3151 2 0.002735978 1  
## 78 3165 2 0.002735978 1  
## 79 3511 2 0.002735978 1  
## 80 3583 2 0.002735978 1  
## 81 3697 2 0.002735978 1  
## 82 4760 2 0.002735978 1  
## 83 4832 2 0.002735978 1  
## 84 5175 2 0.002735978 1  
## 85 5875 2 0.002735978 1  
## 86 5966 2 0.002735978 1  
## 87 6331 2 0.002735978 1  
## 88 6845 2 0.002735978 1  
## 89 17 1 0.001367989 1  
## 90 44 1 0.001367989 1  
## 91 45 1 0.001367989 1  
## 92 311 1 0.001367989 1  
## 93 523 1 0.001367989 1  
## 94 654 1 0.001367989 1  
## 95 813 1 0.001367989 1  
## 96 897 1 0.001367989 1  
## 97 992 1 0.001367989 1  
## 98 1040 1 0.001367989 1  
## 99 1349 1 0.001367989 1  
## 100 1368 1 0.001367989 1  
## 101 1681 1 0.001367989 1  
## 102 1821 1 0.001367989 1  
## 103 2222 1 0.001367989 1  
## 104 2253 1 0.001367989 1  
## 105 2284 1 0.001367989 1  
## 106 3243 1 0.001367989 1  
## 107 3668 1 0.001367989 1  
## 108 3868 1 0.001367989 1  
## 109 4087 1 0.001367989 1  
## 110 4393 1 0.001367989 1  
## 111 4509 1 0.001367989 1  
## 112 4518 1 0.001367989 1  
## 113 4574 1 0.001367989 1  
## 114 4850 1 0.001367989 1  
## 115 5125 1 0.001367989 1  
## 116 5309 1 0.001367989 1  
## 117 5314 1 0.001367989 1  
## 118 5477 1 0.001367989 1  
## 119 5486 1 0.001367989 1  
## 120 5577 1 0.001367989 1  
## 121 5773 1 0.001367989 1  
## 122 5782 1 0.001367989 1  
## 123 5804 1 0.001367989 1  
## 124 5853 1 0.001367989 1  
## 125 6195 1 0.001367989 1  
## 126 6820 1 0.001367989 1  
## 127 3790002 1 0.001367989 1  
## [1] 1 162  
## [1] 731

Number of metrics by taxa and paper

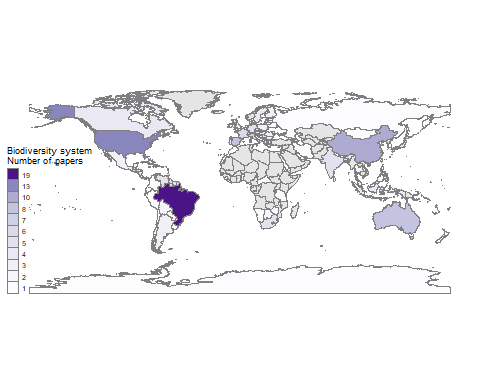
# get summary  
 a <- ddply(survey3, .(taxa), summarize,  
 # total count of entries   
 num\_entries=length(entry\_id),  
 perc\_dataset=length(entry\_id)/nrow(survey3),  
 num\_papers=length(unique(paper\_id)),  
 perc\_papers=length(unique(paper\_id))/length(unique(survey3$paper\_id))) %>%   
 arrange(desc(perc\_dataset))  
  
# view  
 a

## taxa num\_entries perc\_dataset num\_papers perc\_papers  
## 1 Birds 340 0.465116279 36 0.283464567  
## 2 Invertebrates 159 0.217510260 28 0.220472441  
## 3 Plants/Trees/Shrubs 85 0.116279070 30 0.236220472  
## 4 Fish 75 0.102599179 15 0.118110236  
## 5 Mammals 45 0.061559508 13 0.102362205  
## 6 Multiple 17 0.023255814 5 0.039370079  
## 7 Reptiles 9 0.012311902 4 0.031496063  
## 8 Amphibians 1 0.001367989 1 0.007874016

## 4.3 Biodiversity and Flow Categories by Country

Biodiversity country map

shp <- ne\_countries(scale = 50, returnclass = 'sf') %>% #st\_as\_sf() %>%  
 select(name, iso\_a3) %>% ## , economy, income\_grp  
 # filter(name != 'Antarctica') %>%  
 # filter(name == 'France') %>%  
 dplyr::mutate(  
 iso\_a3 = if\_else(name == 'France', 'FRA', iso\_a3),  
 iso\_a3 = if\_else(name == 'Norway', 'NOR', iso\_a3))  
# plot(shp[1])  
  
shp %>% filter(name == 'Norway')  
  
### add new country name that are consistant with names uned in SDG data  
# library(readxl)  
# xls.shp.info <- paste0(dir, '/update\_0503\_SUM\_dist/\_input\_data/ne\_10m\_admin\_0\_countries/Table/ne\_10m\_admin\_0\_countries-Export\_Output.xls')  
#  
# nation\_new\_name <-  
# read\_excel(path = xls.shp.info,  
# sheet = "dat", col\_names = T) %>%  
# select(ADMIN, ADM0\_A3, nation\_name)  
# names(nation\_new\_name)  
# nation\_new\_name <- as.data.frame(nation\_new\_name[,-c(3,4)])  
  
  
  
  
### theme, font --------------------------------------------------------------------------------------  
font <- 'sans' ## = "TT Arial"  
font\_size <- 6.5 ## max = 7; min = 5 for Nature Sustainability  
map\_theme <- ggpubr::theme\_transparent()+  
 theme(  
 axis.title.x=element\_blank(),  
 axis.title.y=element\_blank(),  
 axis.text.x=element\_blank(),  
 axis.ticks = element\_blank(),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 panel.border = element\_blank(),  
 legend.position = c(0.09, 0.38),  
 legend.key.size = unit(0.3, "cm"),  
 # legend.key.height = unit(0.5, "cm"),  
 legend.key = element\_rect(fill = NA, colour = NA, size = 0.25),  
   
 panel.background = element\_rect(fill = "transparent", colour = NA),  
 plot.background = element\_rect(fill = "transparent", colour = NA),  
 legend.background = element\_rect(fill = "transparent", colour = NA),  
 legend.box.background = element\_rect(fill = "transparent", colour = NA),  
 text=element\_text(family=font, size=font\_size))  
  
