

1 **R code to calculate and plot national RLIs weighted by the proportion of each species' distribution within a country or region**

2 **Developed by Maria Dias, BirdLife International**

3 **(ver. March 2017)**

4 **Methods for calculating national RLIs weighted by the fraction of each species' distribution occurring within the country or region.**

5 The Red List Index can be downscaled to show national and regional Red List Indices, weighted by the fraction of each species' distribution occurring within  
6 the country or region, building on the method published by [Rodrigues et al. \(2014\) PLoS ONE 9\(11\): e113934](#). These show an index of aggregate survival  
7 probability (the inverse of extinction risk) for all birds, mammals, amphibians, corals and cycads occurring within the country or region. The index shows  
8 how well species are conserved in a country or region to its potential contribution to global species conservation. The index is calculated as:  
9

10 
$$RLI_{(t,u)} = 1 - \frac{\sum_s (W_{(t,s)} \times \frac{r_{su}}{R_s})}{W_{EX} \times \sum_s (\frac{r_{su}}{R_s})}$$

11 where  $t$  is the year of comprehensive reassessment,  $u$  is the spatial unit (i.e. country),  $W_{(t,s)}$  is the weight of the global Red List category for species  $s$  at  
12 time  $t$  (Least Concern =0, Near Threatened =1, Vulnerable =2, Endangered =3, Critically Endangered =4, Critically Endangered (Possibly Extinct) =5, Critically  
13 Endangered (Possibly Extinct in the Wild) =5, Extinct in the Wild =5 and Extinct =5),  $W_{EX} = 5$  is the weight for Extinct species,  $r_{su}$  is the fraction of the total  
14 range of species  $s$  in unit  $u$ , and  $R_s$  is the total range size of species  $s$ .

15 The index varies from 1 if the country has contributed the minimum it can to the global RLI (i.e., if the numerator is 0 because all species in the country are  
16 LC) to 0 if the country has contributed the maximum it can to the global RLI (i.e., if the numerator equals the denominator because all species in the country  
17 are Extinct or Possibly Extinct).  
18

19 The taxonomic groups included are those in which all species have been assessed for The IUCN Red List of Threatened Species™ more than once. Red List  
20 categories for years in which comprehensive assessments (i.e. those in which all species in the taxonomic group have been assessed) were carried out are  
21 determined following the approach of [Butchart et al. 2007; PLoS ONE 2\(1\): e140](#), i.e. they match the current categories except for those taxa that have  
22 undergone genuine improvement or deterioration in extinction risk of sufficient magnitude to qualify for a higher or lower Red List category.  
23

24 An aggregate Red List Index across all taxonomic groups for each country or region is calculated following [Butchart et al. 2010 Science 328: 1164-1168](#). It is  
25 calculated as the arithmetic mean of modelled Red List Indices for each taxonomic group. The latter are interpolated linearly for years between data points  
26 and extrapolated linearly (with a slope equal to that between the two closest assessed points, except for corals) back to the earliest time point and  
27 forwards to the present for years for which estimates are not available. The start year of the aggregated index is set as ten years before the first assessment

28 year for the taxonomic group with the latest starting point. Corals are not extrapolated linearly because declines are known to have been much steeper  
 29 subsequent to 1996 (owing to extreme bleaching events) than before. Therefore the rate of decline prior to 1996 is set as the average of the rates for the  
 30 other taxonomic groups.

31  
 32 The Red List Indices for each taxonomic group for each year are modelled to take into account various sources of uncertainty: (i) Data Deficiency: Red List  
 33 categories (from Least Concern to Extinct) are assigned to all Data Deficient species, with a probability proportional to the number of species in non-Data  
 34 Deficient categories for that taxonomic group. (ii) Extrapolation uncertainty: although Red List Indices are extrapolated linearly based on the slope of the  
 35 closest two assessed point, there is uncertainty about how accurate this slope may be. To incorporate this uncertainty, rather than extrapolating  
 36 deterministically, the slope used for extrapolation is selected from a normal distribution with a probability equal to the slope of the closest two assessed  
 37 points, and standard deviation equal to 60% of this slope (i.e., the CV is 60%). (iii) Temporal variability: the ‘true’ RLI likely changes from year to year, but  
 38 because assessments are repeated only at multi-year intervals, the precise value for any particular year is uncertain. To make this uncertainty explicit, the  
 39 Red List Index value for a given taxonomic group in a given year is assigned from a moving window of five years, centred on the focal year. Assessment  
 40 uncertainty cannot yet be incorporated into the index. Practically, these uncertainties are incorporated into the aggregated Red List Index as follows: Data  
 41 Deficient species are allotted a category as described above, and a Red List Index for each taxonomic group is calculated interpolating and extrapolating as  
 42 described above. A final Red List Index value is assigned to each taxonomic group for each year from a window of years as described above. Each such ‘run’  
 43 produces a Red List Index for the complete time period for each taxonomic group, incorporating the various sources of uncertainty. One thousand such runs  
 44 is generated for each taxonomic group, and the mean is calculated.

45  
 46 This code requires the following input files (shapefile or CSV format):

- 47  
 48 1. Table "list\_all\_species\_wshapes.csv": Metada table to aux the analyses. “mapped”=if map exists; “Shapefile”=name of the shapefile with the range  
 49 map; “subfolder” =folder where it is stored (assumed that the subfolder is in the folder defined in table “tbfolders.csv” – see above); “size”=size of  
 50 the shapefile (in Kb; only for non-combined shaped); “combinedshape”: if the layers are combined in a single shapefile (1) or in one shapefile per  
 51 species (0); “mapped”: 1 if the species has a range map to be included in the analyses (can be 0 in some cases – e.g. endemic species)

Scientific	Group	TaxonID	Cat_latest	year_ Cat	mapped	ende mic	ordem	shapefile	subfolder	size	combined shape	mappedf
<i>Struthio camelus</i>	Bird	45020636	LC	2015	1		1	<i>Struthio_camelus_45020636.shp</i>	<i>Struthionidae</i>	202.072	0	1
<i>Zamia wallisii</i>	Cycad	2.14E+09	CR	2013	1	<i>Colo mbia</i>	24221	<i>zamia_wallisii.shp</i>	<i>Zamia</i>	0.3	0	1

52

53 2. Table "Iso\_countries.csv": list of ISO codes, country names and respective regions/other classifications (note 0 values – not NA – for missing values  
 54 in "Region", "Developing", "LDC" and "LLDC\_SIDS" fields)

ordem	country_SIS	countryname	region1	ISO3	ISO_BL	ISO_SDG	uname_BL	uname_SDG	Region	Developing	LDC	LLDC_SIDS
247	High Seas	Areas Beyond National Jurisdiction	High Seas	ABNJ	ABNJ	ABNJ	High Seas	High Seas	0	0	0	0
202	Aruba	Aruba	Caribbean Islands	ABW	ABW	ABW	Aruba	Aruba	Latin America & the Caribbean	Developing	0	SIDS

55

56 3. Table "genuine\_changes.csv": table with the details for species that have qualified for genuine changes in their IUCN Red List since the first  
 57 comprehensive assessment for the group. All categories in the Genuine changes file must match one of LC, NT, VU, EN, CR, CR(PE), CR(PEW), EW, or  
 58 EX. (Note DD and NE are not permitted)

Group	Scientific	Taxon ID	Period	cat_start	cat_end	start_year	end_year
Bird	Nothocercus nigrocapillus	22678160	08-12	LC	VU	2008	2012

59

60

61 4. Table "Sp\_cnts\_pp\_FINAL.csv": table with the proportion of each species' distribution in each country.

