

Ecography

ECOG-04568

Bueno, A. S., Masseli, G. S., Kaefer, I. L. and Peres, C. A. 2019. Sampling design may obscure species–area relationships in landscape-scale field studies. – Ecography doi: [10.1111/ecog.04568](https://doi.org/10.1111/ecog.04568)

Supplementary material

Appendix 1

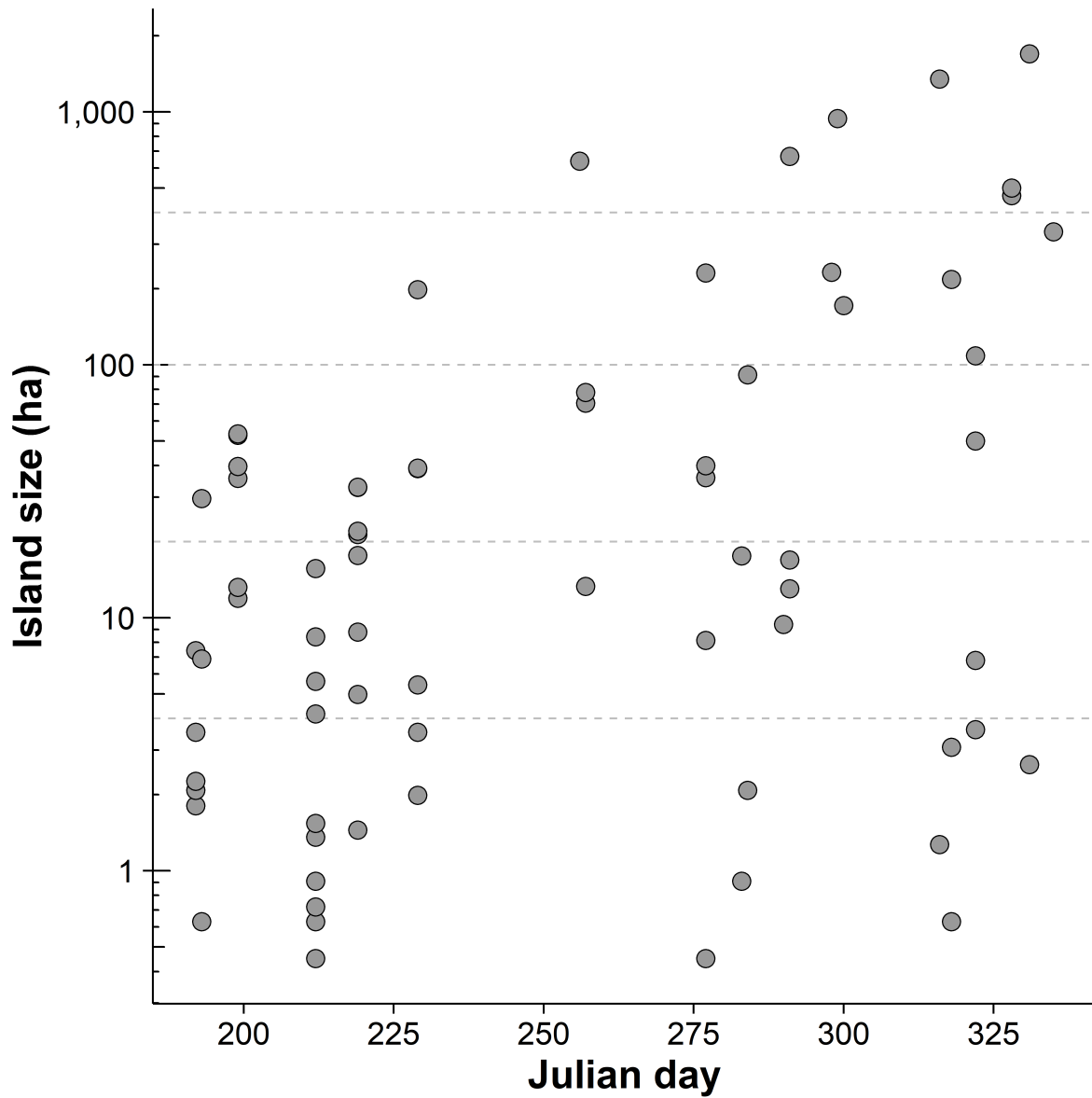


Figure A1 Relationship between island size and Julian day showing the distribution of anuran surveys across island size categories (grey dashed lines; see Fig. 3) throughout the sampling period (11 July to 4 December 2015) at the Balbina Hydroelectric Reservoir landscape.

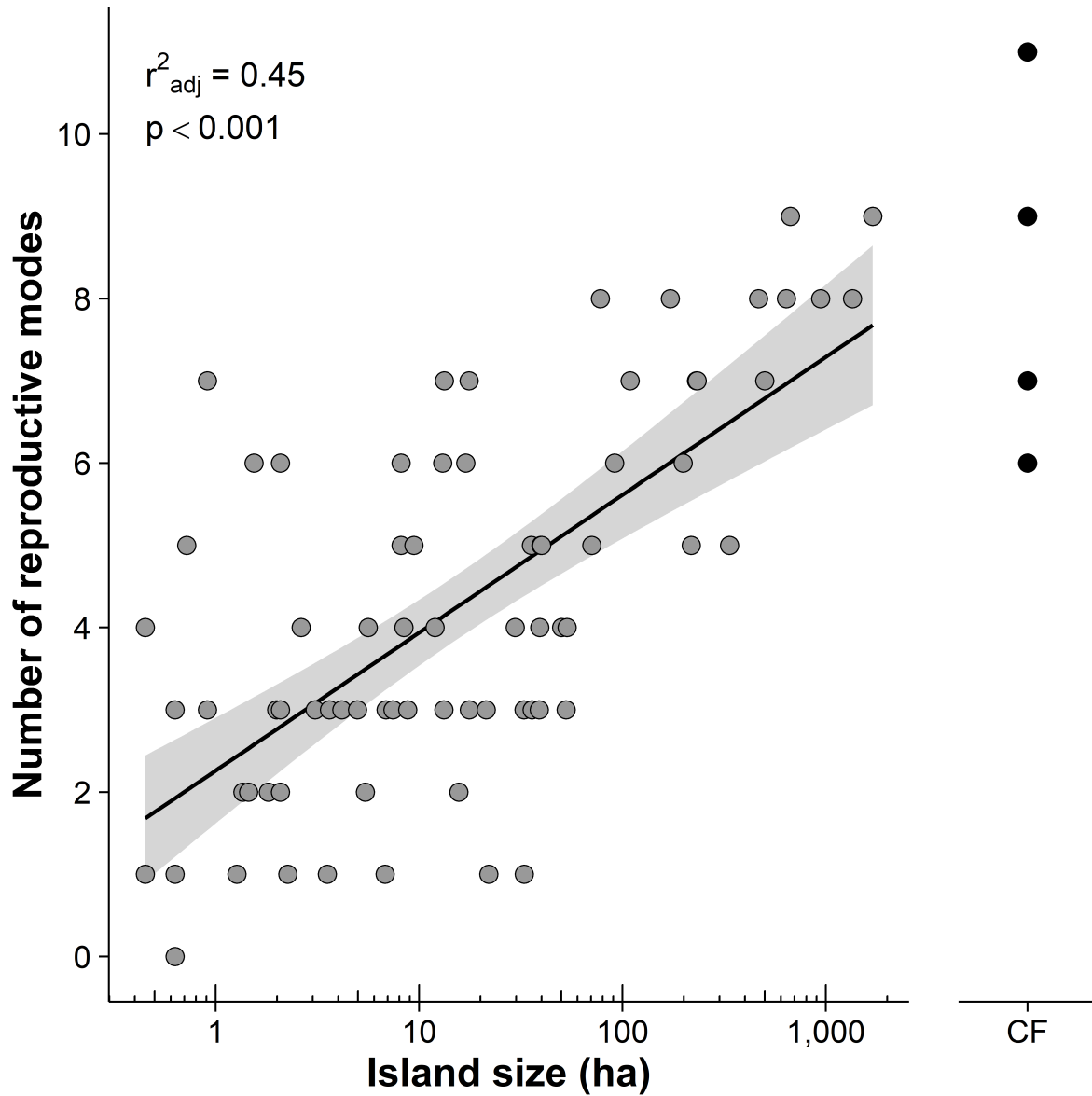


Figure A2 Relationship between the number of reproductive modes represented by at least one anuran species and island size across 74 forest islands (grey circles) surveyed at the Balbina Hydroelectric Reservoir landscape. Continuous forest sites (CF, black circles) were not included in the linear fit.

Table A1 Description of the 78 sites surveyed at the Balbina Hydroelectric Reservoir landscape, including 74 forest islands and four continuous forest (CF) sites. ‘Sampling effort’ corresponds to the number of 1-min recordings.

Site code	Island size (ha)	Number of recorders	Sampling effort	Observed richness	Rarefied richness	Sample coverage
1	0.45	1	62	5	5.000	1.000
2	0.45	1	62	1	1.000	1.000
3	0.63	1	62	2	2.000	1.000
4	0.63	1	59	4	4.000	1.000
5	0.63	1	62	0	—	—
6	0.72	1	62	5	5.000	1.000
7	0.91	1	62	3	3.000	1.000
8	0.91	1	62	9	9.000	1.000
9	1.27	1	62	1	1.000	1.000
10	1.36	1	62	2	2.000	1.000
11	1.45	1	62	2	2.000	1.000
12	1.54	1	62	7	7.000	1.000
13	1.81	1	62	3	3.000	1.000
14	1.99	1	62	3	3.000	1.000
15	2.08	1	62	2	2.000	1.000
16	2.08	1	62	4	4.000	1.000
17	2.08	1	62	9	9.000	1.000
18	2.26	1	62	1	1.000	1.000
19	2.63	1	62	7	7.000	1.000

Site code	Island size (ha)	Number of recorders	Sampling effort	Observed richness	Rarefied richness	Sample coverage
20	3.08	1	62	3	3.000	1.000
21	3.53	1	62	1	1.000	1.000
22	3.53	1	62	3	3.000	0.508
23	3.62	1	62	3	3.000	1.000
24	4.17	1	62	4	4.000	1.000
25	4.98	1	62	3	3.000	1.000
26	5.43	1	62	2	2.000	1.000
27	5.61	1	62	4	4.000	1.000
28	6.79	1	62	1	1.000	1.000
29	6.88	1	62	3	3.000	1.000
30	7.43	1	62	5	5.000	1.000
31	8.15	1	62	7	7.000	0.980
32	8.15	1	62	7	7.000	1.000
33	8.42	1	62	5	5.000	1.000
34	8.78	1	62	3	3.000	1.000
35	9.42	1	62	6	6.000	1.000
36	11.96	1	62	5	5.000	1.000
37	13.04	1	62	7	7.000	0.988
38	13.22	1	60	3	3.000	1.000
39	13.31	1	62	9	9.000	1.000
40	15.67	1	62	2	2.000	1.000
41	16.94	1	62	10	10.000	1.000
42	17.57	1	62	8	8.000	1.000