  
### - Read in Data  
  
df99 <- read.csv(paste0(dat.dir, 'survey2\_cleaned.csv'))  
  
  
### - list\_countries  
ctr\_sys <- df99 %>%  
 select(paper\_id, list\_countries) %>%  
 data.frame(., do.call(rbind, str\_split(.$list\_countries,';'))) %>%  
 gather(key = 'key', value = list\_countries, 3:ncol(.)) %>%  
 dplyr::mutate(list\_countries = trimws(list\_countries)) %>%  
 dplyr::distinct(paper\_id, list\_countries, .keep\_all = T) %>%  
 filter(list\_countries != '') %>%  
 arrange(paper\_id) %>%  
 group\_by(list\_countries) %>%  
 tally()%>%  
 arrange(desc(n)) #%>%  
# str(ctr\_sys)  
  
  
### join new name table  
shp\_df <- shp %>%  
 left\_join(., ctr\_sys, by=c("iso\_a3" = "list\_countries"))  
  
### check the join  
shp\_df\_check <- merge(  
 shp, ctr\_sys, by.x = "iso\_a3", by.y = "list\_countries", all.y=T) %>%  
 st\_drop\_geometry()  
  
# str(shp\_df)  
unique(shp\_df$n)  
  
  
### breaks and colors  
var <- shp\_df$n  
max <- max(var, na.rm = T); max  
min <- min(var, na.rm = T); min  
  
breaks <- seq(min, max, 2); breaks; length(breaks)  
breaks <- sort(unique(shp\_df$n)); breaks; length(breaks)  
myPalette <- colorRampPalette((brewer.pal(8, "Greens")));# myPalette; rev  
colors <- myPalette(length(breaks)); colors  
title <- 'Entire system\nNumber of papers'  
  
  
### - biodiv\_countries  
ctr\_bio <- df99 %>%  
 select(paper\_id, biodiv\_countries) %>%  
 data.frame(., do.call(rbind, str\_split(.$biodiv\_countries,';'))) %>%  
 gather(key = 'key', value = biodiv\_countries, 3:ncol(.)) %>%  
 dplyr::mutate(biodiv\_countries = trimws(biodiv\_countries)) %>%  
 dplyr::distinct(paper\_id, biodiv\_countries, .keep\_all = T) %>%  
 filter(biodiv\_countries != '') %>%  
 arrange(paper\_id) %>%  
 group\_by(biodiv\_countries) %>%  
 tally()%>%  
 arrange(desc(n)) #%>%  
# str(ctr\_bio)  
  
  
### join new name table  
shp\_df <- shp %>%  
 left\_join(., ctr\_bio, by=c("iso\_a3" = "biodiv\_countries"))  
  
# str(shp\_df)  
unique(shp\_df$n)  
  
### check the join  
shp\_df\_check <- shp %>%  
 right\_join(., ctr\_bio, by=c("iso\_a3" = "biodiv\_countries"))  
  
### breaks and colors  
var <- shp\_df$n  
max <- max(var, na.rm = T); max  
min <- min(var, na.rm = T); min  
  
breaks <- seq(min, max, 2); breaks; length(breaks)  
breaks <- sort(unique(shp\_df$n)); breaks; length(breaks)  
myPalette <- colorRampPalette((brewer.pal(8, "Purples")));# myPalette; rev  
colors <- myPalette(length(breaks)); colors  
title <- 'Biodiversity system\nNumber of papers'  
  
### plot  
fig2 <- ggplot(data = shp\_df) +  
 geom\_sf(aes(fill = n), color='gray50', size=0.01) +  
 scale\_fill\_gradientn(  
 name = title,  
 colours= colors, na.value = "gray90",  
 limits = c(min,max), breaks = breaks) +  
 guides(fill = guide\_legend(label.hjust = 0, label = T,  
 reverse = T, title = title))+  
 map\_theme  
fig2



dir  
fname <- paste0(fig.dir, 'Fig\_map\_country\_bio.jpg'); fname

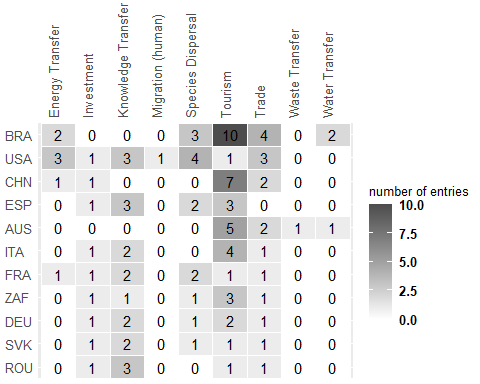
## Simple feature collection with 1 feature and 2 fields  
## geometry type: MULTIPOLYGON  
## dimension: XY  
## bbox: xmin: -9.098877 ymin: 58.02095 xmax: 33.6293 ymax: 80.47783  
## CRS: +proj=longlat +datum=WGS84 +no\_defs +ellps=WGS84 +towgs84=0,0,0  
## name iso\_a3 geometry  
## 1 Norway NOR MULTIPOLYGON (((5.08584 60....  
## [1] NA 3 2 9 4 1 18 11 5 8 6 7 13  
## [1] 18  
## [1] 1  
## [1] 1 3 5 7 9 11 13 15 17  
## [1] 9  
## [1] 1 2 3 4 5 6 7 8 9 11 13 18  
## [1] 12  
## [1] "#F7FCF5" "#EBF7E7" "#DCF1D7" "#C9EAC2" "#B2E0AB" "#98D594" "#7CC77C"  
## [8] "#5CB86A" "#3EA85A" "#2B934B" "#16793E" "#005A32"  
## [1] NA 1 3 8 2 19 4 10 5 6 7 13  
## [1] 19  
## [1] 1  
## [1] 1 3 5 7 9 11 13 15 17 19  
## [1] 10  
## [1] 1 2 3 4 5 6 7 8 10 13 19  
## [1] 11  
## [1] "#FCFBFD" "#F2F1F7" "#E6E5F1" "#D6D7E9" "#C2C2DF" "#ADABD2" "#9794C5"  
## [8] "#827FBB" "#7262AC" "#603E9A" "#4A1486"  
## [1] "C:/Users/Ciara Hovis/Desktop/GradSchool/Co-author papers/MetaBD/MCBD\_Root"  
## [1] "C:/Users/Ciara Hovis/Desktop/GradSchool/Co-author papers/MetaBD/MCBD\_Root/images/Fig\_map\_country\_bio.jpg"