Scientific	TaxonID	marine	ISO3	pp	ovl	method	batch
<i>Abditomys latidens</i>	15337	0	PHL	1	terrestrial	vector_overlap	batch8

62

63 The code outputs several graphs and tables with the results of the analyses (global RLI and per country – XXX = code for each country/region):

- 64 1. global\_all\_taxa.pdf: graph with the global RLI for each taxonomic group and aggregated
- 65 2. global\_aggregated.pdf: graph with the global aggregated RLI
- 66 3. global\_birds.pdf: graph with the global RLI for birds
- 67 4. XXX.csv: table with national rli values and 95% confidence intervals (qn05 and qn95) for each taxonomic group and year
- 68 5. XXX\_cqli.csv: table with national cqli values and 95% confidence intervals (cqli05 and cqli95) for each taxonomic group and year
- 69 6. XXX\_aggregated.pdf: graph with the national aggregated RLI
- 70 7. XXX\_all\_taxa.pdf: graph with the national RLI for each taxonomic group and aggregated

- 71 8. XXX\_birds.pdf: graph with the national RLI for birds
- 72 9. XXX\_crlf.pdf: graph with the national CRLI for each taxonomic group and aggregated
- 73 10. allcountries\_RLI.csv: table with rli values and 95% confidence intervals (qn05 and qn95) for each country, taxonomic group and year (i.e., several
- 74 XXX.csv combined in one file)
- 75 11. allcountries\_CRLI.csv table with crli values and 95% confidence intervals (crlf05 and crlf95) for each country, taxonomic group and year (i.e., several
- 76 XXX\_crlf.csv combined in one file)
- 77 12. RLI\_all\_countries.pdf (graphs XXX\_all\_taxa.pdf combined in a single file)

78

79 Text that is highlighted should be amended to give the appropriate folder names and file paths, number of randomisations (if this is to be adjusted), final  
80 year and ISO code/region to use (see grey columns in file Iso\_countries.csv below) .

81 # National RLIs calculator v1 26/06/2016

82 # Maria Dias/BirdLife International; maria.dias@birdlife.org

83 # Script to calculate and plot the national RLIs

84 ## See formats of input tables above

85

86 ##### Parameters that need to be set

87 wkfolder="ADD FOLDER NAME/LOCATION HERE" ## copy-paste the working folder, replacing "/" by "/" and ensuring no spaces before & after quotes

88 setwd(wkfolder)

89

90 finfolder="ADD FOLDER NAME/LOCATION HERE" ## copy-paste the location of the folder where the country files should be saved, replacing "/" by "/" and ensuring no  
91 spaces before & after quotes

92 finfolder2="ADD FOLDER NAME/LOCATION HERE" ## copy-paste the location of the folder where the final plots should be saved, replacing "/" by "/" and ensuring no  
93 spaces before & after quotes

94

95 isos=read.csv("Iso\_countries.csv") ## file with ISO codes; should be stored in the wkfolder specified above

96 tabsp1=read.csv("list\_all\_species\_wshapes.csv") ## file with the name of shapefiles and folders of the species' range maps; should be stored in the wkfolder specified  
97 above

98 gench=read.csv("genuine\_changes.csv") ### file with list of Red List genuine changes; should be stored in the wkfolder specified above

99 tbf1=read.csv("Sp\_cnts\_pp.csv") ## result of spatial analyses to estimate the proportion of the range of each species in each country

100

101 slopecv=.6 ### standard deviation of the slope

102 maxyear=2017 ### max year of the analyses

```

103 repetitions=1000 ## n repetitions
104 iso2use="M49_Region" #ISO_SDG #ISO_BL #LLDC_SIDS #LDC #Region #Developing ### iso to use in the analyses; choose from those available in isos table
105
106 ##### custom functions
107
108 lu=function (x=x) length(unique(x)) ## function to estimate the number of unique values in a vector
109 first=function (x=x) x[1] ## function to output the first value of a vector
110 q95=function (x=x) quantile(x,.95) ## function to estimate the quantile 95%
111 q05=function (x=x) quantile(x,.05) ## function to estimate the quantile 5%
112 div0=function (x=x, y=y) trunc((x-1)/y)-((x-1)/y) ## function that outputs 0 if y-1 is divisible by x (aux open new devices in graphics outputs)
113 cinz=rgb(217/255,217/255,217/255,.4) ### grey transparent colour for the plots (in rgb)
114
115 #function to calculate global RLI
116 # nreps=number of repetitions; groups2=list of taxonomic groups to analyse; leftextrapol=if slope should be extrapolated to the left; saveleftslope=if left slope should be
117 saved; plotit=if each iteration should be plotted
118 rlicalc=function (nreps=repetitions, groups2=groups2, years=years, leftextrapol=T, saveleftslope=T, plotit=F){
119
120 rliTot=data.frame(group=rep(rep(groups2, each=length(years)),length(nreps)), year=rep(rep(years, length(groups2)), length(nreps)),
121 nrep=rep(1:nreps,each=length(years)*length(groups2)), rli=NA)
122 outpts=list()
123
124 if (saveleftslope) leftslopes=NULL
125 for (r in 1:nreps)
126 {
127 #r=1
128 for (g in 1:length(groups2))
129 {
130 #g=1
131 group=as.character(groups2[g])
132 #group="Bird"
133 rlig=data.frame()
134 group
135 tgroup=tabgrli[tabgrli$group==group,]
136 yearsa=sort(unique(tgroup$year))
137 yearsa
138

```

```

139   ### years of assessment
140   for (y in 1:length(yearsa))
141   {
142     #y=1
143     year=years[a[y]]
144     year
145     tb=tgroup[tgroup$year==year,]
146     tb=merge(tb,wts)
147     head(tb)
148     nrow(tb)
149     rli1=1-(sum(tb$wg)/(wex*nrow(tb)))
150     rli1
151     rlig=rbind(rlig, data.frame(group=group,year=year, rli=rli1))
152   } ## ends years of assessment
153   rlig
154
155   ### intrapolation and extrapolation
156
157   dat=rlig
158   dat
159   #interpolation
160   yearsti=min(dat$year): max(dat$year)
161   irli=approx(dat$year, dat$rli, xout=yearsti, method="linear", rule=2:2)$y
162   irli
163
164   datf=data.frame(year=yearsti, rli=irli)
165   datf
166   #with(datf, plot(year, rli, xlim=c(1975,2018), ylim=c(0.6,1.2)))
167
168   #extrapolation # y=a+bx
169   #left
170   yearl=c(min(years)-2, min(years)-1, years[which(years<min(yearsti))])
171   yearl
172
173   x1=sort(dat$year)[1:2]
174   x1 ### years