Site code	Island size (ha)	Number of recorders	Sampling effort	Observed richness	Rarefied richness	Sample coverage
43	17.66	1	62	3	3.000	1.000
44	21.37	1	62	3	3.000	1.000
45	22.01	1	62	1	1.000	1.000
46	29.62	2	124	8	7.253	0.972
47	32.78	2	124	4	3.251	0.928
48	32.87	2	124	1	0.752	0.504
49	35.60	2	93	5	4.996	0.999
50	35.87	2	124	5	4.924	0.991
51	38.94	2	124	3	3.000	1.000
52	39.12	2	124	4	3.927	0.992
53	39.67	2	124	7	5.756	0.948
54	39.94	2	124	9	8.250	0.988
55	50.08	2	124	7	6.249	0.986
56	52.71	2	124	4	3.000	0.982
57	53.35	2	118	5	3.827	0.966
58	70.55	2	124	8	7.490	0.984
59	77.80	2	124	10	9.456	0.993
60	91.30	3	186	8	7.999	1.000
61	108.76	3	186	11	10.483	0.987
62	171.73	3	186	12	10.401	0.986
63	198.52	3	132	8	7.011	0.936
64	217.63	3	186	13	10.618	0.977
65	230.70	3	186	9	8.421	0.993

Site code	Island size (ha)	Number of recorders	Sampling effort	Observed richness	Rarefied richness	Sample coverage
66	232.49	3	186	11	10.432	0.989
67	336.02	3	186	9	8.382	0.990
68	466.60	3	186	12	11.034	0.986
69	499.91	4	248	13	11.069	0.970
70	638.66	5	310	18	15.095	0.975
71	668.03	7	434	18	13.491	0.976
72	941.71	4	248	16	11.685	0.958
73	1350.56	4	248	21	16.494	0.969
74	1698.84	4	248	13	10.998	0.969
75	CF	10	614	20	13.355	0.937
76	CF	5	310	14	12.219	0.975
77	CF	4	248	13	11.419	0.969
78	CF	4	248	16	12.676	0.962

Table A2 Anuran species recorded across 78 surveyed sites at the Balbina Hydroelectric Reservoir landscape, including 74 forest islands and four continuous forest sites. ‘Sites’ corresponds to the number of sites at which species were captured. Species reproductive modes were assigned according to Haddad & Prado (2005). Taxonomy follows Frost et al. (2019).

Family	Species	Sites	Reproductive mode
Craugastoridae	<i>Pristimantis fenestratus</i>	10	23
	<i>Pristimantis ockendeni</i>	18	23
	<i>Pristimantis zimmermanae</i>	46	23
Bufonidae	<i>Atelopus hoogmoedi</i>	15	1
	<i>Rhaebo guttatus</i>	1	1
	<i>Rhinella marina</i>	2	1
	<i>Rhinella merianae</i>	2	1
Ceratophryidae	<i>Ceratophrys cornuta</i>	6	1
Aromobatidae	<i>Anomaloglossus stepheni</i>	28	21
Dendrobatidae	<i>Ameerega hahneli</i>	12	20
	<i>Ameerega trivittata</i>	54	20
Hylidae	<i>Boana boans</i>	34	4
	<i>Boana calcarata</i>	19	1
	<i>Boana cinerascens</i>	24	1
	<i>Boana lanciformis</i>	5	1
	<i>Boana aff. fasciata</i>	3	1
	<i>Dendropsophus brevifrons</i>	39	24
	<i>Dendropsophus minusculus</i>	45	1

Family	Species	Sites	Reproductive mode
	<i>Dendropsophus parviceps</i>	8	1
	<i>Osteocephalus buckleyi</i>	1	2
	<i>Osteocephalus oophagus</i>	12	6
	<i>Osteocephalus taurinus</i>	6	1
	<i>Trachycephalus coriaceus</i>	4	1
	<i>Trachycephalus resinifictrix</i>	2	26
	<i>Scinax garbei</i>	1	1
	<i>Scinax ruber</i>	4	1
Leptodactylidae	<i>Adenomera andreae</i>	35	32
	<i>Adenomera hylaedactyla</i>	12	32
	<i>Leptodactylus knudseni</i>	7	13
	<i>Leptodactylus longirostris</i>	1	13
	<i>Leptodactylus pentadactylus</i>	14	13
	<i>Leptodactylus stenodema</i>	5	13
Microhylidae	<i>Chiasmocleis shudikarensis</i>	9	1
	<i>Elachistocleis bicolor</i>	21	1
	<i>Synapturanus mirandaribeiroi</i>	8	23
Phyllomedusidae	<i>Phyllomedusa tarsius</i>	1	24
	<i>Phyllomedusa vaillantii</i>	2	24

References

- Frost, D. R. 2019. Amphibian Species of the World: An Online Reference. Version 6.0 (27 February 2019). - Am. Museum Nat. Hist. New York, USA. URL
<<http://research.amnh.org/herpetology/amphibia/index.html>>
- Haddad, C. F. B. and Prado, C. P. D. A. 2005. Reproductive modes in frogs and their unexpected diversity in the Atlantic Forest of Brazil. - *Bioscience* 55: 207.

Table A3 Description of the 25 datasets collated in the literature review (see Fig. 5), including data source, analytical approach employed, and area effect on observed species richness. The column ‘Reanalysis’ indicates the results of island species area-relationships (semi-log model), and the column ‘Matching results’ indicates whether the results from data sources and our reanalysis were convergent.

Data source	Analytical approach	Area effect	Reanalysis	Matching results
Alcala et al. 2004 ^a	Simple linear regression	positive	positive	yes
Almeida-Gomes and Rocha 2014	Model selection	low support	non-significant	yes
Bell and Donnelly 2006 ^b	Correlation test	positive	positive	yes
Bickford et al. 2010	Model selection	low support	positive	no
Bittencourt-Silva and Silva 2014	Path analysis	positive	positive	yes
Cabrera-Guzman and Reynoso 2012	Rarefaction curves	positive	positive	yes
Ferrante et al. 2017 ^c	Multiple linear regression	non-significant	positive	no
Hager et al. 1998 (Georgian Bay islands)	Simple linear regression	positive	positive	yes
Hager et al. 1998 (Lake Erie islands)	Simple linear regression	positive	positive	yes
Hager et al. 1998 (St. Lawrence islands)	Simple linear regression	positive	positive	yes
Herrera 2011	Multiple linear regression	non-significant	positive	no
Hillers et al. 2008	Correlation test	non-significant	non-significant	yes
Krystufek and Kletecki 2007	Simple linear regression	positive	positive	yes
Li et al. 2018	Simple linear regression	non-significant	non-significant	yes
Lima et al. 2015	Simple linear regression	positive	positive	yes
Lion et al. 2014	Simple linear regression	non-significant	non-significant	yes
Pineda and Halffter 2004 ^d	Multiple linear regression	positive	non-significant	no
Rakotondravony 2007	Correlation test	positive	positive	yes
Russildi et al. 2016 ^e	Model averaging	low weight	non-significant	yes
Vallan 2000	Correlation test	positive	positive	yes
Wang et al. 2018	Simple linear regression	positive	positive	yes
Watling et al. 2009	—	—	non-significant	—
Williams and Pearson 1997	Multiple linear regression	positive	positive	yes
Yiming et al. 1998	Simple linear regression	positive	positive	yes
Zimmerman and Bierregaard 1986	Simple linear regression	positive	positive	yes

^a The result of ‘Area effect’ included anurans, lizards and snakes as well as three types of habitat patches (limestone and non-limestone forest fragments, and limestone plantation forest), whereas the result of ‘Reanalysis’ included anurans and limestone forest fragment.

^b The result of ‘Area effect’ included anurans and lizards, and excluded the largest fragment (La Selva Biological Station), whereas the result of ‘Reanalysis’ included anurans and the largest fragment. To disentangle the number of species of anurans from lizards, we consulted Bell (2005).

^c The result of ‘Area effect’ included 16 habitat patches, whereas the result of ‘Reanalysis’ included 22 habitat patches. We consulted Faria (2017) to extract patch size and observed number of anuran species.

^d The result of ‘Area effect’ included three types of habitat patches (tropical montane cloud forest, shaded coffee, and pasture), whereas the result of ‘Reanalysis’ included only tropical montane cloud forest.

^e The result of ‘Area effect’ included anurans and salamanders, whereas the result of ‘Reanalysis’ included only anurans. To disentangle the number of species of anurans from salamanders, we consulted the first author.

Appendix A1: Data sources

- Alcala, E. L. et al. 2004. Amphibians and reptiles in tropical rainforest fragments on Negros Island, the Philippines. - *Environ. Conserv.* 31: 254–261.
- Almeida-Gomes, M. and Rocha, C. F. D. 2014. Landscape connectivity may explain anuran species distribution in an Atlantic forest fragmented area. - *Landsc. Ecol.* 29: 29–40.
- Almeida-Gomes, M. et al. 2016. Patch size matters for amphibians in tropical fragmented landscapes. - *Biol. Conserv.* 195: 89–96.
- Bell, K. E. 2005. Conservation ecology of amphibians and reptiles in Sarapiquí, Costa Rica: forest fragmentation and long-term population change. MSc Thesis. Florida International University, USA.
- Bell, K. E. and Donnelly, M. A. 2006. Influence of forest fragmentation on community structure of frogs and lizards in northeastern Costa Rica. - *Conserv. Biol.* 20: 1750–1760.
- Bickford, D. et al. 2010. Forest fragment and breeding habitat characteristics explain frog diversity and abundance in Singapore. - *Biotropica* 42: 119–125.
- Bittencourt-Silva, G. B. and Silva, H. R. 2014. Effects of fragmentation and sea-level changes upon frog communities of land-bridge islands off the southeastern coast of Brazil. - *PLoS One* 9: e103522.
- Cabrera-Guzmán, E. and Reynoso, V. H. 2012. Amphibian and reptile communities of rainforest fragments: minimum patch size to support high richness and abundance. - *Biodivers. Conserv.* 21: 3243–3265.

- Faria, L. F. de 2017. A síndrome das florestas vazias e a importância dos pequenos fragmentos para a conservação dos anfíbios. MSc Thesis. Instituto Nacional de Pesquisas da Amazônia, Brazil.
- Ferrante, L. et al. 2017. The matrix effect: how agricultural matrices shape forest fragment structure and amphibian composition. - *J. Biogeogr.*: 1–12.
- Hager, H. A. 1998. Area-sensitivity of reptiles and amphibians: Are there indicator species for habitat fragmentation? - *Écoscience* 5: 139–147.
- Herrera, J. B. 2011. Efeitos da heterogeneidade do ambiente, área e variáveis ambientais sobre anfíbios anuros em paisagem fragmentada de Floresta Atlântica. MSc Thesis. Universidade Federal da Bahia, Brazil.
- Hillers, A. et al. 2008. Effects of forest fragmentation and habitat degradation on west African leaf-litter frogs. - *Conserv. Biol.* 22: 762–772.
- Kryštufek, B. and Kletečki, E. 2007. Biogeography of small terrestrial vertebrates on the Adriatic landbridge islands. - *Folia Zool.* 56: 225–234.
- Li, B. et al. 2018. Influence of breeding habitat characteristics and landscape heterogeneity on anuran species richness and abundance in urban parks of Shanghai, China. - *Urban For. Urban Green.* 32: 56–63.
- Lima, J. R. et al. 2015. Amphibians on Amazonian land-bridge islands are affected more by area than isolation. - *Biotropica* 47: 369–376.
- Lion, M. B. et al. 2014. Split distance: a key landscape metric shaping amphibian populations and communities in forest fragments. - *Divers. Distrib.* 20: 1245–1257.