Continent distribution, study level (n=131)

table(ctr\_bio$biodiv\_countries)  
  
# install.packages("jsonlite", repos="https://cran.rstudio.com/")  
library("jsonlite")  
json\_file <- 'https://datahub.io/JohnSnowLabs/country-and-continent-codes-list/datapackage.json'  
json\_data <- fromJSON(paste(readLines(json\_file), collapse=""))  
# get list of all resources:  
print(json\_data$resources$name)  
  
# print all tabular data(if exists any) to 'data'  
for(i in 1:length(json\_data$resources$datahub$type)){  
 if(json\_data$resources$datahub$type[i]=='derived/csv'){  
 path\_to\_file = json\_data$resources$path[i]  
 data <- read.csv(url(path\_to\_file))  
 print(data)  
 }  
}  
  
cont <- cbind(data$Continent\_Name,data$Country\_Name,data$Three\_Letter\_Country\_Code)  
cont <- as.data.frame(cont)  
  
# Match continent and country name to biodiversity countries  
names(cont)[1:3] <- c("continent","ctr\_name","ctr\_code")  
  
ctr\_bio$continent <- cont$continent[match(ctr\_bio$biodiv\_countries,cont$ctr\_code)] # add continent name  
ctr\_bio$ctr\_name <- cont$ctr\_name[match(ctr\_bio$biodiv\_countries,cont$ctr\_code)] # add country name  
  
  
BUC <- aggregate(x = ctr\_bio$n,  
 by = list(ctr\_bio$continent),  
 FUN = sum)  
zz <- ctr\_bio[-which(ctr\_bio$biodiv\_countries == 'BRA' |  
 ctr\_bio$biodiv\_countries == 'USA' |  
 ctr\_bio$biodiv\_countries == 'CHN'),]  
BUC.no <- aggregate(x = zz$n,  
 by = list(zz$continent),  
 FUN = sum)

heat map table of countries and flow type

## Percentage of telecoupling  
  
k <- as.data.frame(cbind(df99$biodiv\_countries, df99$tele\_cat))  
colnames(k) <- c('ctr', 'tc')  
  
ctr\_ls <- ctr\_bio[1]  
  
k$ctr[k$ctr == ""] <- NA  
  
k$tc <- as.character(k$tc)  
  
poo <- separate\_rows(k, tc, sep=";", convert = TRUE)  
doo <-separate\_rows(poo, ctr, sep=";", convert = TRUE)  
  
  
woo <- paste0(doo$ctr, '\_', doo$tc)  
woot <- as.data.frame(table(woo))  
  
ok <- str\_split\_fixed(woot$woo, '\_', 2)  
woot$ctr <- ok[,1]  
woot$tc <- ok[,2]  
  
  
### count by ctr and tc  
booty <- doo %>%  
 dplyr::select(ctr, tc) %>%  
 ### how many tcs in total for each ctr?  
 ungroup() %>%  
 group\_by(ctr) %>%  
 add\_tally(name = 'total\_tc') %>%  
 ### the number of each tc  
 dplyr::ungroup() %>%  
 dplyr::group\_by(ctr, total\_tc) %>%  
 dplyr::count(tc, name = 'n\_tc') %>%  
 arrange(ctr) %>%  
 dplyr::mutate(pct\_tc = n\_tc/total\_tc\*100)  
  
  
booty\_check <- booty %>%  
 dplyr::group\_by(ctr) %>%  
 dplyr::summarise(sum = sum(pct\_tc, na.rm = T))  
  
booty2 <- booty[order(-booty$total\_tc,booty$ctr, -booty$n\_tc),]  
  
# Load data of number of studies by taxa and flow  
nr\_ctr <- aggregate(x = booty2$n\_tc,  
 by = list(booty2$ctr,booty2$tc),  
 FUN = sum)  
  
  
  
# Rename columns  
  
names(nr\_ctr)[1:3] <- c("ctr","tc","count")  
  
# Fill the gaps  
  
mat\_nr\_ctr <-dcast(nr\_ctr, ctr ~ tc, value.var = 'count')  
filled\_nr\_ctr <- melt(mat\_nr\_ctr, id=c("ctr"))  
filled\_nr\_ctr[is.na(filled\_nr\_ctr)] <- 0  
names(filled\_nr\_ctr)[2:3] <- c("tc", "count")  
  
# Subset for countries >5 total tc  
  
filled\_nr\_ctr$n\_tc <- booty2$total\_tc[match(filled\_nr\_ctr$ctr, booty2$ctr)]  
filled\_nr\_ctr<- filled\_nr\_ctr[which(filled\_nr\_ctr$n\_tc > 5),]  
filled\_nr\_ctr<-filled\_nr\_ctr[order(-filled\_nr\_ctr$n\_tc),]  
  
  
# Change order of factor levels for consistency with other figures  
  
filled\_nr\_ctr$ctr <- factor(filled\_nr\_ctr$ctr,levels=c(unique(filled\_nr\_ctr$ctr[order(filled\_nr\_ctr$n\_tc)])))  
  
filled\_nr\_ctr$tc <- factor(filled\_nr\_ctr$tc,levels=c("Energy Transfer","Investment","Knowledge Transfer",  
 "Migration (human)","Migration (non-human)","Species Dispersal",  
 "Tourism","Trade","Waste Transfer","Water Transfer"))  
  