```

```

175 y1=c(dat$rli[dat$year==x1[1]],dat$rli[dat$year==x1[2]])
176 y1 ###rlis
177 nv=lm(y1~x1)
178 mnslp <- coef(nv)[2]
179 useslp <- rnorm(1, mnslp, sd=slopecv*abs(mnslp))
180 if (leftextrapol==F) useslp <- meanleftslopes
181 irlel=y1[1] - (useslp*(x1[1] -(yearl)))
182
183 if (leftextrapol) leftslopes=c(leftslopes,useslp)
184 datf=rbind(datf, data.frame(year=yearl, rli=irlel))
185
186 #with(datf, points(year, rli, col=2, pch=19, cex=.6))
187 datf
188
189 #right
190 yearr=c(years[which(years>max(yearsti))],max(years)+1,max(years)+2)
191 yearr
192
193 x1=sort(dat$year)[(nrow(dat)-1):nrow(dat)]
194 x1 ## years
195 y1=c(dat$rli[dat$year==x1[1]],dat$rli[dat$year==x1[2]])
196 y1 ###rlis
197 nv=lm(y1~x1)
198 mnslp <- coef(nv)[2]
199 useslp <- rnorm(1, mnslp, sd=slopecv*abs(mnslp))
200 irler=y1[2] - (useslp*(x1[2] -(yearr)))
201 datf=rbind(datf, data.frame(year=yearr, rli=irler))
202 #with(datf, points(year, rli, col=2, pch=19, cex=.6))
203
204 datf=datf[order(datf$year),]
205
206
207 if (plotit)
208 {
209 if (r<30) win.graph()
210 with(datf, plot(year, rli, pch=19, col=2, main=paste(group,"s", sep="")))

```

```

211 with(datf[datf$year%in%yearsti,], points(year, rli, pch=19, col=4))
212 }
213
214 restmw=NULL
215 for (m in 3:(nrow(datf)-2)) restmw=c(restmw, sample(datf$rli[(m-2):(m+2)],1))
216 finalmw=restmw
217 rliTot[rliTot$group==group&rliTot$nrep==r,]$rli=finalmw
218 if (plotit) with(rliTot[rliTot$group==group&rliTot$nrep==r,], lines(year, rli, col=8))
219
220 } #ends loop groups
221 } #ends loop nreps
222
223 outpts[[1]]=rliTot
224 if (saveleftslope) outpts[[2]]=mean(leftslopes)
225
226 return(outpts)
227 }
228
229 #####
230
231 isos$isof=isos[,which(names(isos)==iso2use)]
232 head(isos)
233 lu(isos$isof)
234
235 nrow(tbf1)
236 head(tbf1)
237 tbf1=merge(tbf1, isos[,c("ISO3","isof")])
238 tbf1=merge(tbf1,tbsp1[,c("Scientific","Group","Cat_latest")], sort=F)
239 nrow(tbf1)
240 head(tbf1)
241 tbf1$iucn=tbf1$Cat_latest
242 tbf=tbf1[tbf1$iucn!="DD",]
243 nrow(tbf)
244 lu(tbf$isof)
245
246 unique(tbf$iucn)

```

```

247 wts=data.frame(iucn=c("LC","NT","VU","EN","CR","CR(PE)","CR(PEW)","EW","EX", "DD"), wg=c(0,1,2,3,4,5,5,5,5,-1))
248
249 # verify is all IUCN codes are correct
250 cds1=unique(tbf$iucn)
251 length(cds1)
252 length(cds1[cds1%in%wts$iucn]) ## should be the same as previous
253
254 cds1=unique(gench$cat_start)
255 length(cds1)
256 length(cds1[cds1%in%wts$iucn]) ## should be the same as previous
257
258 cds1=unique(gench$cat_end)
259 length(cds1)
260 length(cds1[cds1%in%wts$iucn]) ## should be the same as previous
261
262 #verify if all the scientific names in genuine file are correct and are in in sp-cnts table
263 sp1=unique(gench$Scientific) #tabf$RLC #gench$cat_start #gench$cat_end
264 length(sp1)
265 length(sp1[sp1%in%tbf$Scientific])
266 sp1[!sp1%in%tbf$Scientific] ### species that are missing in the sp-cnts table after excluding the DD (but are in the genuine file)
267
268 sprich=with(tbf1, aggregate(Scientific, list(group=Group), lu))
269 names(sprich)[length(sprich)!="rich"]
270 sprich ### number of species per taxon
271
272 years=min(gench$start_year):maxyear
273 years
274
275 groups=unique(tbf$Group)
276 groups
277
278 ##### reconstruct past tables
279 pastables=data.frame()
280 for (g in 1:length(groups))
281 {
282 #g=1

```

```

283 group=as.character(groups[g])
284 #group="Bird"
285 group
286 tbgn=gench[gench$Group==group,]
287 tball=tb1[tb1$Group==group,] ### al sp sp-cnts
288 nrow(tbgn)
289 head(tbgn)
290 yearsa=sort(unique(c(tbgn$start_year,tbgn$end_year)))
291 yearsa
292 yearsa2=yearsa[1:length(yearsa)-1]
293 yearsa2
294 tabsp=with(tabsp1[tabsp1$Scientific%in%tball$Scientific,], data.frame(Scientific=Scientific, iucn=Cat_latest))
295 tabsp$year=max(yearsa)
296 head(tabsp)
297
298 sp1=unique(tball$Scientific)
299 length(sp1)
300 length(sp1[sp1%in%tabsp$Scientific])
301
302 tabsp2=tabsp
303 tabspnew=tabsp
304 for (y in length(yearsa2):1)
305 {
306 #y=1
307 tabsp3=tabspnew
308 tabsp3$year=years2[y]
309 t2ch=tbgn[tbgn$start_year==years2[y],c("Scientific","cat_start")]
310 head(t2ch)
311 nrow(t2ch)
312
313 for (z in 1:nrow(t2ch))
314 {
315 #z=1
316 tabsp3$iucn[tabsp3$Scientific==as.character(t2ch$Scientific[z])]=as.character(t2ch$cat_start[z])
317 }
318 tabspnew=tabsp3

```

```

319  tabsp2=rbind(tabsp2,tabsp3)
320  }
321
322  nrow(tabsp2)/lu(tabsp2$Scientific) ## should be the number of years of comprehensive assessments
323  tabsp2$group=group
324
325  pastables=rbind(pastables,tabsp2)
326  }
327
328  #write.csv(pastables, "pastables.csv", row.names=F)
329  #pastables=read.csv("pastables.csv")
330  wex=wts$wg[wts$iucn=="EX"]
331  head(pastables)
332
333  ##### overlall RLI and estimate left slope for corals
334
335  ddsp=unique(pastables$Scientific[pastables$iucn=="DD"])
336  tabgrli=pastables[!pastables$Scientific%in%ddsp,]
337  head(tabgrli)
338  unique(tabgrli$iucn)
339
340  groupsf=c(as.character(groups[groups!="Coral"]),"Coral") # to ensure that Corals will be the last (to use the mean slope for left extrapolation)
341  tt=proc.time()
342
343  # non-corals
344
345  groups2=as.character(groups[groups!="Coral"])
346  rliTotGlobal_noncorals_list=rlicalc(nreps=repetitions, groups2=groups2, years=years, leftextrapol=T, saveleftslope=T, plotit=F)
347
348  rliTotGlobal_noncorals=rliTotGlobal_noncorals_list[[1]]
349  head(rliTotGlobal_noncorals)
350  tail(rliTotGlobal_noncorals)
351
352  meanleftslopes=rliTotGlobal_noncorals_list[[2]]
353  meanleftslopes
354