- Palmeirim, A. F. et al. 2017. Herpetofaunal responses to anthropogenic forest habitat modification across the neotropics: insights from partitioning β -diversity. - *Biodivers. Conserv.* 26: 2877–2891.
- Pineda, E. and Halffter, G. 2004. Species diversity and habitat fragmentation: frogs in a tropical montane landscape in Mexico. - *Biol. Conserv.* 117: 499–508.
- Rakotondravony, H. A. 2007. Conséquences de la variation des superficies forestières sur les communautés de reptiles et d'amphibiens dans la région Loky-Manambato, extrême nord-est de Madagascar. - *Rev. d'Écologie (La Terre La Vie)* 62: 209–227.
- Russildi, G. et al. 2016. Species- and community-level responses to habitat spatial changes in fragmented rainforests: assessing compensatory dynamics in amphibians and reptiles. - *Biodivers. Conserv.* 25: 375–392.
- Vallan, D. 2000. Influence of forest fragmentation on amphibian diversity in the nature reserve of Ambohitantely, highland Madagascar. - *Biol. Conserv.* 96: 31–43.
- Wang, Y. et al. 2018. The small-island effect in amphibian assemblages on subtropical land-bridge islands of an inundated lake. - *Curr. Zool.* 64: 303–309.
- Watling, J. I. et al. 2009. Nested species subsets of amphibians and reptiles on Neotropical forest islands. - *Anim. Conserv.* 12: 467–476.
- Williams, S. E. and Pearson, R. G. 1997. Historical rainforest contractions, localized extinctions and patterns of vertebrate endemism in the rainforests of Australia's wet tropics. - *Proc. R. Soc. B Biol. Sci.* 264: 709–716.
- Yiming, L. et al. 1998. Nested distribution of amphibians in the Zhoushan archipelago, China: can selective extinction cause nested subsets of species? - *Oecologia* 113: 557–564.

Zimmerman, B. L. and Bierregaard, R. O. 1986. Relevance of the equilibrium Theory of Island Biogeography and species-area relations to conservation with a case from Amazonia. - *J. Biogeogr.* 13: 133.

Supplementary material

Sampling design may obscure species-area relationships in landscape-scale field studies

```
# Clear workspace and disable scientific notation  
remove(list = ls()); options(scipen = 999)
```

Packages

```
# Load required packages  
library(vegan)  
library(dplyr)  
library(ggplot2)  
library(gridExtra)  
library(iNEXT)
```

Dataset

Data are available from the KNB repository and should be cited as:

Anderson Saldanha Bueno. 2019. Balbina Frog Data, 2015. Knowledge Network for Biocomplexity.
urn:uuid:fb6c7193-eca5-41ba-89dd-146c31c9dbe0.

Import data

```
# Fieldwork data derived from autonomous recordings units  
rawdata = read.csv("https://ndownloader.figshare.com/files/15158558")  
# Remove "Malfunctioning" recordings  
rawdata = subset(rawdata, rawdata$species != "Malfunctioning")  
# Remove the level "Malfunctioning" from "rawdata$species"  
rawdata$species = droplevels(rawdata$species)  
# Add a column indicating that each record corresponds to one detection  
rawdata$occurrence = 1  
  
# Site area (hectares)  
area = read.csv("https://ndownloader.figshare.com/files/15158564")  
rownames(area) = area$site  
area = area[-1]  
  
# Reproductive modes  
traits = read.csv("https://ndownloader.figshare.com/files/15158561")  
# Add a column with the reproductive of each species  
rawdata$rep.mod = traits$reproductive_mode[match(rawdata$species, traits$species)]  
  
# Site-by-reproductive mode matrix  
rep.mod = tapply(rawdata$occurrence, list(rawdata$site, rawdata$rep.mod), sum)  
# Species non detected in a given site are real zeros  
rep.mod[is.na(rep.mod)] = 0
```

```
# Coordinates of the surveyed plots (n = 151)
coordinates = read.csv("https://ndownloader.figshare.com/files/15158555", row.names = 1)
```

Figure S1

The relationship between island size and Julian day shows that anuran surveys covered all island size categories throughout the sampling period (11 Jul to 4 Dec 2015) virtually randomly.

```
# Data to draw the graph
df.74 = data.frame(
  sites = c("10_626", "10_709", "11_497", "12_28", "12_9", "14_7",
            "15_188", "17_697", "185_358", "2_258", "2_333", "2_771",
            "2_794", "2_87", "235_234", "3_207", "3_311", "3_592",
            "33_793", "34_526-A", "34_526-B", "37_028", "37_7", "37_9",
            "4_746", "43_792", "44_174", "44_21", "49_62", "5_708",
            "54_544-A", "54_544-B", "54_8", "7_335", "8_042", "8_672",
            "Abandonada_left", "Abandonada_right", "Abusado", "Adeus",
            "Aline", "Andre", "Arrepiado", "Bacaba", "Beco_do_Catitu",
            "Cafundo", "Cipoal", "Coata", "Formiga", "Furo_de_Santa_Luzia",
            "Fuzaca", "Garrafa", "Gaviao_real", "Jabuti", "Jiquitaia",
            "Joaninha", "Louzivaldo", "Martelo", "Mascote", "Moita",
            "Neto", "Palhal", "Panema", "Pe_Torto", "Piquia", "Pontal",
            "Porto_Seguro", "Relogio", "Sapupara", "Toquinho", "Torem",
            "Tristeza", "Tucumari", "Xibe"),
  area = c(8.78, 5.43, 3.53, 8.42, 11.96, 13.22, 1.81, 17.57, 171.73, 0.72,
           0.63, 0.45, 0.91, 1.45, 230.7, 1.99, 1.36, 2.08, 29.62, 5.61,
           15.67, 32.78, 39.67, 35.6, 2.26, 38.94, 39.12, 35.87, 39.94, 3.53,
           17.66, 22.01, 52.71, 4.17, 6.88, 8.15, 8.15, 0.45, 13.31, 50.08,
           2.08, 2.08, 7.43, 53.35, 638.66, 2.63, 217.63, 16.94, 1.54, 198.52,
           941.71, 9.42, 1698.84, 232.49, 6.79, 0.63, 1.27, 466.6, 668.03,
           91.3, 32.87, 21.37, 3.08, 4.98, 13.04, 108.76, 1350.56, 70.55, 77.8,
           0.63, 3.62, 499.91, 336.02, 0.91),
  julian = c(219, 229, 229, 212, 199, 199, 192, 283, 300, 212, 212, 212,
            212, 219, 277, 229, 212, 192, 193, 212, 212, 219, 199, 199,
            192, 229, 229, 277, 277, 192, 219, 219, 199, 212, 193, 277,
            277, 277, 257, 322, 192, 284, 192, 199, 256, 331, 318, 291,
            212, 229, 299, 290, 331, 298, 322, 193, 316, 328, 291, 284,
            219, 219, 318, 219, 291, 322, 316, 257, 257, 318, 322, 328,
            335, 283))

# Draw the graph of the relationship between Julian day and island size
graph.julian =
  ggplot(aes(x = julian, y = area),
        data = df.74[order(as.numeric(df.74$julian), df.74$area), ]) +

  labs(x = "Julian day",
       y = "Island size (ha)") +

  scale_x_continuous(limits = c(192, 335), breaks = seq(200, 325, 25)) +
  scale_y_log10(limits = c(0.45, 1698.84), breaks = c(1, 10, 100, 1000),
               labels = c("1", "10", "100", "1,000")) +
  annotation_logticks(base = 10, sides = "l") +
```

```

geom_hline(yintercept = c(4, 20, 100, 400), colour = "grey", linetype = "dashed") +
geom_point(shape = 21, size = 4, colour = "black", fill = "#999999") +

theme_classic(base_size = 20) +
theme(axis.title = element_text(colour = "black", face = "bold"),
      axis.text = element_text(colour = "black"),
      axis.ticks = element_line(size = 0.5, colour = "black"),
      axis.line = element_line(size = 0.5))

ggsave(graph.julian, filename = "figs1.pdf", width = 20, height = 20, units = "cm")

```

Figure S2

```

# Data to draw the graph
rm.area = data.frame(rm = specnumber(rep.mod), area = area$area)

# Relationship between reproductive modes and island size
# Include only forest islands
summary(lm(rm ~ log10(area), data = rm.area[-c(47:50), ]))

##
## Call:
## lm(formula = rm ~ log10(area), data = rm.area[-c(47:50), ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.8047 -1.1534 -0.0975  1.0647  4.8068
##
## Coefficients:
##              Estimate Std. Error t value      Pr(>|t|)
## (Intercept)   2.2618     0.3216   7.032 0.0000000009647 ***
## log10(area)   1.6765     0.2163   7.751 0.0000000000447 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.699 on 72 degrees of freedom
## Multiple R-squared:  0.4549, Adjusted R-squared:  0.4473
## F-statistic: 60.08 on 1 and 72 DF,  p-value: 0.0000000000447

# Draw the graph of the relationship between reproductive modes and island size
graph.rm.isl =
  ggplot(aes(x = area, y = rm),
        # Include only forest islands
        data = rm.area[order(rm.area$area, rm.area$rm), ][-c(75:78), ]) +

  labs(x = "Island size (ha)",
       y = "Number of reproductive modes") +

  scale_x_log10(breaks = c(1, 10, 100, 1000),
               labels = c("1", "10", "100", "1,000")) +
  scale_y_continuous(limits = c(0, 11),
                    breaks = seq(0, 10, 2)) +
  annotation_logticks(base = 10, sides = "b") +

```

```

geom_smooth(method = "lm", colour = "black") +
geom_point(shape = 21, size = 4, colour = "black", fill = "#999999") +

theme_classic(base_size = 20) +
theme(axis.title = element_text(colour = "black", face = "bold"),
      axis.text = element_text(colour = "black"),
      axis.ticks = element_line(size = 0.5, colour = "black"),
      axis.line = element_line(size = 0.5)) +

annotate("text", x = min(rm.area$area), y = max(rm.area$rm),
          hjust = 0, vjust = 1, fontface = "bold", size = 6,
          parse = T, label = as.character(expression(r2*"[adj]" = 0.45))) +
annotate("text", x = min(rm.area$area), y = max(rm.area$rm),
          hjust = 0, vjust = 3, fontface = "bold", size = 6,
          parse = T, label = "p < 0.001")

# Draw the graph for continuous forest sites
graph.rm.cf =
  ggplot() +

  scale_x_discrete(labels = c("16988.4" = "CF")) +
  scale_y_continuous(limits = c(min(rm.area$rm), max(rm.area$rm))) +

  geom_point(shape = 21, size = 4, colour = "black", fill = "black",
            aes(x = as.factor(area), y = rm),
            # Include only continuous forest sites
            data = subset(rm.area, rm.area$area > 2000)) +

  theme_classic(base_size = 20) +
  theme(axis.title.x = element_text(colour = "white", face = "bold"),
        axis.text.x = element_text(colour = "black"),
        axis.ticks.x = element_line(size = 0.5, colour = "black"),
        axis.line.x = element_line(size = 0.5),
        axis.title.y = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank(),
        axis.line.y = element_blank())

# Combine and save the graphs
ggsave(grid.arrange(graph.rm.isl, graph.rm.cf, ncol = 2, widths = c(5, 1)),
       file = "figs2.pdf", width = 20, height = 20, units = "cm")

# Display the graphs
#grid.arrange(graph.rm.isl, graph.rm.cf, ncol = 2, widths = c(5, 1))