   
# Heatmap plot  
  
  
heatmap\_ctr <- ggplot(filled\_nr\_ctr,  
 aes(x=tc, y=ctr, fill=count)) +  
 geom\_tile(color = "white", size=0.25) +  
 geom\_text(aes(label = count )) +  
 labs(x="",y="")+  
 scale\_y\_discrete(expand=c(0,0))+  
 theme\_grey(base\_size=8)+  
 scale\_fill\_gradientn(colours = c('#FFFFFF','gray30')) +  
 theme(legend.title = element\_text(size = 10),  
 legend.text = element\_text(size = 10, face="bold"),  
 plot.background=element\_blank(),  
 plot.title = element\_blank(),  
 axis.title= element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.text.x.top = element\_text(size=10, angle = 90, vjust=0,  
 hjust = 0),  
 axis.text.y.left = element\_text(size=10, angle = 360, vjust=0.5,  
 hjust = 0),  
 strip.background = element\_rect(fill="white"),  
 panel.spacing = unit(0.1, "lines")) +  
 scale\_x\_discrete(position = "top") +  
 labs(fill = "number of entries")  
  
heatmap\_ctr

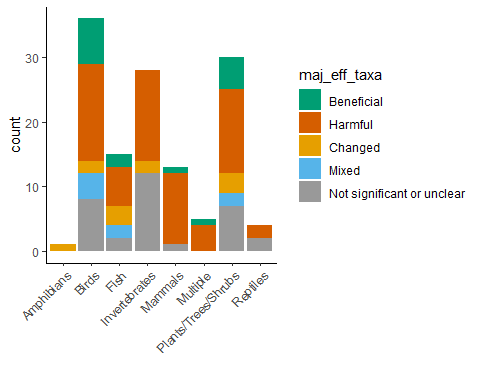


# 5. Telecoupling Impact by Taxa

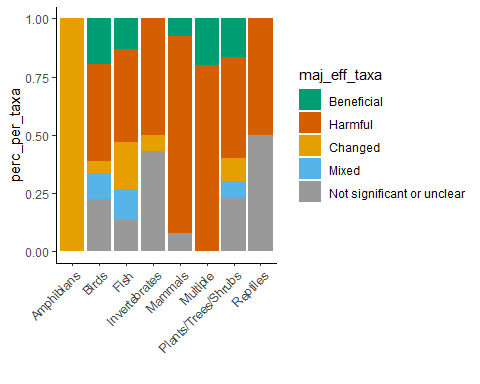
Note that n=132, some papers reported metrics for multiple taxa. Maj\_eff was determined based on paper and taxa for these figures.

* When studies reported significant results, the majority were significantly Harmful for biodiversity.
* Studies that quantified impacts to mammals were overwhelmingly Harmful compared to the other groups.
* Birds and plants had more beneficial impacts, relative to the other groups.
* Amphibians were greatly underrepresented. Only one paper measured amphibian diversity separate from other groups.

# get summary  
 a <- ddply(taxEff\_maj, .(taxa, maj\_eff\_taxa), summarize,  
 # total count of entries   
 count=length(taxa)) %>%  
 # percents of effects within totals per taxa  
 group\_by(taxa) %>%  
 nest() %>%   
 mutate(perc\_per\_taxa=map(data, function(x) x$count/sum(x$count))) %>%   
 unnest(cols = c(data, perc\_per\_taxa))  
# save as csv  
# write.csv(a, paste0(tab.dir,'survey3\_summary\_effects\_by\_taxa.csv'),row.names = TRUE)  
  
# Stack plot (count)  
 quick\_plot <- ggplot(a, aes(fill=maj\_eff\_taxa, y=count, x=taxa)) +   
 geom\_bar(position="stack", stat="identity") +  
 theme\_classic() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_fill\_manual(values=majeff\_col) +  
 xlab("")  
quick\_plot



# Stack plot (percentage)  
 quick\_plot <- ggplot(a, aes(fill=maj\_eff\_taxa, y=perc\_per\_taxa, x=taxa)) +   
 geom\_bar(position="stack", stat="identity") +  
 theme\_classic() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_fill\_manual(values=majeff\_col) +  
 xlab("")  
quick\_plot

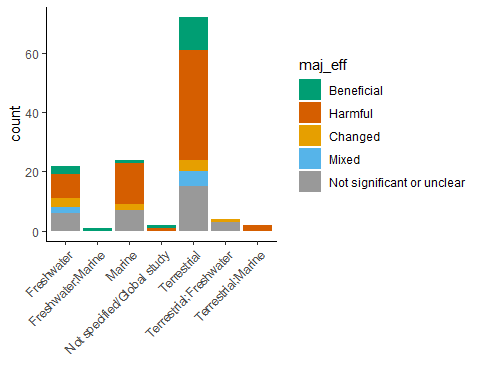


# 6. Telecoupling Impact by Habitat

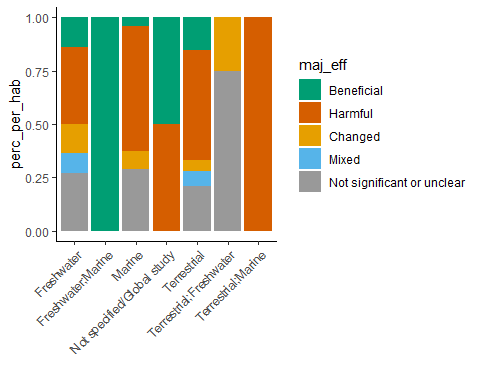
Figure moving to supp.

* Terrestrial habitats were studied the most.
* All habitats had more Harmful impacts on their respective biodiversity metrics.

# get summary  
 a <- ddply(s23\_maj, .(hab, maj\_eff), summarize,  
 # total count of entries   
 count=length(hab)) %>%  
 # percents of effects within totals per hab  
 group\_by(hab) %>%  
 nest() %>%   
 mutate(perc\_per\_hab=map(data, function(x) x$count/sum(x$count))) %>%   
 unnest(cols = c(data, perc\_per\_hab))  
# save as csv  
# write.csv(a, paste0(tab.dir,'survey3\_summary\_effects\_by\_hab.csv'), row.names = TRUE)  
  
# Stack plot (count)  
 quick\_plot <- ggplot(a, aes(fill=maj\_eff, y=count, x=hab)) +   
 geom\_bar(position="stack", stat="identity") +  
 theme\_classic() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_fill\_manual(values=majeff\_col) +  
 xlab("")  
  
# view  
 quick\_plot



# Stack plot (percent)  
 quick\_plot <- ggplot(a, aes(fill=maj\_eff, y=perc\_per\_hab, x=hab)) +   
 geom\_bar(position="stack", stat="identity") +  
 theme\_classic() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_fill\_manual(values=majeff\_col) +  
 xlab("")  
quick\_plot



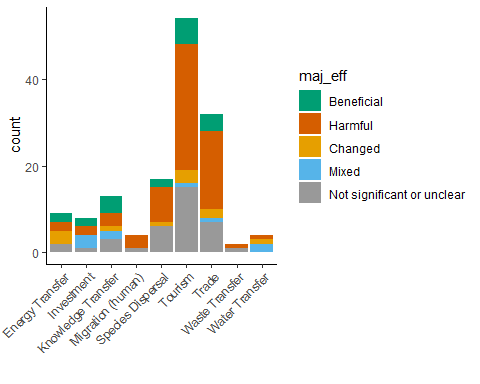
# 7. Telecoupling Impact by Flow

Duplicate rows with multiple flow types. This means that effects and other results will be *duplicated* as well.