```

```

355   ### corals
356
357   rliTotGlobal_corals_list=rlicalc(nreps=repetitions, groups2="Coral", years=years, leftextrapol=F, saveleftslope=F, plotit=F)
358   rliTotGlobal_corals=rliTotGlobal_corals_list[[1]]
359   head(rliTotGlobal_corals)
360   tail(rliTotGlobal_corals)
361
362   rliTotGlobal=rbind(rliTotGlobal_noncorals,rliTotGlobal_corals)
363
364   rlitotagrs=with(rliTotGlobal, aggregate(rli,list(year=year, nrep=nrep), mean))
365   names(rlitotagrs)[length(rlitotagrs)=="rli"]
366   rlitotagrs$group="aggregated"
367   head(rlitotagrs)
368
369   rliTotF=rbind(rliTotGlobal,rlitotagrs[,c("group","year","nrep","rli")])
370   head(rliTotF)
371   tail(rliTotF)
372   max(rliTotF$nrep)
373   max(rliTotF$rli)
374
375   resg=with(rliTotF, aggregate(rli, list(year=year, group=group), length))
376   resg$rli=with(rliTotF, aggregate(rli, list(year=year, group=group), median))$x
377   resg$qn95=with(rliTotF, aggregate(rli, list(year=year, group=group), q95))$x
378   resg$qn05=with(rliTotF, aggregate(rli, list(year=year, group=group), q05))$x
379   resg
380
381   gs=unique(resg$group)
382   resgf=data.frame()
383   for (g in 1:lu(gs))
384   {
385     #g=1
386     gs1=as.character(gs[g])
387     gs1
388     if (gs1!="aggregated") yrs=min(gench$start_year[gench$Group==gs1]):max(gench$end_year[gench$Group==gs1]) else yrs=(max(with(gench, aggregate(start_year,
389     list(Group), min))$x)-10):maxyear
390     rli3=resg[resg$group==gs1&resg$year%in%yrs,]

```

```

391 rli3=rli3[order(rli3$year),]
392 resgf=rbind(resgf,rli3)
393 }
394 head(resgf,20)
395
396 tab2plot=resgf
397
398 par(las=1)
399 with(tab2plot,plot(year,rli, ylim=c(min(qn05), max(qn95)), xlim=c(min(year), max(year)+4), type="n", xlab="Year", ylab="Red List Index of species survival"))
400
401 gs=unique(tab2plot$group[tab2plot$group!="aggregated"])
402 lu(gs)
403 for (g in 1:length(gs))
404 {
405 #g=1
406 gs1=as.character(gs[g])
407 res2=tab2plot[tab2plot$group==gs1,]
408 with(res2,polygon(c(year,rev(year)),c(qn05,rev(qn95)), col=cinz, border=cinz))
409 }
410 with(tab2plot[tab2plot$group=="aggregated",], polygon(c(year,rev(year)),c(qn05,rev(qn95)), col=cinz, border=cinz))
411
412 for (g in 1:length(gs))
413 {
414 gs1=as.character(gs[g])
415 res2=tab2plot[tab2plot$group==gs1,]
416 with(res2,lines(year,rli, col=2, lwd=2))
417 text(max(res2$year)+0.5, res2$rli[nrow(res2)],paste(gs1,"s", sep=""), cex=.6, adj=c(0,0.5))
418 }
419 with(tab2plot[tab2plot$group=="aggregated",], lines(year,rli, col=4, lwd=2))
420 #abline(v=2015, lty=3, col=8)
421 #write.csv(resgf, "global_RLIs.csv", row.names=F)
422 #pastables=read.csv("pastables.csv")
423 #meanleftslopes=(-0.0009302144)
424
425 #####
426 ### pdf with global rlis

```

```

427
428 rlinew=resgf
429 pdfs=T
430 pdf1name=paste(finfolder2, "/", "global", "_all_taxa.pdf", sep="")
431 pdf1name
432 if (pdfs) pdf(file=pdf1name, width = 12, height = 10)
433 if (pdfs) par(las=1,cex.axis=1.2, cex.lab=1.3, cex.main=1.5, mar=c(5,5,4,2), mgp=c(3.5, 1, 0))
434
435 #graph all
436 miny=.15
437 topy=min(c(1.02, max(rlinew$qn95)*1.1))
438 topy
439 basy=min(rlinew$qn05)*.99
440 difrgy=diff(range(basy, topy))
441 if (difrgy<miny) {
442   basy=topy-miny
443   difrgy=diff(range(basy, topy))}
444
445 with(rlinew,plot(year,rli, ylim=c(basy*.99,topy), xlim=c(min(year), max(year)+1), type="n", xlab="Year", yaxt="n", ylab="Red List Index of species survival"))
446 #with(rlinew,plot(year,rli, ylim=c(0,topy), type="n", xlab="Year", yaxt="n", ylab="Red List Index of species survival", main=namect1))
447 if (difrgy<0.3) axis(2, seq(0,1, by=.05))
448 if (difrgy>0.3) axis(2, seq(0,1, by=.1))
449
450 gs=unique(rlinew$group[rlinew$group!="aggregated"])
451 lu(gs)
452 for (g in 1:length(gs))
453 {
454   #g=1
455   gs1=as.character(gs[g])
456   res2=rlinew[rlinew$group==gs1,]
457   with(res2,polygon(c(year,rev(year)),c(qn05,rev(qn95)), col=cinz, border=cinz))
458 }
459 with(rlinew[rlinew$group=="aggregated",], polygon(c(year,rev(year)),c(qn05,rev(qn95)), col=cinz, border=cinz))
460
461 for (g in 1:length(gs))
462 {

```

```

463 gs1=as.character(gs[g])
464 res2=rlinew[rlinew$group==gs1,]
465 with(res2,lines(year,rli, col=2, lwd=2))
466 if (pdfs) text(max(res2$year)+0.5, res2$rli[nrow(res2)],paste(gs1,"s", sep=""), cex=1, adj=c(0,0.5))
467 }
468 with(rlinew[rlinew$group=="aggregated",], lines(year,rli, col=4, lwd=2))
469
470 if (pdfs) dev.off()
471
472 #graph aggregated
473 pdf2name=paste(finfolder2,"/", "global","_aggregated.pdf", sep="")
474 pdf2name
475 if (pdfs) pdf(file=pdf2name, width = 12, height = 10)
476 if (pdfs) par(las=1,cex.axis=1.2, cex.lab=1.3, cex.main=1.5, mar=c(5,5,4,2), mgp=c(3.5, 1, 0))
477
478 ttuse=rlinew[rlinew$group=="aggregated",]
479 topy=min(c(1.02, max(ttuse$qn95)*1.1))
480 basy=min(ttuse$qn05)*.99
481 difrgy=diff(range(basy, topy))
482 if (difrgy<miny) {
483 basy=topy-miny
484 difrgy=diff(range(basy, topy))}
485
486 with(ttuse,plot(year,rli, yaxt="n", ylim=c(basy,topy), type="n", xlab="Year", ylab="Red List Index of species survival"))
487 with(ttuse, polygon(c(year,rev(year)),c(qn05,rev(qn95)), col=cinz, border=cinz))
488 with(ttuse, lines(year,rli, col=4, lwd=2))
489 if (difrgy<0.3) axis(2, seq(0,1, by=.05))
490 if (difrgy>0.3) axis(2, seq(0,1, by=.1))
491
492 if (pdfs) dev.off()
493
494 # graph birds
495 pdf3name=paste(finfolder2,"/", "global","_birds.pdf", sep="")
496 pdf3name
497 if (pdfs) pdf(file=pdf3name, width = 12, height = 10)
498 if (pdfs) par(las=1,cex.axis=1.2, cex.lab=1.3, cex.main=1.5, mar=c(5,7,4,2), mgp=c(5, 1, 0))