```

Frog data

```

# Create site-by-species matrix for 78 sites
frogs = as.data.frame(tapply(rawdata$occurrence,
                             list(rawdata$site, rawdata$species), sum))
# Species not detected in a site are real zeros
frogs[is.na(frogs)] = 0

```



```

# Transpose the data frames stored in the list objec
# This step creates species-by-sampling-unit matrices (i.e. species-by-recording matrices)
inext.site = lapply(inext.site, function(z) {t(z)})

# Remove sites where no species was recorded
inext.site = inext.site[-which(lapply(inext.site, sum) == 0)] # 1 site removed (#74)

```

Calculation of the rarefied number of species

```

# Rarefied number of species per site standardised to 62 recordings
richness.site = matrix(nrow = 77, ncol = 5)
for (i in 1:77){
  # Remove site "Toquinho" (#74) where no species was recorded
  richness.site[i, 1] = rownames(frogs)[-74][i]
  richness.site[i, 2:5] = as.numeric(iNEXT(inext.site[[i]], q = 0,
                                         datatype = "incidence_raw", # iNEXT::iNEXT()
                                         endpoint = effort.mode, knot = effort.mode,
                                         nboot = 1000)$iNextEst[effort.mode, 4:7])
}

# Add site "Toquinho" (#74)
richness.site = rbind(richness.site, cbind("Toquinho", "0", "0", "0", "NA"))

# Order table by site names
richness.site = rbind(richness.site[1:73, ],
                      cbind("Toquinho", "0", "0", "0", "NA"),
                      richness.site[74:77, ])

# Check if the site "Toquinho" was placed back in the right row (#74)
richness.site[74, 1] == "Toquinho"

# Finish the data frame
# Name rows according to site names
rownames(richness.site) = richness.site[, 1]
# Remove the column "site"
richness.site = richness.site[, -1]
# Rename columns
colnames(richness.site) = c("richness", "lowerCI", "upperCI", "coverage")
# Convert the matrix "richness.site" into a data frame
richness.site = as.data.frame(richness.site)

# Convert factor data into numeric data
richness.site$richness = as.numeric(as.character(richness.site$richness))
# Convert factor data into numeric data
richness.site$lowerCI = as.numeric(as.character(richness.site$lowerCI))
# Convert factor data into numeric data
richness.site$upperCI = as.numeric(as.character(richness.site$upperCI))
# Convert factor data into numeric data
richness.site$coverage = as.numeric(as.character(richness.site$coverage))

```

Correlation between observed number of species and sampling effort

```
# Pearson's product-moment correlation between
# observed number of species and sampling effort
cor.test(specnumber(frogs), effort, method = "pearson")

##
## Pearson's product-moment correlation
##
## data: specnumber(frogs) and effort
## t = 12.452, df = 76, p-value < 0.00000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7296882 0.8811067
## sample estimates:
##      cor
## 0.819199
```

Figure 2a

```
# Data to draw the graph
obs.effort = data.frame(obs = specnumber(frogs), effort = effort)

# Draw the graph of the correlation between
# observed number of species and sampling effort
graph.obs.effort =
  ggplot(aes(x = effort, y = obs),
         data = obs.effort[order(obs.effort$effort, obs.effort$obs), ]) +

  labs(x = "Sampling effort",
       y = "Observed number of species") +

  scale_x_continuous(breaks = c(62, 124, 186, 248, 310, 434, 614)) +

  geom_point(shape = 21, size = 4, colour = "black", fill = "#999999") +

  theme_bw(base_size = 20) +
  theme(panel.grid = element_blank(),
        panel.border = element_rect(colour = "black"),
        axis.title = element_text(colour = "black", face = "bold"),
        axis.text = element_text(colour = "black"),
        axis.ticks = element_line(colour = "black", size = 0.5)) +

  annotate("text", x = min(obs.effort$effort), y = max(obs.effort$obs),
         hjust = 0, vjust = 1, fontface = "bold", size = 8,
         parse = T, label = as.character(expression(italic(r)*" = 0.82"))) +
  annotate("text", x = min(obs.effort$effort), y = max(obs.effort$obs),
         hjust = 0, vjust = 2.5, fontface = "bold", size = 8,
         parse = T, label = "p < 0.001") +
  annotate("text", x = Inf, y = -Inf,
         hjust = 1.3, vjust = -1, fontface = "bold", size = 10, label = "(a)")
```

```
#graph.obs.effort
```

Correlation between observed and rarefied number of species

```
# Pearson's product-moment correlation between observed and rarefied number of species
cor.test(specnumber(frogs), richness.site$richness, method = "pearson")

##
## Pearson's product-moment correlation
##
## data: specnumber(frogs) and richness.site$richness
## t = 47.348, df = 76, p-value < 0.000000000000000022
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.9741280 0.9894553
## sample estimates:
##      cor
## 0.9834688
```

Figure 2b

```
# Data to draw the graph
# The rarefied number of speices for the site "Toquinho" was assigned to 0 (zero)
obs.raref = data.frame(obs = specnumber(frogs), raref = richness.site$richness)

# Draw the graph of the correlation between observed and rarefied number of species
graph.obs.raref =
  ggplot(aes(x = raref, y = obs),
    data = obs.raref[order(obs.raref$raref, obs.raref$obs), ]) +

  labs(x = "Rarefied number of species",
    y = "Observed number of species") +

  geom_point(shape = 21, size = 4, colour = "black", fill = "#999999") +
  geom_point(shape = 21, size = 4, colour = "black", fill = "white",
    aes(x = 0, y = 0)) +

  theme_bw(base_size = 20) +
  theme(panel.grid = element_blank(),
    panel.border = element_rect(colour = "black"),
    axis.title = element_text(colour = "black", face = "bold"),
    axis.text = element_text(colour = "black"),
    axis.ticks = element_line(colour = "black", size = 0.5)) +

  annotate("text", x = min(obs.raref$raref), y = max(obs.raref$obs),
    hjust = 0, vjust = 1, fontface = "bold", size = 8,
    parse = T, label = as.character(expression(italic(r)*" = 0.98"))) +
  annotate("text", x = min(obs.raref$raref), y = max(obs.raref$obs),
    hjust = 0, vjust = 2.5, fontface = "bold", size = 8,
    parse = T, label = "p < 0.001") +
  annotate("text", x = Inf, y = -Inf,
```



```
hjust = 1.3, vjust = -1, fontface = "bold", size = 10, label = "(b)")
```

```
#graph.obs.raref
```

Sample coverage

Figure 2c

```
# Data to draw the graph
coverage = data.frame(site = NA, coverage = richness.site[4])
# Order the data frame according to site size
coverage = coverage[order(area$area), ]
# Add site number
coverage$site = 1:nrow(coverage)

# Draw the graph showing sample coverage per site
graph.coverage =
  ggplot(aes(x = site, y = coverage),
        data = coverage) +

  labs(x = "Site code",
       y = "Sample coverage") + # sample coverage in 62 1-min recordings

  scale_x_continuous(breaks = c(1, 10, 20, 30, 40, 50, 60, 70, 78)) +
  scale_y_continuous(limits = c(0.5, 1)) +

  geom_point(shape = 21, size = 4, colour = "black", fill = "#999999") +
  geom_point(shape = 21, size = 4, colour = "black", fill = "white",
            aes(x = 5, y = 0.5)) +

  theme_bw(base_size = 20) +
  theme(panel.grid = element_blank(),
        panel.border = element_rect(colour = "black"),
        axis.title = element_text(colour = "black", face = "bold"),
        axis.text = element_text(colour = "black"),
        axis.ticks = element_line(colour = "black", size = 0.5)) +

  annotate("text", x = Inf, y = -Inf,
        hjust = 1.3, vjust = -1, fontface = "bold", size = 10, label = "(c)")

#graph.coverage
```

Combine graphs

Figure 2

```
# Combine and save the graphs
ggsave(grid.arrange(graph.effort, graph.obs.raref, graph.coverage, ncol = 3),
       file = "fig2.pdf", width = 17*3, height = 17, units = "cm")
```

Sample coverage was above 90% for 75 out of 78 sites, indicating that our sampling effort was overall satisfactory.

Table S1

Site attributes

```
tables1 = data.frame(Island.size = area$area, # forest area in hectares
  Recorders = area$recorders, # number of recording stations
  Effort = effort, # number of 1-min recordings
  S.observed = specnumber(frogs),
  S.rarefied = richness.site$richness,
  # lower bound of the 95% confidence intervals
  S.rarefied.lowerCI = richness.site$lowerCI,
  # upper bound of the 95% confidence intervals
  S.rarefied.upperCI = richness.site$upperCI,
  Coverage = richness.site$coverage)

# Order table by site area
tables1 = tables1[order(tables1$Island.size), ]
# Number surveyed sites (the numbers match those of Fig. 2c)
tables1$Site.code = 1:nrow(tables1)
# Move "site.code" to the first column
tables1 = tables1[, c(9, 1:8)]
rownames(tables1) = NULL

tables1$Island.size[tables1$Island.size == 16988.40] = "Continuous forest"

#tables1
```

Table S2

List of frog species recorded across all 151 plots in 78 surveyed sites and the number of sites occupied per species. Taxonomy follows Frost (2018).