The reason for doing this is so that we can attribute biodiversity impacts to the specific flows that may have been studied together, and whose individual impacts cannot be decoupled.

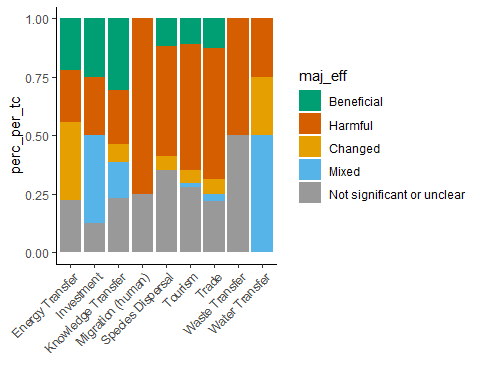
* Tourism and Trade were the most represented in our sample and both had majority Harmful imapcts reported.
* The flow categories that showed somewhat beneficial impacts where Investment, Water Transfer, and Knowledge Transfer.

# set as character  
 s23\_maj$tc <- as.character(s23\_maj$tc)  
  
# show original length of rows  
 print(paste('Original number of rows:',nrow(s23\_maj)))  
  
# split multiple flows listed in a row into other new rows  
 tcRep\_maj <- separate\_rows(s23\_maj, tc, sep=";", convert = TRUE)  
 tcRep\_maj$tc <- as.factor(tcRep\_maj$tc)  
  
# show new length of rows  
 print(paste('Number of rows after separating multiple flows per paper:',nrow(tcRep\_maj)))  
  
# show levels  
# summary(tcRep$tc)  
  
# get summary  
 a <- ddply(tcRep\_maj, .(tc, maj\_eff), summarize,  
 # total count of entries   
 count=length(tc)) %>%  
 # percents of effects within totals per tc  
 group\_by(tc) %>%  
 nest() %>%   
 mutate(perc\_per\_tc=map(data, function(x) x$count/sum(x$count))) %>%   
 unnest(cols = c(data, perc\_per\_tc))  
# save as csv  
# write.csv(a, paste0(tab.dir,'survey3\_summary\_effects\_by\_flow.csv'),row.names = TRUE)  
   
# Stack plot (count)  
 quick\_plot <- ggplot(a, aes(fill=maj\_eff, y=count, x=tc)) +   
 geom\_bar(position="stack", stat="identity") +  
 theme\_classic() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_fill\_manual(values=majeff\_col) +  
 xlab("")  
  
  
# view  
 quick\_plot



## [1] "Original number of rows: 127"  
## [1] "Number of rows after separating multiple flows per paper: 143"

# Stack plot (percent)  
 quick\_plot <- ggplot(a, aes(fill=maj\_eff, y=perc\_per\_tc, x=tc)) +   
 geom\_bar(position="stack", stat="identity") +  
 theme\_classic() +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1)) +  
 scale\_fill\_manual(values=majeff\_col) +  
 xlab("")  
  
quick\_plot



# 8. Telecoupling impacts by at the entry level

##Replicate TC flows by entry

* This figure shows the direction of telecoupling impacts at the entry level (i.e. all 700+ metrics recorded)
* Most studies report multiple species richness and abundance.
* Most habitat metrics were significantly Harmful, except in the case of knowledge transfer telecouplings.

# set as character  
 s23$tc <- as.character(s23$tc)  
 poo<-s23[is.na(s23$tc),]  
   
   
# show original length of rows  
 print(paste('Original number of rows:',nrow(s23)))  
  
# split multiple flows listed in a row into other new rows  
 tcRep <- separate\_rows(s23, tc, sep=";", convert = TRUE)  
 tcRep$tc <- as.factor(tcRep$tc)  
  
# show new length of rows  
 print(paste('Number of rows after separating multiple flows per paper:',nrow(tcRep)))  
   
 tcRep\_maj <- tcRep[!duplicated(tcRep$paper\_id),]

## [1] "Original number of rows: 731"  
## [1] "Number of rows after separating multiple flows per paper: 771"

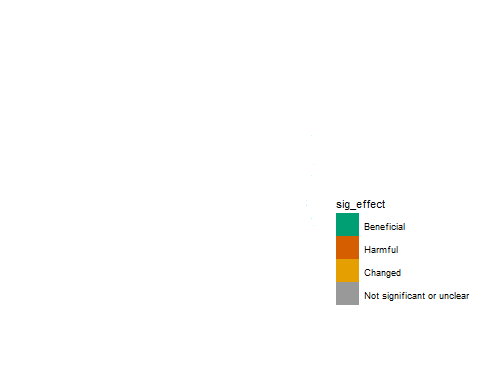
Convert to long format, where metric is indicated.