```

```

499
500 ttuse=rlinew[rlinew$group=="Bird",]
501
502 if (nrow(ttuse)==0) plot(1:10, 1:10, type="n", main="no data for birds", xaxt="n", yaxt="n", xlab="", ylab="")
503 if (nrow(ttuse)>0){
504   miny=.05
505   topy=min(c(1.02, max(ttuse$qn95)*1.001))
506   basy=min(ttuse$qn05)*.999
507   difrgy=diff(range(basy, topy))
508   #if (difrgy<miny) {
509   #basy=topy-miny
510   #difrgy=diff(range(basy, topy))}
511
512   with(ttuse,plot(year,rli, ylim=c(basy,topy), yaxt="n", type="n", xlab="", ylab="Red List Index of species survival"))
513   with(ttuse, polygon(c(year,rev(year)),c(qn05,rev(qn95)), col=cinz, border=cinz))
514   with(ttuse, lines(year,rli, col=2, lwd=2))
515   if (difrgy<0.3)
516   {
517   if (difrgy<0.1) {axis(2, seq(0,1, by=.002))} else axis(2, seq(0,1, by=.05))
518   }
519   if (difrgy>0.3) axis(2, seq(0,1, by=.1))
520   }
521   par(mgp=c(3.5, 1, 0))
522   title(xlab="Year")
523
524   if (pdfs) dev.off()
525
526   #####
527   ### rlis per country
528
529   tt=proc.time()
530
531   nreps=repetitions
532   plotit=F
533   plotit2=T
534   finalrlis=data.frame()

```

```

535
536 cts=unique(tbf1$isof)
537 lu(cts)
538 cts=cts[cts!="0"]
539 cts
540 tt=proc.time()
541
542 for (x in 1:length(cts)) #length(cts)
543 {
544 #x=1
545 #x=which(cts=="ABW")
546 cnt=cts[x]
547 cnt
548 print(paste(x, cnt))
549
550 tbct=tbf1[tbf1$isof==cnt,]
551
552 groupsf=as.character(unique(tbct$Group))
553 groupsf
554
555 rliext=data.frame(country=cnt, group=rep(rep(groupsf, each=length(years)),nreps), year=rep(rep(years, length(groupsf)),nreps),
556 nrep=rep(1:nreps,each=length(years)*length(groupsf)), rli=NA)
557 head(rliext)
558 tail(rliext)
559
560 for (r in 1: nreps) #nreps
561 {
562 print(paste(x, cnt, r))
563 #r=1
564
565 for (g in 1:length(groupsf)) ### note that loop of groups has to be the first, to save the slopes to apply to corals
566 {
567 #g=1
568 group=as.character(groupsf[g])
569 group
570

```

```
571 spcnt=tbct[tbct$Group==group,]
572
573 lu(spcnt$Scientific)
574 nrow(spcnt)
575
576 spcnt2=with(spcnt, aggregate(pp, list(Scientific=Scientific), sum)) ## sum pp for species and country (could be split for marine/terrestrial or disjunct polygons in
577 country/eez layer with multiple entries)
578 nrow(spcnt2)
579 names(spcnt2)[length(spcnt2)]= "pp"
580 head(spcnt2)
581 max(spcnt2$pp)
582 spcnt2[spcnt2$pp==1,] ## endemic species
583
584 spcnt3=merge(spcnt2,pastables)
585 head(spcnt3)
586 tail(spcnt3)
587 nrow(spcnt3)/nrow(spcnt2) ## should be the number of years of assessment for this group
588
589 yearsa=sort(unique(spcnt3$year))
590 yearsa
591 yearsa2=yearsa[1:length(yearsa)-1]
592 yearsa2
593
594 ### years of assessment
595
596 rlig=data.frame()
597 for (y in 1:length(yearsa))
598 {
599 #y=1
600 year=yearsa[y]
601 year
602
603 tb=spcnt3[spcnt3$year==year,]
604 head(tb)
605 nrow(tb)
606
```

```

607 unique(tb$iucl)
608
609 tb[tb$iucl=="DD",]
610 tb[!tb$iucl%in%c("DD", "LC"),]
611 tb[tb$iucl=="LC",]
612
613 tb=merge(tb,wts)
614 head(tb)
615 nrow(tb)
616 dds=tb$Scientific[tb$iucl=="DD"]
617 dds
618 tb[tb$iucl=="DD",]
619 allwgs=tb$wgt[tb$wgt>(-1)]
620 if (length(allwgs)==0) allwgs=0
621 if (length(dds)>0) for (d in 1:length(dds)) tb$wgt[tb$Scientific==dds[d]]=sample(allwgs,1)
622 tb[tb$iucl=="DD",]
623
624 tb$wxpp=tb$wgt*tb$pp
625 #tb$wxpp=5*tb$pp
626 head(tb)
627
628 sum(tb$wxpp)
629 wex*sum(tb$pp)
630 sum(tb$wxpp)/(wex*sum(tb$pp))
631 1-round(sum(tb$wxpp)/(wex*sum(tb$pp)),7)
632 rli1=1-round((sum(tb$wxpp)/(wex*sum(tb$pp))),7)
633 rli1
634 crli=round((sum(tb$wxpp)/(wex*sprich$rich[sprich$group==group])),7)
635 rlig=rbind(rlig, data.frame(group=group,year=year, rli=rli1))
636
637 if (y==1) tb1=tb
638
639 } ## ends loop year
640 rlig
641 dat=rlig
642 dat

```

```

643
644 #interpolation
645 yearsti=min(dat$year): max(dat$year)
646 irli=approx(dat$year, dat$rli, xout=yearsti, method="linear", rule=2:2)$y
647 irli
648
649 datf=data.frame(year=yearsti, rli=irli)
650 datf
651 #with(datf, plot(year, rli, xlim=c(1975,2018), ylim=c(0.6,1.2)))
652
653 #extrapolation # y=a+bx
654 #left
655 yearl=c(min(years)-2, min(years)-1, years[which(years<min(yearsti))])
656 yearl
657
658 x1=sort(dat$year)[1:2]
659 x1 ### years
660 y1=c(dat$rli[dat$year==x1[1]],dat$rli[dat$year==x1[2]])
661 y1 ###rlis
662 nv=lm(y1~x1)
663 mnslp <- coef(nv)[2]
664 useslp <- rnorm(1, mnslp, sd=slopecv*abs(mnslp))
665 if (group=="Coral") useslp <- meanleftslopes
666 irlel=y1[1] - (useslp*(x1[1] -(yearl)))
667
668 datf=rbind(datf, data.frame(year=yearl, rli=irlel))
669 datf
670
671 #right
672 yearr=c(years[which(years>max(yearsti))],max(years)+1,max(years)+2)
673 yearr
674
675 x1=sort(dat$year)[(nrow(dat)-1):nrow(dat)]
676 x1 ## years
677 y1=c(dat$rli[dat$year==x1[1]],dat$rli[dat$year==x1[2]])
678 y1 ### rlis

