

Supplemental, the discription of the empirical LIT dataset and the R code

The top six rows of the empirical LIT dataset

The data is available in the Dryad Digital Repository: <https://doi.org/10.5061/dryad.0rxwds35>.

Year	Month	Day	Location	Depth	Latitude	Longitude	Transect	Adj_Transect_marker	Major_Category_RC	Category_RC	Hard.coral_species.name
2019	7	5	Yehliu	Shallow	25.20396	121.6813	1	0	TF	TF	
2019	7	5	Yehliu	Shallow	25.20396	121.6813	1	1	TF	TF	
2019	7	5	Yehliu	Shallow	25.20396	121.6813	1	2	TF	TF	
2019	7	5	Yehliu	Shallow	25.20396	121.6813	1	3	TF	TF	
2019	7	5	Yehliu	Shallow	25.20396	121.6813	1	4	TF	TF	
2019	7	5	Yehliu	Shallow	25.20396	121.6813	1	5	TF	TF	

```
## 'data.frame':    26900 obs. of  12 variables:
## $ Year           : int   2019 2019 2019 2019 2019 2019 2019 2019 2019 2019 ...
## $ Month          : int   7 7 7 7 7 7 7 7 7 7 ...
## $ Day            : int   5 5 5 5 5 5 5 5 5 5 ...
## $ Location       : chr   "Yehliu" "Yehliu" "Yehliu" "Yehliu" ...
## $ Depth          : chr   "Shallow" "Shallow" "Shallow" "Shallow" ...
## $ Latitude       : num  25.2 25.2 25.2 25.2 25.2 ...
## $ Longitude      : num  122 122 122 122 122 ...
## $ Transect       : int   1 1 1 1 1 1 1 1 1 1 ...
## $ Adj_Transect_marker : int   0 1 2 3 4 5 6 7 8 9 ...
## $ Major_Category_RC : chr   "TF" "TF" "TF" "TF" ...
## $ Category_RC     : chr   "TF" "TF" "TF" "TF" ...
## $ Hard_coral_species.name: chr   "" "" "" "" ...
```

"Year", "Month", and "Day" indicates the dates of the surveys conducted, respectively.

"Location" indicates reefs where the surveys were conducted. There are four levels, including Yeliu, Chinwan Inner Bay, Kihaw, and Tiao-Shi.

"Depth" includes the two depths, shallow and deep, of the survey conducted.

"Transect" indicates the three consecutive transects surveyed at each site.

"Adj_Transect_marker" indicates the point, from 0 to 1,499, of each centimeter along the 15 meters transect. There are 1,500 points in each transect except for the first transect at the shallow depth of Tiao-Shi.

"Category_RC" indicates the ten categories of Reef Check's Tropical Program protocol. Noted that rock (RC) was divided into rock (RC), turf algae (TF), and crustose coralline algae (CCA) in the raw dataset.

"Major_Category_RC". In this column, "hard coral (HC)" and "soft coral (SC)" were replaced by "Coral" for calculating the total coral cover of each transect, while the rest categories remain the same.

"Hard.coral_species.name" records the species name of hard coral.

Function to compute the coverage (frequency) of a given species from a sample

```
commRand.fn <- function(x){
  out <- rep(0,NROW(spID.list)) #Candidate species
  dat.rand <- dat[x,]
  target.sp <- as.character(dat.rand$Category_RC) # sampled target species
  if(NROW(target.sp)!=0){
    freq <- as.data.frame(table(target.sp))
    row.ix <- which(spID.list==freq[,1])
    out[row.ix] <- freq[,2] # assign frequency (No. of species) to community vector
  }
  else{
    out <- out
  }
  return(out)
}
```

The code for determining the optimum sampling interval for the Point Intercept Transect method on 15 m transects