- Because of taxonomic revision, *Boana fasciata* is no longer considered present in Brazil according to Frost (2018). However, the species present in Brazil (i.e. *Boana aff. fasciata*) has not yet been assigned to a new taxon. Therefore, we held the name *Boana fasciata*.
- *Leptodactylus knudseni* and *L. pentadactylus* are hardly distinguished through vocalisation by both hearing and inspecting sonograms. In the fieldwork data (available online at KNB repository), we assigned *Leptodactylus knudseni pentadactylus* whenever our best guess was *Leptodactylus knudseni*, and *Leptodactylus pentadactylus knudseni* whenever our best guess was *L. pentadactylus*. Thus, we acknowledge that one species maybe the other in some instances.

```
tables2 = names(frogs)

# Rename two species
tables2[tables2 == "Leptodactylus knudseni pentadactylus"] = "Leptodactylus knudseni"
tables2[tables2 == "Leptodactylus pentadactylus knudseni"] = "Leptodactylus pentadactylus"

# Split genus and specific epithet into separate columns
tables2 = data.frame(do.call(rbind, strsplit(as.character(tables2), " ", fixed = TRUE)))

# Species family
families = c("Leptodactylidae", "Leptodactylidae", "Dendrobatidae", "Dendrobatidae",
  "Aromobatidae", "Bufonidae", "Hylidae", "Hylidae", "Hylidae",
  "Hylidae", "Hylidae", "Ceratophryidae", "Microhylidae", "Hylidae",
  "Hylidae", "Hylidae", "Microhylidae", "Leptodactylidae", "Leptodactylidae",
```

```

      "Leptodactylidae", "Leptodactylidae", "Hylidae", "Hylidae", "Hylidae",
      "Phyllomedusidae", "Phyllomedusidae", "Craugastoridae", "Craugastoridae",
      "Craugastoridae", "Bufonidae", "Bufonidae", "Bufonidae", "Hylidae",
      "Hylidae", "Microhylidae", "Hylidae", "Hylidae")

# Taxonomic sequence
id = c(860, 867, 333, 337, 326, 153, 401, 407, 409, 436, 428, 251, 971,
      498, 526, 534, 980, 895, 901, 911, 923, 572, 578, 581, 1077, 1079,
      89, 99, 114, 193, 221, 223, 695, 717, 999, 601, 610)

# Species reproductive modes
spp.rep.mod = c(32, 32, 20, 20, 21, 1, 4, 1, 1, 1, 1, 1, 24, 1, 1, 1,
      13, 13, 13, 13, 2, 6, 1, 24, 24, 23, 23, 23, 1, 1, 1, 1, 23, 1, 26)

# Species list and number of sites where each species was recorded
tables2 = data.frame(id = id,
      Family = families,
      Species = as.character(paste(tables2$X1, tables2$X2, sep = " ")),
      Sites = cbind(colSums(decostand(frogs, method = "pa"))),
      Reproductive.mode = spp.rep.mod)

tables2 = tables2[order(tables2$id), ] # order table according to the taxonomic sequence
tables2 = tables2[-1] # remove column "id"
rownames(tables2) = 1:nrow(tables2) # number the rows sequentially

#tables2

```

Species-area relationships (ISAR)

```

# Data to create ISAR models and draw the graph
sar = data.frame(richness = richness.site$richness, area)
# Order table by site area followed by species richness
sar = sar[order(sar$area, sar$richness), ]

# Create a column with site size category
sar$class[sar$area < 4] = "very_small" # up to 4.00 ha = very small
sar$class[sar$area > 4 & sar$area <= 20 ] = "small" # 4.01 - 20.00 ha = small
sar$class[sar$area > 20 & sar$area <= 100 ] = "medium" # 20.01 - 100.00 ha = medium
sar$class[sar$area > 100 & sar$area <= 400 ] = "large" # 100.01 - 400.00 = large
sar$class[sar$area > 400 & sar$area < 2000 ] = "very_large" # > 400.01 = very large
sar$class[sar$area > 2000 ] = "continuous" # continuous forest sites

# Create a column with an unique number per site
sar$id = 1:nrow(sar)

```

ISAR – semi-log models

```

# ISAR models
sar.74 = lm(richness ~ log10(area), data = subset(sar, sar$area < 2000))
sar.67 = lm(richness ~ log10(area), data = subset(sar, sar$area <= 400))

```

```

sar.60 = lm(richness ~ log10(area), data = subset(sar, sar$area <= 100))
sar.43 = lm(richness ~ log10(area), data = subset(sar, sar$area <= 20))
sar.23 = lm(richness ~ log10(area), data = subset(sar, sar$area < 4))

```

```

# ISAR results

```

```

sar.results = rbind(
  c(sar.74$coefficients[1], sar.74$coefficients[2],
    confint(sar.74)[2], confint(sar.74)[4],
    summary(sar.74)$adj.r.squared, summary(sar.74)$coefficients[8]),
  c(sar.67$coefficients[1], sar.67$coefficients[2],
    confint(sar.67)[2], confint(sar.67)[4],
    summary(sar.67)$adj.r.squared, summary(sar.67)$coefficients[8]),
  c(sar.60$coefficients[1], sar.60$coefficients[2],
    confint(sar.60)[2], confint(sar.60)[4],
    summary(sar.60)$adj.r.squared, summary(sar.60)$coefficients[8]),
  c(sar.43$coefficients[1], sar.43$coefficients[2],
    confint(sar.43)[2], confint(sar.43)[4],
    summary(sar.43)$adj.r.squared, summary(sar.43)$coefficients[8]),
  c(sar.23$coefficients[1], sar.23$coefficients[2],
    confint(sar.23)[2], confint(sar.23)[4],
    summary(sar.23)$adj.r.squared, summary(sar.23)$coefficients[8]))

```

```

# Convert the matrix "sar.results" in to a data frame

```

```

sar.results = as.data.frame(sar.results)

```

```

# Name the columns

```

```

colnames(sar.results) = c("intercept", "slope", "lowerCI", "upperCI", "r2", "p-value")

```

```

sar.results

```

```

##   intercept      slope    lowerCI upperCI      r2
## 1  2.302099  2.8364991  2.17011698 3.502881 0.49305716
## 2  2.857095  2.0359273  1.23228702 2.839568 0.27151241
## 3  3.260687  1.3001936  0.31134217 2.289045 0.09129034
## 4  3.218833  1.6885078  0.09017261 3.286843 0.07797120
## 5  3.442906  0.2327528 -3.67518220 4.140688 -0.04685428
##
##                p-value
## 1 0.000000000001903107
## 2 0.000003684038432128
## 3 0.010859533769233076
## 4 0.038917176533921849
## 5 0.902604188833153742

```

```

# Draw the graph for forest islands

```

```

graph.sar =

```

```

  ggplot() +

```

```

    labs(x = "Island size (ha)",
         y = "Rarefied number of species") +

```

```

    scale_x_log10(limits = c(0.4, NA),
                 breaks = c(1, 10, 100, 1000),
                 labels = c("1", "10", "100", "1,000")) +

```

```

    scale_y_continuous(limits = c(min(sar$richness), max(sar$richness))) +
    annotation_logticks(base = 10, sides = "b") +

```

```

geom_vline(xintercept = 4, alpha = 0.2, linetype = "dashed") +
geom_vline(xintercept = 20, alpha = 0.2, linetype = "dashed") +
geom_vline(xintercept = 100, alpha = 0.2, linetype = "dashed") +
geom_vline(xintercept = 400, alpha = 0.2, linetype = "dashed") +

geom_smooth(size = 1, method = "lm", colour = "#4daf4a", se = FALSE,
            aes(x = area, y = richness),
            data = subset(sar, sar$area < 2000)) +
geom_smooth(size = 1, method = "lm", colour = "dodgerblue", se = FALSE,
            aes(x = area, y = richness),
            data = subset(sar, sar$area <= 400)) +
geom_smooth(size = 1, method = "lm", colour = "#984ea3", se = FALSE,
            aes(x = area, y = richness),
            data = subset(sar, sar$area <= 100)) +
geom_smooth(size = 1, method = "lm", colour = "#ff7f00", se = FALSE,
            aes(x = area, y = richness),
            data = subset(sar, sar$area <= 20)) +
geom_smooth(size = 1, method = "lm", colour = "#e41a1c", se = FALSE,
            aes(x = area, y = richness),
            data = subset(sar, sar$area < 4)) +

geom_point(shape = 21, size = 4, colour = "black", fill = "#e41a1c",
           aes(x = area, y = richness),
           data = subset(sar, sar$class == "very_small")) +
geom_point(shape = 21, size = 4, colour = "black", fill = "#ff7f00",
           aes(x = area, y = richness),
           data = subset(sar, sar$class == "small")) +
geom_point(shape = 21, size = 4, colour = "black", fill = "#984ea3",
           aes(x = area, y = richness),
           data = subset(sar, sar$class == "medium")) +
geom_point(shape = 21, size = 4, colour = "black", fill = "dodgerblue",
           aes(x = area, y = richness),
           data = subset(sar, sar$class == "large")) +
geom_point(shape = 21, size = 4, colour = "black", fill = "#4daf4a",
           aes(x = area, y = richness),
           data = subset(sar, sar$class == "very_large")) +

theme_classic(base_size = 20) +
theme(axis.title = element_text(colour = "black", face = "bold"),
      axis.text = element_text(colour = "black"),
      axis.ticks = element_line(size = 0.5, colour = "black"),
      axis.line = element_line(size = 0.5)) +

annotate("text", x = 0.4, y = max(sar$richness),
         hjust = 0, vjust = 0, fontface = "bold", size = 4.5,
         parse = T, label = as.character(expression(r2*"[adj]"))) +
annotate("text", x = 0.4, y = max(sar$richness) - 1,
         hjust = 0, vjust = 0, size = 4.5, colour = "#4daf4a",
         label = round(sar.results$r2[1], 2)) +
annotate("text", x = 0.4, y = max(sar$richness) - 2,
         hjust = 0, vjust = 0, size = 4.5, colour = "dodgerblue",
         label = round(sar.results$r2[2], 2)) +
annotate("text", x = 0.4, y = max(sar$richness) - 3,

```

```

      hjust = 0, vjust = 0, size = 4.5, colour = "#984ea3",
      label = round(sar.results$r2[3], 2)) +
annotate("text", x = 0.4, y = max(sar$richness) - 4,
      hjust = 0, vjust = 0, size = 4.5, colour = "#ff7f00",
      label = round(sar.results$r2[4], 2)) +
annotate("text", x = 0.4, y = max(sar$richness) - 5,
      hjust = 0, vjust = 0, size = 4.5, colour = "#e41a1c",
      label = round(sar.results$r2[5], 2)) +

annotate("text", x = min(sar$area) + 0.9, y = max(sar$richness),
      hjust = 0, vjust = 0, fontface = "bold", size = 4.5,
      parse = T, label = as.character(expression(slope~{*}*"[ ]))) +
annotate("text", x = min(sar$area) + 0.9, y = max(sar$richness) - 1,
      hjust = 0, vjust = 0, size = 4.5, colour = "#4daf4a",
      label = round(sar.results$slope[1], 2)) +
annotate("text", x = min(sar$area) + 0.9, y = max(sar$richness) - 2,
      hjust = 0, vjust = 0, size = 4.5, colour = "dodgerblue",
      label = round(sar.results$slope[2], 2)) +
annotate("text", x = min(sar$area) + 0.9, y = max(sar$richness) - 3,
      hjust = 0, vjust = 0, size = 4.5, colour = "#984ea3",
      label = paste(round(sar.results$slope[3], 2), "0", sep = "")) +
annotate("text", x = min(sar$area) + 0.9, y = max(sar$richness) - 4,
      hjust = 0, vjust = 0, size = 4.5, colour = "#ff7f00",
      label = round(sar.results$slope[4], 2)) +
annotate("text", x = min(sar$area) + 0.9, y = max(sar$richness) - 5,
      hjust = 0, vjust = 0, size = 4.5, colour = "#e41a1c",
      label = round(sar.results$slope[5], 2))

# Draw the graph for continuous forest sites
graph.cf =
  ggplot() +

  scale_x_discrete(labels = c("16988.4" = "CF")) +
  scale_y_continuous(limits = c(min(sar$richness), max(sar$richness))) +

  geom_point(shape = 21, size = 4, colour = "black", fill = "black",
    aes(x = as.factor(area), y = richness),
    data = subset(sar, sar$class == "continuous")) +

  theme_classic(base_size = 20) +
  theme(axis.title.x = element_text(colour = "white", face = "bold"),
    axis.text.x = element_text(colour = "black"),
    axis.ticks.x = element_line(size = 0.5, colour = "black"),
    axis.line.x = element_line(size = 0.5),
    axis.title.y = element_blank(),
    axis.text.y = element_blank(),
    axis.ticks.y = element_blank(),
    axis.line.y = element_blank())