```

```

679  nv=lm(y1~x1)
680  mns1p <- coef(nv)[2]
681  uses1p <- rnorm(1, mns1p, sd=slopecv*abs(mns1p))
682  ir1er=y1[2] - (uses1p*(x1[2] -(yearr)))
683  datf=rbind(datf, data.frame(year=yearr, rli=ir1er))
684  datf=datf[order(datf$year),]
685
686  if (plotit)
687  {
688  win.graph()
689  with(datf, plot(year, rli, pch=19, col=2, main=paste(group,"s", sep="")))
690  with(datf[datf$year%in%yearsti,], points(year, rli, pch=19, col=4))
691  }
692
693  restmw=NULL
694  for (m in 3:(nrow(datf)-2)) restmw=c(restmw, sample(datf$rli[(m-2):(m+2)],1))
695  finalmw=restmw
696  rliext[rliext$group==group&rliext$nrep==r,]$rli=finalmw
697  if (plotit) with(rliext[rliext$group==group&rliext$nrep==r,], lines(year, rli, col=8))
698  } # ends loop group
699  } # ends loop rep
700
701  head(rliext)
702  tail(rliext)
703  max(rliext$nrep)
704  max(rliext$rli)
705
706  rlitotags=with(rliext, aggregate(rli,list(year=year, nrep=nrep), mean))
707  names(rlitotags)[length(rlitotags)]= "rli"
708  rlitotags$group="aggregated"
709  rlitotags$country=cnt
710  head(rlitotags)
711
712  rliextF=rbind(rliext,rlitotags[,c("country","group","year","nrep","rli")])
713  head(rliextF)
714  tail(rliextF)

```

```

715 max(rliextF$nrep)
716 max(rliextF$rli)
717
718 resg=with(rliextF, aggregate(rli, list(year=year, group=group), length))
719 resg$rli=with(rliextF, aggregate(rli, list(year=year, group=group), median))$x
720 resg$qn95=with(rliextF, aggregate(rli, list(year=year, group=group), q95))$x
721 resg$qn05=with(rliextF, aggregate(rli, list(year=year, group=group), q05))$x
722 head(resg)
723
724 resg$rli[resg$rli>1]=1
725 resg$rli[resg$rli<0]=0
726 resg$qn95[resg$qn95>1]=1
727 resg$qn95[resg$qn95<0]=0
728 resg$qn05[resg$qn05>1]=1
729 resg$qn05[resg$qn05<0]=0
730
731 gs=unique(resg$group)
732 resgf=data.frame()
733 for (g in 1:lu(gs))
734 {
735 #g=1
736 gs1=as.character(gs[g])
737 gs1
738 if (gs1!="aggregated") yrs=min(gench$start_year[gench$Group==gs1]):max(gench$end_year[gench$Group==gs1]) else yrs=(max(with(gench, aggregate(start_year,
739 list(Group), min))$x)-10):maxyear
740 rli3=resg[resg$group==gs1&resg$year%in%yrs,]
741 rli3=rli3[order(rli3$year),]
742 resgf=rbind(resgf,rli3)
743 }
744 head(resgf,30)
745
746 final=resg
747 final$country=cnt
748 head(final)
749 finalrlis=rbind(finalrlis,final)
750

```

```

751 final2save=final[, c("country","group","year","rli","qn05","qn95")]
752 finalname=paste(finfolder, "/", cnt, ".csv", sep="")
753 finalname
754 write.csv(final2save,finalname, row.names=F)
755
756 if (plotit2){
757   if (div0(x,6)==0)
758   {
759     win.graph(50,30)
760     par(mfrow=c(2,3), mar=c(2,2,2,1))
761   }
762
763   tab2plot=resgf
764   with(tab2plot,plot(year,rli, ylim=c(min(qn05), max(qn95)), xlim=c(min(year), max(year)+4), type="n", xlab="Year", ylab="Red List Index of species survival", main=cnt))
765
766   gs=unique(tab2plot$group[tab2plot$group!="aggregated"])
767   lu(gs)
768   for (g in 1:length(gs))
769   {
770     #g=1
771     gs1=as.character(gs[g])
772     res2=tab2plot[tab2plot$group==gs1,]
773     with(res2,polygon(c(year,rev(year)),c(qn05,rev(qn95)), col=cinz, border=cinz))
774   }
775   with(tab2plot[tab2plot$group=="aggregated",], polygon(c(year,rev(year)),c(qn05,rev(qn95)), col=cinz, border=cinz))
776
777   for (g in 1:length(gs))
778   {
779     gs1=as.character(gs[g])
780     res2=tab2plot[tab2plot$group==gs1,]
781     with(res2,lines(year,rli, col=2, lwd=2))
782     text(max(res2$year)+0.5, res2$rli[nrow(res2)],paste(gs1,"s", sep=""), cex=.6, adj=c(0,0.5))
783   }
784   with(tab2plot[tab2plot$group=="aggregated",], lines(year,rli, col=4, lwd=2))
785 }
786 } # ends loop country

```

```

787
788 print((proc.time()-tt)[1]/60) ## time in minutes
789
790 str(finalrlis)
791 head(finalrlis)
792 lu(finalrlis$country)
793 max(finalrlis$x)
794 max(finalrlis$rli)
795 min(finalrlis$rli)
796
797 #####
798
799 ### combine all files in 1 (in case the country files are calculated in different machines, to save time)
800 cmbfiles=F
801
802 if(cmbfiles){
803 allfics=data.frame()
804 fics=dir("//AZ-FILESERVER2/gis_data/Tracking Ocean Wanderers/IUCN_R_scripts/RLI/country_files/ISO_BL_test/", pattern = ".csv", full.names =T) ### copy paste here the
805 folder with the countries' files
806 for (u in 1:length(fics)) allfics=rbind(allfics, read.csv(fics[u]))
807 lu(allfics$country)
808 finalrlis=allfics
809 }
810
811 str(finalrlis)
812 head(finalrlis)
813 lu(finalrlis$country)
814 max(finalrlis$rli)
815
816
817 #####
818 ## country final files and graphs
819 iso2use
820 sumsp=with(tbf1, aggregate(pp, list(country=isof, group=Group), sum))
821 names(sumsp)[length(sumsp)]= "sumpp"
822 head(sumsp)