The code is for (1) calculating the standard deviation (SD) of the total coral cover (TCC) of each of the 200 PIT communities generated with the same number of random points on each 15 m transect using the bootstrap approach with a Monte Carlo procedure from the empirical LIT datasets, and (2) computing community similarity index between each PIT community and the LIT community of the same transect.

```
library(vegan)
datt <- The empirical LIT dataset
datt$Category_RC[datt$Category_RC=="TF"] <- "RC" ### Convert to fit the Reef Check 10 category
datt$Category_RC[datt$Category_RC=="CCA"] <- "RC" ### Convert to fit the Reef Check 10 category
datt$transectID <- paste0(datt$Location,datt$Depth,datt$Transect)
ID.list <- as.character(unique(datt$transectID))
spID.list <- as.character(unique(datt$Category_RC))
#
coverageSD <- matrix(NA,length(ID.list),1500) # 15m transect
distcommMean <- distcommSD <- matrix(NA,length(ID.list),6) # 6 interval for no. of points
for(j in 1:length(ID.list)){
  print(ID.list[j])
  dat <- datt[datt$transectID==ID.list[j],]
  dat$coral <- 0
  dat$coral[dat$Major_Category_RC=="Coral"] <- 1
  if(all(dat$coral==0)){print(paste0("WARNING: ",ID.list[j]," contains no coral !!!!!!!!!!!!!"))}
  corals <- as.numeric(dat$coral)
  nrep <- 200
  maxMarker <- NROW(dat$coral) # maximum marker value
  coverage <- matrix(0,nrep,maxMarker)
  for(i in 1:maxMarker){
    if(i%%100==0){print(paste0("For loops at ",i))}
    rand <- replicate(nrep, sample(dat$Adj_Transect_marker+1, i, replace=FALSE)) # Transect_marker contains zero
    rand.mat <- matrix(rand, ncol=nrep)
    coverage.i <- apply(rand.mat,2,function(x){corals[x]})
    if(is.null(dim(coverage.i))){
      coverage[,i] <- coverage.i
    }
    else{
      coverage[,i] <- colMeans(coverage.i)
    }
  }
  n <- 1500-maxMarker
  if(n>0){
    coverage <- cbind(coverage,matrix(NA,NROW(coverage),n))
    coverageSD[j,] <- apply(coverage,2,sd)
  }
  else{
    coverageSD[j,] <- apply(coverage,2,sd)
  }
  #
  # computing community similarity index for PIT vs. LIT
  markerList <- c(15,30,60,150,300,750) # markers (points) for 15m transect
  gdist.all <- c()
  for(i in 1:length(markerList)){
    rand <- replicate(nrep, sample(dat$Adj_Transect_marker+1, markerList[i], replace=FALSE))
    rand.mat <- matrix(rand, ncol=nrep)
    commRand.all <- apply(rand.mat,2,commRand.fn)/markerList[i] # random community _percentage
    commLIT <- commRand.fn(1:nrow(dat))/maxMarker # baseline community _percentage
    gdist <- c()
    for(k in 1:ncol(commRand.all)){
      commk <- cbind(commRand.all[,k],commLIT)
      gdist[k] <- as.numeric(vegdist(decostand(t(commk), "norm"), method="bray"))
    }
    gdist.all <- cbind(gdist.all,gdist)
  }
  distcommMean[j,] <- colMeans(gdist.all)
  distcommSD[j,] <- apply(gdist.all,2,sd)
}
#
distcommMean.all <- cbind(distcommMean,0)
distcommSD.all <- cbind(distcommSD,0)
colnames(distcommMean.all) <- c(markerList,"LIT")
rownames(distcommMean.all) <- ID.list
colnames(distcommSD.all) <- c(markerList,"LIT")
rownames(distcommSD.all) <- ID.list
#
rownames(coverageSD) <- ID.list
```

The code for testing whether the optimum sampling interval depends on transect length