```

Figure 3

```
# Combine and save the graphs
ggsave(grid.arrange(graph.sar, graph.cf, ncol = 2, widths = c(5, 1)),
        file = "fig3.pdf", width = 20, height = 20, units = "cm")

# Display the graphs
#grid.arrange(graph.sar, graph.cf, ncol = 2, widths = c(5, 1))
```

Stratified random selection (full range)

```
# Exclude continuous forest sites as they will not be used to produce species-area curves
sar.isl = sar[-c(75:78), ]
```

```
#####
##### 25 #####
#####
```

```
# Generate and save 100 random numbers
# sample(1:Machine$integer.max, 100) # This command generated the following values
```

```
seed.full.25 = c(983130207, 383464058, 2017550925, 324031272, 1355246268,
1466882829, 135869051, 2003688700, 670295328, 917206259, 1510315544,
464385672, 1143533469, 1377184171, 272370526, 68508786, 865409067,
572626257, 216861211, 1525481127, 1817804462, 1005149655, 844395643,
115244292, 1577066482, 1362947697, 142092602, 299929719, 1392054026,
1763778950, 1370019851, 1185576929, 606262959, 639263150, 1622779910,
98441430, 1390362571, 670120600, 949563928, 186954742, 1574383261,
986984798, 667612850, 1258974800, 512876766, 1427057290, 1651618970,
219655950, 820545953, 387449755, 478465250, 853418124, 850681705, 273795464,
364350513, 490331164, 1124787011, 21106045, 721221771, 2100270184,
1759723902, 918280273, 886231669, 397522777, 696864331, 1907841509,
696786153, 2097466496, 833093004, 1191889090, 2048925380, 642337978,
2067077169, 971378045, 872419266, 424610369, 2001795318, 214703074,
77460136, 1199542182, 25340488, 709315005, 508980850, 364845407, 674333206,
748936353, 633356058, 120463230, 1498777378, 220758228, 883591132,
796722650, 1119550301, 1328061235, 475824336, 687856599, 1846199925,
233731714, 145666729, 470558232)
```

```
# Select 25 islands 100 times
sample.full.25 = matrix(nrow = 25, ncol = 100)
```

```
for (i in 1:100) {
  set.seed(seed.full.25[i])
  sample.full.25[, i] = data.frame(sar.isl %>% group_by(class) %>% sample_n(size = 5))$id
}
```

```
sample.full.25 = as.data.frame(sample.full.25) # convert the matrix into a data frame
```

```
# ISAR for each subset of 25 sites (n = 100)
slope.full.25 = r2.full.25 = p.full.25 = c()
```

```

for (i in 1:100) {
  slope.full.25[i] = lm(richness ~ log10(area),
    # Regression slope
    data = sar.isl[sample.full.25[, i], ])$coefficients[2]
  r2.full.25[i] = summary(lm(richness ~ log10(area),
    # Model fit (r2)
    data = sar.isl[sample.full.25[, i], ]))$adj.r.squared
  p.full.25[i] = summary(lm(richness ~ log10(area),
    # p-value
    data = sar.isl[sample.full.25[, i], ]))$coefficients[8]
}

# Store results in a data frame
sar.full.25 = data.frame(slope = slope.full.25,
  r2 = r2.full.25, p = p.full.25, n.isl = 25)

#####
##### 20 #####
#####

# Generate and save 100 random numbers
# sample(1:Machine$integer.max, 100) # This command generated the following values
seed.full.20 = c(1528803611, 1565203937, 1378459504, 1974162804, 1485726889,
  1722217419, 1083945832, 309442241, 2135586820, 904398258, 630425221,
  1571984846, 1881240840, 93071931, 1709954502, 1003892181, 242471852,
  374885596, 1471200150, 694386905, 1523256789, 1959936107, 102932741,
  1517589256, 2065922824, 1584981789, 815772258, 1290188023, 1479248051,
  1003253307, 1738192827, 1249221035, 592567972, 4847476, 241637366,
  856766312, 1250916602, 2097921953, 371046614, 1778211159, 732954012,
  2025770722, 1314949296, 357269182, 1399243432, 1255779923, 4390814,
  1154419746, 291235419, 1054835586, 980654265, 1066732189, 994545268,
  1009814846, 1894401743, 244382523, 766067747, 253772578, 1454223960,
  1007622885, 999021627, 1380324380, 1774930409, 214573555, 1492109997,
  1015109461, 747951269, 467889625, 785792844, 1613356238, 1152406342,
  780090909, 1582859756, 243340419, 955561798, 942675456, 1967585248,
  2116004579, 710299037, 1773143935, 360606364, 622966034, 1925227232,
  1285169228, 176215505, 572024367, 1009649767, 1424840105, 2053775408,
  690646673, 528845198, 398899374, 1986016719, 1228361023, 83457039,
  984747662, 578408909, 700617890, 1037532108, 776063694)

# Select 20 islands 100 times
sample.full.20 = matrix(nrow = 20, ncol = 100)

for (i in 1:100) {
  set.seed(seed.full.20[i])
  sample.full.20[, i] = data.frame(sar.isl %>% group_by(class) %>% sample_n(size = 4))$id
}

sample.full.20 = as.data.frame(sample.full.20) # convert the matrix into a data frame

# ISAR for each subset of 20 sites (n = 100)
slope.full.20 = r2.full.20 = p.full.20 = c()

```



```

for (i in 1:100) {
  slope.full.20[i] = lm(richness ~ log10(area),
    # Regression slope
    data = sar.isl[sample.full.20[, i], ])$coefficients[2]
  r2.full.20[i] = summary(lm(richness ~ log10(area),
    # Model fit (r2)
    data = sar.isl[sample.full.20[, i], ]))$adj.r.squared
  p.full.20[i] = summary(lm(richness ~ log10(area),
    # p-value
    data = sar.isl[sample.full.20[, i], ]))$coefficients[8]
}

# Store results in a data frame
sar.full.20 = data.frame(slope = slope.full.20,
  r2 = r2.full.20, p = p.full.20, n.isl = 20)

#####
##### 15 #####
#####

# Generate and save 100 random numbers
# sample(1:Machine$integer.max, 100) # This command generated the following values
seed.full.15 = c(1032401521, 673699241, 1139750126, 142050191, 368795250,
  704181501, 663119545, 1012593324, 1282130941, 510145834, 1521414786,
  542345656, 99629308, 760242161, 156335171, 1443509149, 602490744, 1365239361,
  279730440, 182891111, 1948787162, 1446143351, 582764462, 211891494,
  264602481, 720226263, 1152228529, 535122942, 501508315, 1423455275,
  1938960957, 915067793, 1363188482, 516054131, 1718048524, 1558759182,
  1074540444, 1961880847, 197450330, 1958961253, 1987503820, 1470673232,
  872996307, 998258619, 24518139, 2035293581, 341314754, 659447199, 1505443998,
  286454269, 804362813, 945106698, 2003549731, 1365988459, 1213480577,
  1326706029, 550594977, 699060407, 1998830047, 56226447, 1142154175,
  668901331, 2063475684, 643829539, 945402271, 2124625018, 1690827344,
  948726833, 456447170, 866335614, 1032176747, 1794769238, 1958945420,
  2079412266, 1827338298, 645632015, 905251202, 1532502124, 1411979196,
  1184782471, 725942691, 1212558348, 1930511492, 921147012, 1364134716,
  1878989525, 721655347, 10415448, 125429553, 1711864830, 1294311706,
  484985294, 1710229680, 1927940813, 156218122, 543966632, 1557837957,
  245772028, 1153814762, 739406449)

# Select 15 islands 100 times
sample.full.15 = matrix(nrow = 15, ncol = 100)

for (i in 1:100) {
  set.seed(seed.full.15[i])
  sample.full.15[, i] = data.frame(sar.isl %>% group_by(class) %>% sample_n(size = 3))$id
}

sample.full.15 = as.data.frame(sample.full.15) # convert the matrix into a data frame

# ISAR for each subset of 15 sites (n = 100)
slope.full.15 = r2.full.15 = p.full.15 = c()

```

```

for (i in 1:100) {
  slope.full.15[i] = lm(richness ~ log10(area),
                        # Regression slope
                        data = sar.isl[sample.full.15[, i], ])$coefficients[2]
  r2.full.15[i] = summary(lm(richness ~ log10(area),
                             # Model fit (r2)
                             data = sar.isl[sample.full.15[, i], ]))$adj.r.squared
  p.full.15[i] = summary(lm(richness ~ log10(area),
                            # p-value
                            data = sar.isl[sample.full.15[, i], ]))$coefficients[8]
}

# Store results in a data frame
sar.full.15 = data.frame(slope = slope.full.15,
                        r2 = r2.full.15, p = p.full.15, n.isl = 15)

#####
##### 10 #####
#####

# Generate and save 100 random numbers
# sample(1:Machine$integer.max, 100) # This command generated the following values
seed.full.10 = c(1836448795, 1602227588, 1258488269, 566114368, 1776020126,
  894391246, 939975163, 83309568, 507790578, 1992867076, 71644900, 1168011998,
  2207129, 617292461, 1404656791, 1491500252, 283127481, 1486520641,
  4151586, 350243671, 1600104000, 636115113, 2087365855, 944554374, 613956427,
  195147222, 1653167479, 203780141, 1135993341, 429403506, 1752383166,
  1965071688, 1334116186, 1253729813, 773317255, 2101729052, 2093757280,
  1932699623, 128678624, 1034608016, 2073892305, 121390581, 1391591900,
  747430867, 906679344, 435651795, 1774937385, 2029771941, 59277239,
  2128512526, 2008467108, 192065477, 853393523, 960774878, 1438757988,
  1804104337, 1527600430, 108492762, 1115622710, 289274517, 179332826,
  1910766374, 626250152, 1642845096, 502806666, 1562003476, 1781175387,
  809230528, 490314009, 1402021001, 368756102, 732395523, 84496965, 352270870,
  375811602, 1971730604, 146631313, 2060064172, 1198951728, 1563389336,
  1726472535, 1674206838, 475363694, 1517318119, 1865912176, 303053240,
  1361078265, 267293077, 171185485, 44395409, 143330223, 921602179, 2060765949,
  92596341, 2131167233, 683355650, 728572135, 987009280, 389601379, 1214888602)

# Select 10 islands 100 times
sample.full.10 = matrix(nrow = 10, ncol = 100)

for (i in 1:100) {
  set.seed(seed.full.10[i])
  sample.full.10[, i] = data.frame(sar.isl %>% group_by(class) %>% sample_n(size = 2))$id
}

sample.full.10 = as.data.frame(sample.full.10) # convert the matrix into a data frame

# ISAR for each subset of 10 sites (n = 100)
slope.full.10 = r2.full.10 = p.full.10 = c()