# subset  
 metrics <- subset(tcRep,  
 select=c('paper\_id','entry\_id',  
 'taxa','tc','sig\_effect',  
 'biodiv\_cat\_1sp',  
 'biodiv\_cat\_multsp',  
 'biodiv\_cat\_habitat'))  
# melt  
 metrics\_long <- melt(metrics, id.vars=c('paper\_id','entry\_id',  
 'taxa','tc','sig\_effect'))  
   
# remove NA fields  
 metrics\_long <- metrics\_long[!is.na(metrics\_long$value),]  
  
# change colnames  
 colnames(metrics\_long)[6] <- 'metric\_type'  
 colnames(metrics\_long)[7] <- 'metric'  
  
  
# change levels  
 levels(metrics\_long$metric\_type) <- c('Single species',  
 'Multiple species',  
 'Habitat')  
  
 levels(as.factor(metrics\_long$metric))  
  
# remove parentheses  
 metrics\_long$metric <- gsub("\\s\*\\([^\\)]+\\)", '',  
 as.character(metrics\_long$metric))  
  
# capitalize each word  
 metrics\_long$metric <- tools::toTitleCase(metrics\_long$metric)  
   
# adjust name  
 metrics\_long$metric <- gsub("With-in", 'Within',  
 as.character(metrics\_long$metric))  
   
# change levels, moving other to the end  
 metrics\_long$metric <- factor(metrics\_long$metric,  
 c("Abundance/Density",  
 "Amount",  
 "Composition",  
 "Detection",  
 "Diversity Index",  
 "Evenness",  
 "Movement",  
 "Occurrence",  
 "Population Dynamics",  
 "Quality",  
 "Richness",  
 "Within Species Diversity"  
 ))  
   
# check levels  
 levels(as.factor(metrics\_long$metric))  
  
# get summary  
 metric\_flow <- ddply(metrics\_long,  
 .(metric\_type, metric, tc, sig\_effect),  
 summarize,  
 # total count of entries   
 count=length(metric))  
  
# save as csv  
# write.csv(metric\_flow, paste0(tab.dir,'synthesis\_sig\_effects\_metric\_and\_flow.csv'),row.names = TRUE)  
  
# sig\_effects level change  
 metric\_flow$sig\_effect <- gsub("\\s\*\\([^\\)]+\\)", '',  
 as.character(metric\_flow$sig\_effect))  
 metric\_flow$sig\_effect <- factor(metric\_flow$sig\_effect,  
 c( "Beneficial","Harmful","Changed","Not significant or unclear"))  
# telecoupling level change  
 metric\_flow$tc <- factor(metric\_flow$tc,  
 c("Energy Transfer","Investment","Knowledge Transfer",  
 "Migration (human)","Migration (non-human)",  
 "Species Dispersal","Tourism","Trade","Waste Transfer","Water Transfer"  
 ))

## [1] "Abundance/Density (biomass, mass, number of individuals, individuals/unit area)"  
## [2] "Abundance/Density (number of individuals, individuals/unit area, biomass)"   
## [3] "Amount (e.g. land use change from non-habitat to habitat)"   
## [4] "Composition (what species?)"   
## [5] "Diversity index (Shannon-Weiner, Simpson's, Inverse Simpson's, etc.)"   
## [6] "Evenness"   
## [7] "Movement"   
## [8] "Occurrence"   
## [9] "Occurrence (presence, range, persistence, etc., NOT detection)"   
## [10] "Population dynamics (survival, fitness, reproduction, mortality, etc.)"   
## [11] "Quality (pollution, connectence, disturbance, etc.)"   
## [12] "Richness (number of species)"   
## [13] "With-in species diversity (genetic diversity, age structure, etc.)"   
## [1] "Abundance/Density" "Amount"   
## [3] "Composition" "Detection"   
## [5] "Diversity Index" "Evenness"   
## [7] "Movement" "Occurrence"   
## [9] "Population Dynamics" "Quality"   
## [11] "Richness" "Within Species Diversity"

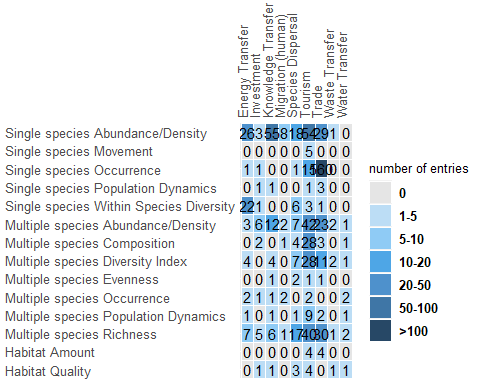
Metric & Flow Figure code

# Matrix of pie charts  
sig\_effect\_cols <- sigeff\_col  
  
 metric\_flow\_fig <- ggplot(metric\_flow) +   
 aes(x='', y = count, fill = sig\_effect) +  
 geom\_bar(position='fill',  
 stat='identity',  
 width = 1,  
 size = 0.01) +  
 # activate for vertical or horizontal version  
 facet\_grid(metric\_type ~ metric ~ tc, switch='y') + #horizontal  
 #facet\_grid(tc ~ metric, switch='y') + #vertical  
 # convert to pie chart  
 coord\_polar(theta = 'y') +  
 # colors  
 scale\_fill\_manual(  
 values=sig\_effect\_cols,  
 guide = guide\_legend(reverse = FALSE)) +   
 scale\_y\_discrete(position='right',expand=c(0,0))+  
 # style  
 theme\_gray(base\_size=8)+  
 theme(  
 axis.title=element\_blank(),  
 axis.line=element\_blank(),  
 axis.ticks=element\_blank(),  
 axis.text=element\_blank(),  
 panel.border = element\_blank(),  
 panel.grid=element\_blank(),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 plot.background = element\_rect(),  
 plot.title=element\_blank(),  
 strip.text.x = element\_text(size=10,  
 angle=90, vjust=0, hjust = 0),  
 strip.text.y.left = element\_text(size=10,  
 angle=0, vjust=0, hjust = 0),  
 strip.background = element\_rect(fill='white'),  
 panel.spacing = unit(0.1, 'lines')  
 ) +  
 # labels  
 ylab('') + xlab('')  
   
  
  
  
# display image  
 metric\_flow\_fig



Metric & Flow Heatmap

# Load data of number of studies by taxa and flow  
nr\_metric\_tele <- aggregate(x = metric\_flow$count,  
 by = list(metric\_flow$metric\_type,metric\_flow$metric, metric\_flow$tc),  
 FUN = sum)  
  
   
  
# Rename columns  
  
names(nr\_metric\_tele)[1:4] <- c("type","metric", "telecoupling", "count")  
  
nr\_metric\_tele$metricType <- paste(nr\_metric\_tele$type,nr\_metric\_tele$metric)   
  