The code first stacked three empirical transects from each of the six sites were first stacked end to end to form a 45 m long virtual transect. Five replicates of each transect lengths interested (10m, 20m, 30m, 40m) were randomly extracted from each of the 45 m virtual transects. "pitSize" and "markerList" were used to control the length of transects extracted. For instance, "pitSize <- 2,000" and "markerList <- c(20,40,80,200,400,1000)" were used for extracting 20m transects. The same process above were then used to calculate the SD of the total coral cover (TCC) of each of the 200 PIT communities generated with the same number of random points and the community similarity index between each PIT community and the LIT community of the same transect.

```
library(vegan)
datt <- The empirical LIT dataset
datt$Category_RC[datt$Category_RC=="TF"] <- "RC" ### Convert to fit the Reef Check 10 category
datt$Category_RC[datt$Category_RC=="CCA"] <- "RC" ### Convert to fit the Reef Check 10 category
datt$transectID <- paste0(datt$Location,datt$Depth,datt$Transect)
ID.list <- as.character(unique(datt$transectID))
spID.list <- as.character(unique(datt$Category_RC))
datt$combinedID <- gsub("[:A-Z:]", "", datt$transectID)
datt$Adj_Transect_marker_combined <- 0
datt$Adj_Transect_marker_combined[datt$combinedID=="CFS"] <- 1:length(which(datt$combinedID=="CFS"))
datt$Adj_Transect_marker_combined[datt$combinedID=="TSD"] <- 1:length(which(datt$combinedID=="TSD"))
datt$Adj_Transect_marker_combined[datt$combinedID=="CWS"] <- 1:length(which(datt$combinedID=="CWS"))
datt$Adj_Transect_marker_combined[datt$combinedID=="TSS"] <- 1:length(which(datt$combinedID=="TSS"))
datt$Adj_Transect_marker_combined[datt$combinedID=="YLS"] <- 1:length(which(datt$combinedID=="YLS"))
datt$Adj_Transect_marker_combined[datt$combinedID=="YLD"] <- 1:length(which(datt$combinedID=="YLD"))
#
combID.list <- as.character(unique(datt$combinedID)) # the six site and depth combination
spID.list <- as.character(unique(datt$Category_RC)) # Reef Check categories
#
# pitSize <- 1000 #pitSize of the combined transect 1000=10m ## this should change in accordance with "markerList"
#
pitSize <- 2000
# pitSize <- 3000
# pitSize <- 4000

coverageSD <- matrix(NA,length(combID.list), pitSize)
distcommMean <- distcommSD <- matrix(NA,length(combID.list),6) # 6 intervals for marker points of combined transe
cts
#
coverageSD.list <- distcommMean.list <- vector("list",5) # 5 replicates for N=6 sites of combined transects
for(m in 1:5){
  for(j in 1:length(combID.list)){
    print(combID.list[j])
    datcomb <- datt[datt$combinedID==combID.list[j],]
    rand <- ceiling(runif(1,0,NROW(datcomb)-pitSize))
    markerConmb <- rand:(rand+pitSize-1)
    dat <- datcomb[markerConmb,]
    #
    dat$coral <- 0
    dat$coral[dat$Major_Category_RC=="Coral"] <- 1
    if(all(dat$coral==0)){print(paste0("WARNING: ",combID.list[j]," contains no coral !!!!!!!!!!!!!"))}
    corals <- as.numeric(dat$coral)
    nrep <- 200 # 200 replicates for each sampling procedure
    maxMarker <- NROW(dat$coral) # maximum marker value
    coverage <- matrix(0,nrep,maxMarker)
    for(i in 1:maxMarker){
      if(i%%100==0){print(paste0("For loops at ",i))}
      rand <- replicate(nrep, sample(1:maxMarker, i, replace=FALSE)) # Transect_marker contains zero
      rand.mat <- matrix(rand, ncol=nrep)
      coverage.i <- apply(rand.mat,2,function(x){corals[x]})
      if(is.null(dim(coverage.i))){
        coverage[,i] <- coverage.i
      }
      else{
        coverage[,i] <- colMeans(coverage.i)
      }
    }
    coverageSD[j,] <- apply(coverage,2,sd)
  }
  #
  # computing community similarity index for PIT vs. LIT
  markerList <- c(10,20,40,100,200,500) # inputs of markers (points) for 10m transect ## this should change i
n accordance with "pitSize"
  markerList <- c(20,40,80,200,400,1000) # inputs of markers (points) for 20m transect
  markerList <- c(30,60,120,300,600,1500) # inputs of markers (points) for 30m transect
  markerList <- c(40,80,160,400,800,2000) # inputs of markers (points) for 40m transect

  gdist.all <- c()
  for(i in 1:length(markerList)){
    rand <- replicate(nrep, sample(1:NROW(dat), markerList[i], replace=FALSE))
    rand.mat <- matrix(rand, ncol=nrep)
    commRand.all <- apply(rand.mat,2,commRand.fn)/markerList[i] # random community _percentage
    commLIT <- commRand.fn(1:nrow(dat))/maxMarker # baseline community _percentage
    gdist <- c()
    for(k in 1:ncol(commRand.all)){
      commk <- cbind(commRand.all[,k],commLIT)
      gdist[k] <- as.numeric(vegdist(decostand(t(commk), "norm"), method="bray"))
    }
    gdist.all <- cbind(gdist.all,gdist)
  }
  distcommMean[j,] <- colMeans(gdist.all)
}
colnames(distcommMean) <- markerList
rownames(distcommMean) <- combID.list
coverageSD.list[[m]] <- coverageSD
distcommMean.list[[m]] <- distcommMean
}
```