```

```
823 sum(sumspp$sumpp)
824
825 rliextf=finalrlis
826 min(rliextf$rli)
827 max(rliextf$rli)
828 head(rliextf)
829 tail(rliextf)
830
831 pdfs=T
832 plotit=T
833 writefiles=T
834 if (pdfs) plotit=F
835
836 allcountries_RLI=data.frame()
837 allcountries_CRLI=data.frame()
838
839 cts=unique(rliextf$country)
840 length(cts)
841
842 #graphics.off()
843 for (i in 1:length(cts)) #length(cts)
844 {
845 #i=1
846 #i=which(cts=="ABW")
847 i
848
849 ct1=as.character(cts[i])
850 ct1
851 rli2=rliextf[rliextf$country==ct1,]
852
853 namect1=as.character(ct1)
854 if (iso2use=="ISO_BL") namect1=paste(isos$uname_BL[isos$isof==as.character(ct1)][1], "(", ct1, ")", sep="")
855 if (iso2use=="ISO_SDG") namect1=paste(isos$uname_SDG[isos$isof==as.character(ct1)][1], "(", ct1, ")", sep="")
856 namect1
857
858
```

```

859 gs=unique(rli2$group)
860 lu(gs)
861
862 rlinew=data.frame()
863 crlis=data.frame()
864 for (g in 1:lu(gs))
865 {
866 #g=1
867 gs1=as.character(gs[g])
868 gs1
869 if (gs1!="aggregated") yrs=min(gech$start_year[gech$Group==gs1]):max(gech$end_year[gech$Group==gs1]) else yrs=(max(with(gech, aggregate(start_year,
870 list(Group), min))$x)-10):maxyear
871 rli3=rli2[rli2$group==gs1&rli2$year%in%yrs,]
872 rli3=rli3[order(rli3$year),]
873
874 if (gs1!="aggregated")
875 {
876 crli1=rli2[rli2$group==gs1,]
877 crli1$crli=crli1$rli*sumspp$sumpp[sumspp$group==gs1&sumspp$country==ct1][1]/sprich$rich[sprich$group==gs1]
878 crli1$crli05=crli1$qn05*sumspp$sumpp[sumspp$group==gs1&sumspp$country==ct1][1]/sprich$rich[sprich$group==gs1]
879 crli1$crli95=crli1$qn95*sumspp$sumpp[sumspp$group==gs1&sumspp$country==ct1][1]/sprich$rich[sprich$group==gs1]
880 crlis=rbind(crlis,crli1[,c("country","group","year","crli","crli05","crli95")])
881 }
882 rlinew=rbind(rlinew,rli3)
883 }
884 rlinew
885 tail(crlis)
886
887 crlisag=with(crlis, aggregate(crli, list(country=country, year=year), mean))
888
889 names(crlisag)[length(crlisag)]= "crli"
890 crlisag$group="aggregated"
891 crlisag$crli05=with(crlis, aggregate(crli05, list(country=country, year=year), mean))$x
892 crlisag$crli95=with(crlis, aggregate(crli95, list(country=country, year=year), mean))$x
893
894 yrs=(max(with(gech, aggregate(start_year, list(Group), min))$x)-10):maxyear

```

```

895   crlisag=crlisag[crlisag$year%in%yrs,]
896   crlisag
897   crlisag[,c("country","group","year","crli","crli05","crli95")]
898
899   crlisf=data.frame()
900   gs2=unique(crlis$group)
901   for (g in 1:length(gs2))
902   {
903     gs1=as.character(gs2[g])
904     yrs=min(gench$start_year[gench$Group==gs1]):max(gench$end_year[gench$Group==gs1])
905     crlisf=rbind(crlisf, crlis[crlis$group==gs1&crlis$year%in%yrs,])
906   }
907   crlisf=rbind(crlisf,crlisag)
908
909   rlinew2save=rlinew[, c("country","group","year","rli","qn05","qn95")]
910   fnamect=paste(finfolder2, "/", ct1, ".csv", sep="")
911   fnamect
912   if (writefiles) write.csv(rlinew2save,fnamect, row.names=F)
913
914   fnamect2=paste(finfolder2, "/", ct1, "_crli.csv", sep="")
915   fnamect2
916   if (writefiles) write.csv(crlisf,fnamect2, row.names=F)
917
918   allcountries_RLI=rbind(allcountries_RLI,rlinew2save)
919   allcountries_CRLI=rbind(allcountries_CRLI,crlisf)
920
921   if (plotit) {
922     if (div0(i,4)==0)
923     {
924       win.graph(30,20)
925       par(las=1, mfrow=c(4,4))
926     }
927   }
928
929   pdf1name=paste(finfolder2, "/", ct1, "_all_taxa.pdf", sep="")
930   pdf1name

```

```

931 if (pdfs) pdf(file=pdf1name, width = 12, height = 10)
932 if (pdfs) par(las=1,cex.axis=1.2, cex.lab=1.3, cex.main=1.5, mar=c(5,5,4,2), mgp=c(3.5, 1, 0))
933
934 #graph all
935 miny=.15
936 topy=min(c(1.02, max(rlinew$qn95)*1.1))
937 topy
938 basy=min(rlinew$qn05)*.99
939 difrgy=diff(range(basy, topy))
940 if (difrgy<miny) {
941 basy=topy-miny
942 difrgy=diff(range(basy, topy))}
943
944 with(rlinew,plot(year,rli, ylim=c(basy*.99,topy), xlim=c(min(year), max(year)+1), type="n", xlab="Year", yaxt="n", ylab="Red List Index of species survival"))
945 if (plotit) title(main=namect1)
946 #with(rlinew,plot(year,rli, ylim=c(0,topy), type="n", xlab="Year", yaxt="n", ylab="Red List Index of species survival", main=namect1))
947 if (difrgy<0.3) axis(2, seq(0,1, by=.05))
948 if (difrgy>0.3) axis(2, seq(0,1, by=.1))
949
950 gs=unique(rlinew$group[rlinew$group!="aggregated"])
951 lu(gs)
952 for (g in 1:length(gs))
953 {
954 #g=1
955 gs1=as.character(gs[g])
956 res2=rlinew[rlinew$group==gs1,]
957 with(res2,polygon(c(year,rev(year)),c(qn05,rev(qn95)), col=cinz, border=cinz))
958 }
959 with(rlinew[rlinew$group=="aggregated",], polygon(c(year,rev(year)),c(qn05,rev(qn95)), col=cinz, border=cinz))
960
961 for (g in 1:length(gs))
962 {
963 gs1=as.character(gs[g])
964 res2=rlinew[rlinew$group==gs1,]
965 with(res2,lines(year,rli, col=2, lwd=2))
966 #res22=rli2[rli2$group==gs1,]

```

```

967 #with(res22,lines(year,rli, col=2, lwd=1))
968 if (plotit) text(max(res2$year)+0.5, res2$rli[nrow(res2)],paste(gs1,"s", sep=""), cex=.6, adj=c(0,0.5))
969 if (pdfs) text(max(res2$year)+0.5, res2$rli[nrow(res2)],paste(gs1,"s", sep=""), cex=1, adj=c(0,0.5))
970 }
971 with(rlinew[rlinew$group=="aggregated",], lines(year,rli, col=4, lwd=2))
972
973 if (pdfs) dev.off()
974
975 #graph aggregated
976 pdf2name=paste(finfolder2, "/", ct1, "_aggregated.pdf", sep="")
977 pdf2name
978 if (pdfs) pdf(file=pdf2name, width = 12, height = 10)
979 if (pdfs) par(las=1,cex.axis=1.2, cex.lab=1.3, cex.main=1.5, mar=c(5,5,4,2), mgp=c(3.5, 1, 0))
980
981 ttuse=rlinew[rlinew$group=="aggregated",]
982 topy=min(c(1.02, max(ttuse$qn95)*1.1))
983 basy=min(ttuse$qn05)*.99
984 difrgy=diff(range(basy, topy))
985 if (difrgy<miny) {
986 basy=topy-miny
987 difrgy=diff(range(basy, topy))}
988
989 with(ttuse,plot(year,rli, yaxt="n", ylim=c(basy,topy), type="n", xlab="Year", ylab="Red List Index of species survival"))
990 with(ttuse, polygon(c(year,rev(year)),c(qn05,rev(qn95)), col=cinz, border=cinz))
991 with(ttuse, lines(year,rli, col=4, lwd=2))
992 if (difrgy<0.3) axis(2, seq(0,1, by=.05))
993 if (difrgy>0.3) axis(2, seq(0,1, by=.1))
994
995 if (pdfs) dev.off()
996
997 # graph birds
998 pdf3name=paste(finfolder2,"/", ct1, "_birds.pdf", sep="")
999 pdf3name
1000 if (pdfs) pdf(file=pdf3name, width = 12, height = 10)
1001 if (pdfs) par(las=1,cex.axis=1.2, cex.lab=1.3, cex.main=1.5, mar=c(5,5,4,2), mgp=c(3.5, 1, 0))
1002