# Fill the gaps  
  
mat\_nr\_metric\_tele <-dcast(nr\_metric\_tele, metricType ~ telecoupling, value.var = 'count')  
filled\_nr\_metric\_tele <- melt(mat\_nr\_metric\_tele, id=c("metricType"))  
filled\_nr\_metric\_tele[is.na(filled\_nr\_metric\_tele)] <- 0  
names(filled\_nr\_metric\_tele)[2:3] <- c("telecoupling", "count")  
  
# Create 'count' categories for representation  
  
filled\_nr\_metric\_tele$count\_factor <- cut(filled\_nr\_metric\_tele$count,  
 breaks = c(-1,0,5,10,20,50,100,max(filled\_nr\_metric\_tele$count,na.rm=T)),  
 labels=c("0","1-5","5-10","10-20","20-50","50-100",">100"))  
  
  
# Change order of factor levels for consistency with other figures  
  
filled\_nr\_metric\_tele$metricType <- as.character(filled\_nr\_metric\_tele$metricType)  
  
filled\_nr\_metric\_tele$metricType <- factor(filled\_nr\_metric\_tele$metricType,  
 levels=c("Habitat Quality",  
 "Habitat Amount",  
 "Multiple species Richness",  
 "Multiple species Population Dynamics",  
 "Multiple species Occurrence",  
 "Multiple species Evenness",  
 "Multiple species Diversity Index",  
 "Multiple species Composition",  
 "Multiple species Abundance/Density",  
 "Single species Within Species Diversity",  
 "Single species Population Dynamics",  
 "Single species Occurrence",  
 "Single species Movement",  
 "Single species Abundance/Density"  
 ))  
# Heatmap plot  
  
heatmap\_telemetric <- ggplot(filled\_nr\_metric\_tele,  
 aes(x=telecoupling, y=metricType, fill=count\_factor)) +  
 geom\_tile(color = "white", size=0.25) +  
 geom\_text(aes(label = count )) +  
 scale\_fill\_manual(values=c("grey90","#BCDDF5","#8FCBF5","#4EA6E6",  
 "#4E91CC","#3F76A6","#274866"))+  
 labs(x="",y="")+  
 scale\_y\_discrete(expand=c(0,0))+  
 theme\_grey(base\_size=8)+  
  
 theme(legend.title = element\_text(size = 10),  
 legend.text = element\_text(size = 10, face="bold"),  
 plot.background=element\_blank(),  
 plot.title = element\_blank(),  
 axis.title= element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.text.x.top = element\_text(size=10, angle = 90, vjust=0,  
 hjust = 0),  
 axis.text.y.left = element\_text(size=10, angle = 360, vjust=0.5,  
 hjust = 0),  
 strip.background = element\_rect(fill="white"),  
 panel.spacing = unit(0.1, "lines")) +  
 scale\_x\_discrete(position = "top") +  
 labs(fill = "number of entries")  
  
heatmap\_telemetric



Metric & Flow fig with sample size –CAN’T FIGURE OUT 10-19

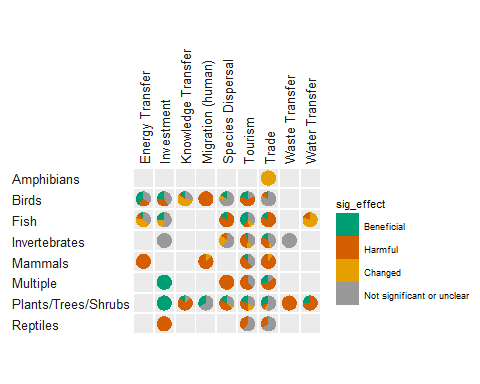
# Matrix of pie charts  
#   
# sig\_effect\_cols <- sigeff\_col  
#   
# var <- metric\_flow$count  
# max <- max(var, na.rm = T); max  
# min <- min(var, na.rm = T); min  
#   
# breaks <- seq(min, max, 2); breaks; length(breaks)  
# breaks <- sort(unique(shp\_df$n)); breaks; length(breaks)  
# myPalette <- colorRampPalette((brewer.pal(8, "Purples")));# myPalette; rev  
# colors <- myPalette(length(breaks)); colors  
#   
# metric\_flow\_fig <- ggplot(metric\_flow) +   
# aes(x='', y = count, fill = sig\_effect) +  
#   
# geom\_bar(position='fill',  
# stat='identity',  
# width = 1,  
# size = 0.01) +  
#   
# # activate for vertical or horizontal version  
# facet\_grid(metric\_type ~ metric ~ tc, switch='y') + #horizontal  
# #facet\_grid(tc ~ metric, switch='y') + #vertical  
#   
# # convert to pie chart  
# coord\_polar(theta = 'y') +  
#   
# # colors for pies  
# scale\_fill\_manual("Pies", values=sig\_effect\_cols, guide = guide\_legend(reverse = FALSE)) +  
# scale\_y\_discrete(position='right',expand=c(0,0)) +  
#   
# new\_scale("fill") + # add new scale  
#   
# # attempt at showing sample size  
# geom\_tile(data = filled\_nr\_metric\_tele, aes(fill = count\_factor)) +  
# scale\_fill\_manual("Tiles",values=c("grey90","#BCDDF5","#8FCBF5","#4EA6E6",  
# "#4E91CC","#3F76A6","#274866"))+  
#   
# # style  
# theme\_gray(base\_size=8)+  
# theme(  
# axis.title=element\_blank(),  
# axis.line=element\_blank(),  
# axis.ticks=element\_blank(),  
# axis.text=element\_blank(),  
# panel.border = element\_blank(),  
# panel.grid=element\_blank(),  
# panel.grid.major = element\_blank(),  
# panel.grid.minor = element\_blank(),  
# plot.background = element\_rect(),  
# plot.title=element\_blank(),  
# strip.text.x = element\_text(size=10,  
# angle=90, vjust=0, hjust = 0),  
# strip.text.y.left = element\_text(size=10,  
# angle=0, vjust=0, hjust = 0),  
# strip.background = element\_rect(fill='white'),  
# panel.spacing = unit(0.1, 'lines')  
# ) +  
# # labels  
# ylab('') + xlab('')  
#   
#   
#   
#   
# # display image  
# metric\_flow\_fig  
#

##Flow & Taxa data Convert to long format, where metric is indicated.