```

```

for (i in 1:100) {
  slope.full.10[i] = lm(richness ~ log10(area),
    # Regression slope
    data = sar.isl[sample.full.10[, i], ])$coefficients[2]
  r2.full.10[i] = summary(lm(richness ~ log10(area),
    # Model fit (r2)
    data = sar.isl[sample.full.10[, i], ]))$adj.r.squared
  p.full.10[i] = summary(lm(richness ~ log10(area),
    # p-value
    data = sar.isl[sample.full.10[, i], ]))$coefficients[8]
}

# Store results in a data frame
sar.full.10 = data.frame(slope = slope.full.10,
  r2 = r2.full.10, p = p.full.10, n.isl = 10)

#####
##### 5 #####
#####

# Generate and save 100 random numbers
# sample(1:Machine$integer.max, 100) # This command generated the following values
seed.full.5 = c(156379320, 1690806060, 1629423097, 1414352733, 1676991359,
  976730569, 1863137538, 1725172610, 1245716941, 2144377928, 1892644476,
  902300213, 744495280, 1713546279, 1774310168, 1851413986, 553780345,
  1405284858, 108015246, 1673947428, 2097155812, 1056236678, 1721697030,
  2049545505, 603117198, 947333476, 2122239900, 1440513547, 1927913765,
  454943762, 290176211, 1153586770, 1180513862, 1453353980, 1040440908,
  1536727835, 941783144, 11760277, 1505622231, 787314459, 1802471460,
  792349783, 608428158, 1074061213, 1162948024, 424745217, 1828967737,
  1108684111, 603935622, 1659622245, 68119010, 125836783, 756516015,
  265540766, 1278239777, 336513225, 1804560728, 2138761109, 478843279,
  951316917, 1179275132, 710946839, 1149044041, 887556158, 462340394,
  406182211, 2039798191, 1659142329, 178380253, 1751210418, 781822530,
  991432089, 1517513217, 252335960, 977309456, 264568238, 1563478621,
  1169537062, 1679809152, 7449048, 708562664, 1081425022, 842892613,
  1489221701, 965928975, 964844400, 503102680, 1179685632, 846189621,
  570467263, 813254137, 209997858, 1093310310, 723541863, 192976009,
  252496991, 1843360057, 493987252, 1851015650, 1690781569)

# Select 5 islands 100 times
sample.full.5 = matrix(nrow = 5, ncol = 100)

for (i in 1:100) {
  set.seed(seed.full.5[i])
  sample.full.5[, i] = data.frame(sar.isl %>% group_by(class) %>% sample_n(size = 1))$id
}

sample.full.5 = as.data.frame(sample.full.5) # convert the matrix into a data frame

# ISAR for each subset of 5 sites (n = 100)
slope.full.5 = r2.full.5 = p.full.5 = c()

```

```

for (i in 1:100) {
  slope.full.5[i] = lm(richness ~ log10(area),
                      # Regression slope
                      data = sar.isl[sample.full.5[, i], ])$coefficients[2]
  r2.full.5[i] = summary(lm(richness ~ log10(area),
                           # Model fit (r2)
                           data = sar.isl[sample.full.5[, i], ]))$adj.r.squared
  p.full.5[i] = summary(lm(richness ~ log10(area),
                           # p-value
                           data = sar.isl[sample.full.5[, i], ]))$coefficients[8]
}

# Store results in a data frame
sar.full.5 = data.frame(slope = slope.full.5,
                       r2 = r2.full.5, p = p.full.5, n.isl = 5)

#####
##### All #####
#####

# Combine the results
sar.full = rbind(sar.full.25, sar.full.20, sar.full.15, sar.full.10, sar.full.5)

# Indicate the direction of the relationship
sar.full$dir = NA
sar.full$dir[sar.full$p > 0.05] = "non-significant"
sar.full$dir[sar.full$p < 0.05 & sar.full$slope > 0] = "positive"
sar.full$dir[sar.full$p < 0.05 & sar.full$slope < 0] = "negative"

# Colour code
sar.full$col = NA
sar.full$col[sar.full$p > 0.05] = "black"
sar.full$col[sar.full$p < 0.05 & sar.full$slope > 0] = "dodgerblue"
sar.full$col[sar.full$p < 0.05 & sar.full$slope < 0] = "#e41a1c"

# Function to produce summary statistics (mean and +/- sd)
data_summary = function(z) {
  m <- mean(z)
  ymin <- m-sd(z)
  ymax <- m+sd(z)
  return(c(y = m, ymin = ymin, ymax = ymax))
}

```

Figure 4a

```

box.full.slope =
  ggplot(aes(x = as.factor(n.isl),
             y = (atan(sar.full$slope) * 180 / pi) /
                 (atan(sar.74$coefficients[2]) * 180 / pi)),
         data = sar.full) +

```

```

ggtitle("Full range of island size (0.45-1,699 ha)") +

labs(x = "",
     y = "Slope deviance",
     colour = expression(bold(Effect))) +

scale_y_continuous(limits = c(-1.2, 1.2)) +

scale_colour_manual(values = c("#999999", "dodgerblue")) +

geom_boxplot(fatten = NULL, outlier.shape = NA, notch = TRUE) +
geom_jitter(shape = 19, size = 2, aes(colour = sar.full$dir),
            alpha = 0.25, width = 0.25, height = 0) +

stat_summary(fun.data = data_summary, colour = "#e41a1c", size = 1) +

theme_bw(base_size = 20) +
  theme(panel.grid = element_blank(),
        panel.border = element_rect(colour = "black"),
        axis.title = element_text(colour = "black", face = "bold"),
        axis.text = element_text(colour = "black"),
        axis.ticks = element_line(colour = "black", size = 0.5)) +
theme(plot.title = element_text(size = 20, face = "bold", hjust = 0.5)) +
theme(legend.title = element_text(size = 16),
      legend.text = element_text(size = 14),
      legend.position = c(0.05, 0.05),
      legend.justification = c(0.05, 0.05),
      legend.background = element_rect(colour = NULL),
      legend.key = element_rect(fill = NA)) +
guides(colour = guide_legend(override.aes = list(size = 5),
                             reverse = TRUE, order = 1)) +

annotate("text", x = "5", y = 1.2,
         hjust = 1.2, vjust = 0, size = 6, colour = "dodgerblue",
         label = table(subset(sar.full, sar.full$n.isl == 5)$dir)[2]) +
annotate("text", x = "5", y = 1.2,
         hjust = -0.2, vjust = 0, size = 6, colour = "black",
         label = table(subset(sar.full, sar.full$n.isl == 5)$dir)[1]) +

annotate("text", x = "10", y = 1.2,
         hjust = 1.2, vjust = 0, size = 6, colour = "dodgerblue",
         label = table(subset(sar.full, sar.full$n.isl == 10)$dir)[2]) +
annotate("text", x = "10", y = 1.2,
         hjust = -0.2, vjust = 0, size = 6, colour = "black",
         label = table(subset(sar.full, sar.full$n.isl == 10)$dir)[1]) +

annotate("text", x = "15", y = 1.2,
         hjust = 0.5, vjust = 0, size = 6, colour = "dodgerblue",
         label = table(subset(sar.full, sar.full$n.isl == 15)$dir)[1]) +

annotate("text", x = "20", y = 1.2,
         hjust = 0.5, vjust = 0, size = 6, colour = "dodgerblue",
         label = table(subset(sar.full, sar.full$n.isl == 20)$dir)[1]) +

```

```

annotate("text", x = "25", y = 1.2,
         hjust = 0.5, vjust = 0, size = 6, colour = "dodgerblue",
         label = table(subset(sar.full, sar.full$n.isl == 25)$dir)[1]) +
annotate("text", x = Inf, y = -Inf,
         hjust = 1.3, vjust = -1, fontface = "bold", size = 10, label = "(a)")

```

#box.full.slope

Figure 4c

```

box.full.r2 =
  ggplot(aes(x = as.factor(n.isl),
            y = r2),
        data = sar.full) +

  ggtitle("") +

  labs(x = "Number of islands",
       y = (expression(bolditalic(r)^{2}*"[adj]*")) +
  scale_y_continuous(limits = c(-0.35, 1),
                    breaks = seq(-0.2, 1, 0.2)) +

  geom_boxplot(fatten = NULL, outlier.shape = NA, notch = TRUE) +
  geom_jitter(shape = 19, size = 2, colour = sar.full$col,
             alpha = 0.25, width = 0.25, height = 0) +

  stat_summary(fun.data = data_summary, colour = "#e41a1c", size = 1) +

  theme_bw(base_size = 20) +
  theme(panel.grid = element_blank(),
        panel.border = element_rect(colour = "black"),
        axis.title = element_text(colour = "black", face = "bold"),
        axis.text = element_text(colour = "black"),
        axis.ticks = element_line(colour = "black", size = 0.5)) +
  theme(plot.title = element_text(hjust = 0.5)) +

  annotate("text", x = Inf, y = -Inf,
         hjust = 1.3, vjust = -1, fontface = "bold", size = 10, label = "(c)")

```

#box.full.r2

Non-stratified random selection (short range)

```

#####
##### 25 #####
#####

# Generate and save 100 random numbers
# sample(1:Machine$integer.max, 100) # This command generated the following values
seed.short.25 = c(546470929, 1854569192, 472457862, 1519942303, 895556555,

```

```

1919452227, 1930033853, 2063048861, 895797531, 671426628, 689697640,
293680556, 1933282389, 1821079978, 1498918864, 1394247354, 153968628,
271206201, 92592477, 2081273213, 188826510, 698750039, 947001660, 1773674421,
1410373373, 2104025889, 1143083811, 1033814086, 1769894669, 872329229,
1163073271, 1148528459, 560434502, 1224909623, 1031145262, 582656922,
308392546, 67389807, 263459132, 391259468, 1576323070, 1212875871,
2070690239, 633301053, 1672178552, 2073513731, 56937991, 1549026011,
1198479649, 525813008, 151116584, 197262112, 1051865805, 1207593570,
452395609, 728516306, 366933408, 1457426627, 1353714277, 2100197408,
1818641172, 1132032800, 2046009994, 1110071781, 623413129, 985610257,
344982842, 1741372437, 213322852, 132985715, 319609380, 1389669427,
1940295069, 1497755854, 758207049, 1601760415, 2143143284, 1661429508,
1548933504, 1475819423, 955483790, 3214408, 622301579, 235955175, 741148058,
637739887, 1904814664, 330630631, 687650001, 1881465793, 1563203045,
730846447, 916417468, 88085882, 1532756237, 1813116892, 1486690846,
611013103, 21024891, 1152909322)

# Select 25 islands 100 times
sample.short.25 = matrix(nrow = 25, ncol = 100)

for (i in 1:100) {
  set.seed(seed.short.25[i])
  sample.short.25[, i] = sample(subset(sar.isl, sar.isl$area < 100)$id, 25)
}

sample.short.25 = as.data.frame(sample.short.25) # convert the matrix into a data frame

# ISAR for each subset of 25 sites (n = 100)
slope.short.25 = r2.short.25 = p.short.25 = c()

for (i in 1:100) {
  slope.short.25[i] = lm(richness ~ log10(area),
                        # Regression slope
                        data = sar.isl[sample.short.25[, i], ])$coefficients[2]
  r2.short.25[i] = summary(lm(richness ~ log10(area),
                             # Model fit (r2)
                             data = sar.isl[sample.short.25[, i], ]))$adj.r.squared
  p.short.25[i] = summary(lm(richness ~ log10(area),
                             # p-value
                             data = sar.isl[sample.short.25[, i], ]))$coefficients[8]
}

# Store results in a data frame
sar.short.25 = data.frame(slope = slope.short.25,
                          r2 = r2.short.25, p = p.short.25, n.isl = 25)

#####
##### 20 #####
#####

# Generate and save 100 random numbers
# sample(1:.Machine$integer.max, 100) # This command generated the following values

