### Supplementary material

**Supp. Mat. 1 :** Rogers’ genetic distance

Given the number of loci, the number ofalleles observed in locus *k*, and and the proportions of allele *j* at locus *k* in populations 1 and 2, respectively:

The sum of the allele proportions for any locus in any population is 1.

For biallelic markers such as SNP, the calculation becomes even simpler as seen in the example in **Table 1**, A. Indeed, because the proportion of an allele is always 1 minus the proportion of the other allele, the sum per locus is always twice the term for the a single allele and the equation simplifies to:

**Supp. Mat. 2 :** TGI function and output.

# mat1: the genotypic matrix associated with the first sampling; must be a genind object

# mat2: the genotypic matrix associated with the second sampling; must be a genind object

# nperm: the the number of permutations used in the evaluation of significance

# seed.: you may specify a seed by using this argument

# method : a number between 1 and 5. Five genetic distances are available in function dist.genpop # of the adegenet package.

# They are : (1) Nei’s D, (2) Edwards’ angular D, (3) Reynolds’ coancestry coefficient, (4)

# Rogers’ D, (5) Prevosti’s absolute genetic D. Methods 2, 3 and 4 produce Euclidean distances,

# whereas methods 1 and 5 produce non-Euclidean distances, which produce negative eigenvalues and # complex eigenvectors in principal coordinate analysis.

# correc: correction for multiple # inference; see ?p.adjust

# thresh\_for\_GL: indicate here the threshold you want to use

TGI <- function (mat1, mat2, nperm = 999, replace = FALSE, seed. = NULL, method = 4, correc = "holm", thresh\_for\_GL = 0.05) {

#### genind to genpop objects

mat1p <- genind2genpop(mat1)

mat1p <- mat1p[,order(colnames(mat1p@tab))]

mat2p <- genind2genpop(mat2)

mat2p <- mat2p[,order(colnames(mat2p@tab))]

##### Function to compute genetic distances

dissim <- function(mat1p, mat2p, method) {

dis <- vector(mode = "numeric", length = nrow(mat1p@tab))

for (i in 1:nrow(mat1p@tab)){

if (i == 1){

trick <- 2

} else {

trick <- 1

}

temp\_genpop <- mat1p

temp\_genpop@tab[trick,] <- mat2p@tab[i,]

dis[i] <- dist.genpop(temp\_genpop[c(trick, i),], method = method)

}

list(dis = dis)

}

##### Initialization of seed, tolerance

if (!is.null(seed.)){

set.seed(seed.)

}

epsilon <- sqrt(.Machine$double.eps)

##### Dimensions check

n <- nrow(mat1p@tab)

p <- ncol(mat1p@tab)

if ((nrow(mat2p@tab) != n) | (ncol(mat2p@tab) != p)){

stop("The matrices are not of the same size!")

}

##### Empirical genetic distances

tmp <- dissim(mat1p, mat2p, method)

dis.ref <- tmp$dis

##### Permutations

if (nperm > 0) {

my.vec <- sample(1:(10 \* nperm), size = nperm)

outlier.count = rep(1, n)

for (iperm in 1:nperm) {

set.seed(my.vec[iperm])

mat1.perm <- mat1p

mat1.perm <- shufflepop(mat1.perm, method=4)

set.seed(my.vec[iperm])

mat2.perm <- mat2p

mat2.perm <- shufflepop(mat2.perm, method=4)

tmp <- dissim(mat1.perm, mat2.perm, method)

dis.perm <- tmp$dis

ge <- which(dis.perm + epsilon >= dis.ref)

if (length(ge) > 0) {

outlier.count[ge] <- outlier.count[ge] + 1

}

}

p.dist <- outlier.count/(nperm + 1)

}

p.adj <- p.adjust(p.dist, method = correc)

##### Gain or loss?

n.pop1 <- seppop(mat1)

n.pop2 <- seppop(mat2)

mean.hexp1 <- do.call("c", lapply(n.pop1, function(x) mean(summary(x)$Hexp)))

mean.hexp2 <- do.call("c", lapply(n.pop2, function(x) mean(summary(x)$Hexp)))

mean.hexp1[is.nan(mean.hexp1)] <- NA

mean.hexp2[is.nan(mean.hexp2)] <- NA

simple\_diff <- mean.hexp2 - mean.hexp1

output <- list(TBI = dis.ref, p.TBI = p.dist, p.adj = p.adj, gainloss = simple\_diff[p.adj < thresh\_for\_GL])

class(output) <- "TGI"

return(output)

}

#############################################################################

**> goby\_test <- TGI(goby\_first, goby\_second, nperm = 9999, method = 4)**

**> goby\_test**

$TBI #index values

[1] 0.06432131 0.06089485 **0.15212258** 0.02258920 0.07247326 0.04463856 0.06672004 0.02238467

$p.TBI #unadjusted permutation p-values

[1] 0.4283 0.4891 **0.0004** 0.9943 0.3188 0.7756 0.3947 0.9949

$p.adj #adjusted p-values

[1] 1.0000 1.0000 **0.0032** 1.0000 1.0000 1.0000 1.0000 1.0000

$gainloss #difference in expected heterozygosity

**ELK**

**-0.04567755**

attr(,"class")

[1] "TGI"



20km

kkkkkkk km

N

N

**Supp. Mat. 3 :** Satellite map of the Californian sampling stations used in the goby analysis. The red circle marks the Elk population which was the only one found to have significantly changed.