```

```

1003 ttuse=rlinew[rlinew$group=="Bird",]
1004
1005 if (nrow(ttuse)==0) plot(1:10, 1:10, type="n", main="no data for birds", xaxt="n", yaxt="n", xlab="", ylab="")
1006 if (nrow(ttuse)>0){
1007 topy=min(c(1.02, max(ttuse$qn95)*1.1))
1008 basy=min(ttuse$qn05)*.99
1009 difrgy=diff(range(basy, topy))
1010 if (difrgy<miny) {
1011 basy=topy-miny
1012 difrgy=diff(range(basy, topy))}
1013
1014 with(ttuse,plot(year,rli, yaxt="n", ylim=c(basy,topy), type="n", xlab="Year", ylab="Red List Index of species survival"))
1015 with(ttuse, polygon(c(year,rev(year)),c(qn05,rev(qn95)), col=cinz, border=cinz))
1016 with(ttuse, lines(year,rli, col=2, lwd=2))
1017 if (difrgy<0.3) axis(2, seq(0,1, by=.05))
1018 if (difrgy>0.3) axis(2, seq(0,1, by=.1))
1019 }
1020 if (pdfs) dev.off()
1021
1022
1023 ### crli
1024
1025 pdf3name=paste(finfolder2, "/", ct1, "_crli.pdf", sep="")
1026 pdf3name
1027 if (pdfs) pdf(file=pdf3name, width = 12, height = 10)
1028 if (pdfs) par(las=1,cex.axis=1.2, cex.lab=1.3, cex.main=1.5, mar=c(5,6,4,2), mgp=c(4.5, 1, 0))
1029
1030 with(crlisf,plot(year,crli, ylim=c(min(crli05),max(crli05)), xlim=c(min(year), max(year)+1), type="n", xlab="", ylab="National contribution to the global RLI"))
1031 par(mgp=c(3.5, 1, 0))
1032 title(xlab="Year")
1033
1034 gs=unique(crlisf$group[crlisf$group!="aggregated"])
1035 lu(gs)
1036 for (g in 1:length(gs))
1037 {
1038 #g=1

```

```

1039 gs1=as.character(gs[g])
1040 res2=crlisf[crlisf$group==gs1,]
1041 with(res2,polygon(c(year,rev(year)),c(crli05,rev(crli95)), col=cinz, border=cinz))
1042 }
1043 with(crlisf[crlisf$group=="aggregated",], polygon(c(year,rev(year)),c(crli05,rev(crli95)), col=cinz, border=cinz))
1044
1045 for (g in 1:length(gs))
1046 {
1047 gs1=as.character(gs[g])
1048 res2=crlisf[crlisf$group==gs1,]
1049 with(res2,lines(year,crli, col=2, lwd=2))
1050 #res22=rli2[rli2$group==gs1,]
1051 #with(res22,lines(year,rli, col=2, lwd=1))
1052 if (plotit) text(max(res2$year)+0.5, res2$crli[nrow(res2)],paste(gs1,"s", sep=""), cex=.6, adj=c(0,0.5))
1053 if (pdfs) text(max(res2$year)+0.5, res2$crli[nrow(res2)],paste(gs1,"s", sep=""), cex=1, adj=c(0,0.5))
1054 }
1055 with(crlisf[crlisf$group=="aggregated",], lines(year,crli, col=4, lwd=2))
1056
1057 if (pdfs) dev.off()
1058
1059 }
1060 # ends loop countries
1061
1062 lu(allcountries_RLI$country)
1063 lu(allcountries_CRLI$country)
1064
1065 rliallcntsfile=paste(finfolder2,"/", "allcountries_RLI.csv", sep="")
1066 if (writefiles) write.csv(allcountries_RLI,rliallcntsfile, row.names=F)
1067
1068 crliallcntsfile=paste(finfolder2,"/", "allcountries_CRLI.csv", sep="")
1069 if (writefiles) write.csv(allcountries_CRLI,crliallcntsfile, row.names=F)
1070
1071 ##### single pdf with all plots
1072
1073 pdfallname=paste(finfolder2,"/", "RLI_all_countries.pdf", sep="")
1074 pdfallname

```

```

1075 pdf(file=pdfallname, width = 12, height = 10)
1076 par(las=1,cex.axis=1.2, cex.lab=1.3, cex.main=1.5, mar=c(5,5,4,2), mgp=c(3.5, 1, 0))
1077
1078 cts=unique(allcountries_RLI$country)
1079 for (i in 1:length(cts)) #length(cts)
1080 {
1081   #i=1
1082   #i=which(cts=="ABW")
1083   i
1084
1085   ct1=as.character(cts[i])
1086   ct1
1087   rli2=rliextf[rliextf$country==ct1,]
1088
1089   namect1=as.character(ct1)
1090   if (iso2use=="ISO_BL") namect1=paste(isos$uname_BL[isos$isof==as.character(ct1)][1], "(", ct1, ")", sep="")
1091   if (iso2use=="ISO_SDG") namect1=paste(isos$uname_SDG[isos$isof==as.character(ct1)][1], "(", ct1, ")", sep="")
1092   namect1
1093
1094   fspct=pastables[pastables$Scientific%in%unique(tbf1$Scientific[tbf1$isof==ct1]),]
1095   head(fspct)
1096   respct=with(fspct, aggregate(year, list(iucn=iucn, year=year, group=group), length))
1097
1098   rlinew=allcountries_RLI[allcountries_RLI$country==ct1,]
1099
1100
1101   #graph all
1102   miny=.15
1103   topy=min(c(1.02, max(rlinew$qn95)*1.1))
1104   topy
1105   basy=min(rlinew$qn05)*.99
1106   difrgy=diff(range(basy, topy))
1107   if (difrgy<miny) {
1108     basy=topy-miny
1109     difrgy=diff(range(basy, topy))}
1110   with(rlinew,plot(year,rli, ylim=c(basy*.99,topy), xlim=c(min(year), max(year)+1), type="n", xlab="Year", yaxt="n", ylab="Red List Index of species survival", main=namect1))

```

```
1111 if (difrgy<0.3) axis(2, seq(0,1, by=.05))
1112 if (difrgy>0.3) axis(2, seq(0,1, by=.1))
1113 gs=unique(rlinew$group[rlinew$group!="aggregated"])
1114 lu(gs)
1115 for (g in 1:length(gs))
1116 {
1117   #g=1
1118   gs1=as.character(gs[g])
1119   res2=rlinew[rlinew$group==gs1,]
1120   with(res2,polygon(c(year,rev(year)),c(qn05,rev(qn95)), col=cinz, border=cinz))
1121 }
1122 with(rlinew[rlinew$group=="aggregated",], polygon(c(year,rev(year)),c(qn05,rev(qn95)), col=cinz, border=cinz))
1123 for (g in 1:length(gs))
1124 {
1125   gs1=as.character(gs[g])
1126   res2=rlinew[rlinew$group==gs1,]
1127   with(res2,lines(year,rli, col=2, lwd=2))
1128   text(max(res2$year)+0.5, res2$rli[nrow(res2)],paste(gs1,"s", sep=""), cex=1, adj=c(0,0.5))
1129 }
1130 with(rlinew[rlinew$group=="aggregated",], lines(year,rli, col=4, lwd=2))
1131 }
1132 }
1133 dev.off()
```