# get summary  
 metric\_flow <- ddply(metrics\_long,  
 .(taxa, tc, sig\_effect),  
 summarize,  
 # total count of entries   
 count=length(metric))  
  
# save as csv  
# write.csv(metric\_flow, paste0(tab.dir,'synthesis\_sig\_effects\_metric\_and\_flow.csv'),row.names = TRUE)  
  
# sig\_effects level change  
 metric\_flow$sig\_effect <- gsub("\\s\*\\([^\\)]+\\)", '',  
 as.character(metric\_flow$sig\_effect))  
 metric\_flow$sig\_effect <- factor(metric\_flow$sig\_effect,  
 c( "Beneficial","Harmful","Changed","Not significant or unclear"))  
# telecoupling level change  
 metric\_flow$tc <- factor(metric\_flow$tc,  
 c("Energy Transfer","Investment","Knowledge Transfer",  
 "Migration (human)","Migration (non-human)",  
 "Species Dispersal","Tourism","Trade","Waste Transfer","Water Transfer"  
 ))

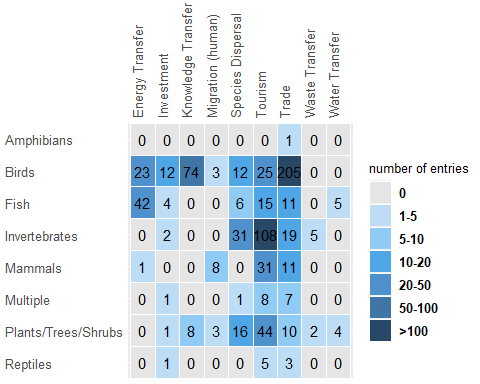
Flow and taxa figure

# Matrix of pie charts  
 sig\_effect\_cols <- sigeff\_col  
  
 metric\_flow\_fig <- ggplot(metric\_flow) +  
 aes(x='', y = count, fill = sig\_effect) +  
 geom\_bar(position='fill',  
 stat='identity',  
 width = 1,  
 size = 0.01) +  
 # activate for vertical or horizontal version  
 facet\_grid(taxa ~ tc, switch='y') + #horizontal  
 #facet\_grid(tc ~ metric, switch='y') + #vertical  
 # convert to pie chart  
 coord\_polar(theta = 'y') +  
 # colors  
 scale\_fill\_manual(  
 values=sig\_effect\_cols,  
 guide = guide\_legend(reverse = FALSE)) +  
 scale\_y\_discrete(position='right',expand=c(0,0))+  
 # style  
 theme\_gray(base\_size=8)+  
 theme(  
 axis.title=element\_blank(),  
 axis.line=element\_blank(),  
 axis.ticks=element\_blank(),  
 axis.text=element\_blank(),  
 panel.border = element\_blank(),  
 panel.grid=element\_blank(),  
 panel.grid.major = element\_blank(),  
 panel.grid.minor = element\_blank(),  
 plot.background = element\_rect(),  
 plot.title=element\_blank(),  
 strip.text.x = element\_text(size=10,  
 angle=90, vjust=0, hjust = 0),  
 strip.text.y.left = element\_text(size=10,  
 angle=0, vjust=0, hjust = 0),  
 strip.background = element\_rect(fill='white'),  
 panel.spacing = unit(0.1, 'lines')  
 ) +  
 # labels  
 ylab('') + xlab('')  
  
  
  
# display image  
 metric\_flow\_fig



Flow Taxa Heatmap

# Load data of number of studies by taxa and flow  
nr\_metric\_tele <- aggregate(x = metric\_flow$count,  
 by = list(metric\_flow$taxa, metric\_flow$tc),  
 FUN = sum)  
  
   
  
# Rename columns  
  
names(nr\_metric\_tele)[1:3] <- c("taxa", "telecoupling", "count")  
  
# Fill the gaps  
  
mat\_nr\_metric\_tele <-dcast(nr\_metric\_tele, taxa ~ telecoupling, value.var = 'count')  
filled\_nr\_metric\_tele <- melt(mat\_nr\_metric\_tele, id=c("taxa"))  
filled\_nr\_metric\_tele[is.na(filled\_nr\_metric\_tele)] <- 0  
names(filled\_nr\_metric\_tele)[2:3] <- c("telecoupling", "count")  
  
# Create 'count' categories for representation  
  
filled\_nr\_metric\_tele$count\_factor <- cut(filled\_nr\_metric\_tele$count,  
 breaks = c(-1,0,5,10,20,50,100,max(filled\_nr\_metric\_tele$count,na.rm=T)),  
 labels=c("0","1-5","5-10","10-20","20-50","50-100",">100"))  
  
  
# Change order of factor levels for consistency with other figures  
  
filled\_nr\_metric\_tele$taxa <- factor(filled\_nr\_metric\_tele$taxa, c('Reptiles','Plants/Trees/Shrubs','Multiple','Mammals',  
 'Invertebrates','Fish','Birds','Amphibians'))  
  
# Heatmap plot  
  
heatmap\_telemetric <- ggplot(filled\_nr\_metric\_tele,  
 aes(x=telecoupling, y=taxa, fill=count\_factor)) +  
 geom\_tile(color = "white", size=0.25) +  
 geom\_text(aes(label = count )) +  
 scale\_fill\_manual(values=c("grey90","#BCDDF5","#8FCBF5","#4EA6E6",  
 "#4E91CC","#3F76A6","#274866"))+  
 labs(x="",y="")+  
 scale\_y\_discrete(expand=c(0,0))+  
 theme\_grey(base\_size=8)+  
  
 theme(legend.title = element\_text(size = 10),  
 legend.text = element\_text(size = 10, face="bold"),  
 plot.background=element\_blank(),  
 plot.title = element\_blank(),  
 axis.title= element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.text.x.top = element\_text(size=10, angle = 90, vjust=0,  
 hjust = 0),  
 axis.text.y.left = element\_text(size=10, angle = 360, vjust=0.5,  
 hjust = 0),  
 strip.background = element\_rect(fill="white"),  
 panel.spacing = unit(0.1, "lines")) +  
 scale\_x\_discrete(position = "top") +  
 labs(fill = "number of entries")  
  
heatmap\_telemetric



## Country/Region Entry Effects

TBD