```

```

seed.short.20 = c(666307173, 769548065, 537021976, 279055760, 62172685,
1367337426, 957009295, 1989292791, 1853230961, 1911443969, 1261353721,
1027415282, 241034004, 161718608, 1275703625, 2111439781, 1700598712,
1592916179, 43374530, 232151158, 209340666, 1085935855, 285882049,
2091206380, 1682878078, 859603848, 1697635446, 1635861806, 1022456555,
2024697679, 2101397481, 1344500639, 1822426816, 1311649218, 543324895,
2087581645, 526123100, 1680050699, 566775562, 345167024, 1602664518,
64968048, 1639855157, 1241354938, 1485552218, 1861526072, 612335669,
1633123308, 1459100985, 1158121107, 580497679, 306604827, 1332928838,
286243345, 2141373777, 2037212311, 1506619446, 521308150, 1420686872,
338641856, 260635652, 1252933919, 15446332, 2052072612, 241882135,
807303067, 393868183, 1032293094, 1926166158, 909554276, 1742475855,
663146752, 1736829215, 423988712, 180014558, 844018892, 1480232952,
1405965368, 313990093, 1332559669, 347747447, 436081704, 1530297586,
447891509, 1992335844, 881663723, 533522389, 1968307086, 719892721,
87469310, 1778822274, 1736369187, 2082516166, 979182080, 1413199227,
489656294, 49916018, 1372516049, 1458701386, 302606507)

# Select 20 islands 100 times
sample.short.20 = matrix(nrow = 20, ncol = 100)

for (i in 1:100) {
  set.seed(seed.short.20[i])
  sample.short.20[, i] = sample(subset(sar.isl, sar.isl$area < 100)$id, 20)
}

sample.short.20 = as.data.frame(sample.short.20) # convert the matrix into a data frame

# ISAR for each subset of 20 sites (n = 100)
slope.short.20 = r2.short.20 = p.short.20 = c()

for (i in 1:100) {
  slope.short.20[i] = lm(richness ~ log10(area),
                        # Regression slope
                        data = sar.isl[sample.short.20[, i], ])$coefficients[2]
  r2.short.20[i] = summary(lm(richness ~ log10(area),
                            # Model fit (r2)
                            data = sar.isl[sample.short.20[, i], ]))$adj.r.squared
  p.short.20[i] = summary(lm(richness ~ log10(area),
                            # p-value
                            data = sar.isl[sample.short.20[, i], ]))$coefficients[8]
}

# Store results in a data frame
sar.short.20 = data.frame(slope = slope.short.20,
                        r2 = r2.short.20, p = p.short.20, n.isl = 20)

#####
##### 15 #####
#####

# Generate and save 100 random numbers

```



```

# sample(1:Machine$integer.max, 100) # This command generated the following values
seed.short.15 = c(1797233886, 1477817566, 1812472833, 282658037, 159644218,
  1058820179, 30092270, 1430848344, 1802764155, 1546478896, 498495623,
  1573434690, 450846291, 1393940053, 653030952, 261493363, 1890241833,
  982501422, 1746246577, 1405205347, 1995988109, 120882561, 1747207129,
  327387968, 2020831209, 1331284056, 1745849496, 1292633655, 203322744,
  41782417, 37050654, 2067894056, 635559297, 479796905, 145114616, 291741248,
  184365873, 71118960, 2124533111, 534668116, 1357261360, 1113033225,
  1044605132, 813411959, 1743315622, 1255166571, 341128038, 781834300,
  233647969, 1726759231, 706151582, 1130525245, 1875316554, 1415337554,
  137326230, 217638676, 1906033574, 2146460365, 466812727, 113915874,
  24889785, 1401542266, 660202450, 314097483, 779164117, 1082807616,
  1390074298, 1416012482, 1012072300, 2025426536, 536217283, 2101306655,
  18189273, 1811934711, 1312938722, 854283991, 1818428581, 505361758,
  240179728, 935751634, 1969290289, 840928766, 781350302, 698273657,
  278682066, 1923288556, 200496280, 8906851, 1251617670, 419489047, 707522070,
  2117554166, 94965522, 1283543567, 1167658624, 160287416, 487054397,
  1478979793, 28832598, 704526487)

# Select 15 islands 100 times
sample.short.15 = matrix(nrow = 15, ncol = 100)

for (i in 1:100) {
  set.seed(seed.short.15[i])
  sample.short.15[, i] = sample(subset(sar.isl, sar.isl$area < 100)$id, 15)
}

sample.short.15 = as.data.frame(sample.short.15) # convert the matrix into a data frame

# ISAR for each subset of 15 sites (n = 100)
slope.short.15 = r2.short.15 = p.short.15 = c()

for (i in 1:100) {
  slope.short.15[i] = lm(richness ~ log10(area),
    # Regression slope
    data = sar.isl[sample.short.15[, i], ])$coefficients[2]
  r2.short.15[i] = summary(lm(richness ~ log10(area),
    # Model fit (r2)
    data = sar.isl[sample.short.15[, i], ]))$adj.r.squared
  p.short.15[i] = summary(lm(richness ~ log10(area),
    # p-value
    data = sar.isl[sample.short.15[, i], ]))$coefficients[8]
}

# Store results in a data frame
sar.short.15 = data.frame(slope = slope.short.15,
  r2 = r2.short.15, p = p.short.15, n.isl = 15)

#####
##### 10 #####
#####

```

```

# Generate and save 100 random numbers
# sample(1:Machine$integer.max, 100) # This command generated the following values
seed.short.10 = c(1490864286, 2047178273, 1728145608, 281818095, 2087499922,
  564366804, 224610476, 1411050063, 281673670, 49572642, 1132043984,
  555201282, 716813516, 1280508556, 1589730837, 1534601435, 146351227,
  514405340, 1698506167, 1554576283, 318444025, 2093255895, 783318015,
  2013258027, 168402407, 576605944, 657233474, 1584910838, 1318819952,
  1820689937, 1249616156, 904786729, 1332431854, 812038017, 1885297493,
  1187631930, 928731614, 1051752178, 1138261024, 108586588, 1343434796,
  839610588, 2063078918, 1104421074, 759849601, 1713214309, 1858689334,
  450407697, 313256126, 1457307133, 1009155087, 505753525, 335191415,
  848433713, 1526577160, 2093789644, 386663266, 684043014, 2010058334,
  767706419, 1800026661, 735429860, 552159751, 1774679058, 2071490816,
  114035331, 1362076937, 452982224, 88437574, 824344601, 1771460610,
  85404261, 1045361323, 133918837, 1955817682, 1861748225, 1024056941,
  1068930095, 855334022, 618819368, 282124553, 131763589, 1642882943,
  1585028290, 1376992579, 133003909, 1692076196, 258150875, 791640343,
  392187150, 828995879, 761964529, 1665508994, 720215031, 1482990698,
  774879948, 1187796754, 288976470, 925947713, 1968126902)

# Select 10 islands 100 times
sample.short.10 = matrix(nrow = 10, ncol = 100)

for (i in 1:100) {
  set.seed(seed.short.10[i])
  sample.short.10[, i] = sample(subset(sar.isl, sar.isl$area < 100)$id, 10)
}

sample.short.10 = as.data.frame(sample.short.10) # convert the matrix into a data frame

# ISAR for each subset of 10 sites (n = 100)
slope.short.10 = r2.short.10 = p.short.10 = c()

for (i in 1:100) {
  slope.short.10[i] = lm(richness ~ log10(area),
    # Regression slope
    data = sar.isl[sample.short.10[, i], ])$coefficients[2]
  r2.short.10[i] = summary(lm(richness ~ log10(area),
    # Model fit (r2)
    data = sar.isl[sample.short.10[, i], ]))$adj.r.squared
  p.short.10[i] = summary(lm(richness ~ log10(area),
    # p-value
    data = sar.isl[sample.short.10[, i], ]))$coefficients[8]
}

# Store results in a data frame
sar.short.10 = data.frame(slope = slope.short.10,
  r2 = r2.short.10, p = p.short.10, n.isl = 10)

#####
##### 5 #####
#####

```

```

# Generate and save 100 random numbers
# sample(1:Machine$integer.max, 100) # This command generated the following values
seed.short.5 = c(1818899806, 572588167, 1792177407, 1659406357, 961103864,
  1530683178, 1134007730, 937830870, 1066204421, 1804204256, 503063391,
  1748500366, 474255682, 1214858577, 1348666140, 852326043, 1085653185,
  1412958084, 337382050, 1819034604, 347289743, 1832647990, 185514679,
  1223109694, 1973347031, 1800567392, 552920802, 830448278, 373898856,
  424348329, 235826459, 1805765409, 2046082354, 746323328, 1748767641,
  1189739601, 923159733, 1850158210, 1356612017, 1053329651, 1474626586,
  1300879879, 2122733924, 2083408007, 536239387, 706011796, 1637424859,
  657349614, 11260282, 17681210, 342174539, 1576634915, 1796853954, 1786492628,
  495993392, 896512922, 676915201, 532818466, 826704236, 662021449, 954280518,
  262499102, 1067107507, 318192681, 674273262, 1473103292, 481573782,
  2077582954, 1927740882, 801457600, 1996286074, 1030628649, 408608466,
  706757404, 927495176, 296551639, 875969451, 725602398, 265102155, 514037893,
  948860582, 758135234, 1709858334, 1720550012, 847904281, 842872595,
  620195336, 1792612878, 286096883, 490370245, 2094825162, 2078260721,
  1757203739, 1944745230, 1532469947, 1698773672, 753595436, 437582555,
  1745664625, 867798478)

# Select 5 islands 100 times
sample.short.5 = matrix(nrow = 5, ncol = 100)

for (i in 1:100) {
  set.seed(seed.short.5[i])
  sample.short.5[, i] = sample(subset(sar.isl, sar.isl$area < 100)$id, 5)
}

sample.short.5 = as.data.frame(sample.short.5) # convert the matrix into a data frame

# ISAR for each subset of 5 sites (n = 100)
slope.short.5 = r2.short.5 = p.short.5 = c()

for (i in 1:100) {
  slope.short.5[i] = lm(richness ~ log10(area),
    # Regression slope
    data = sar.isl[sample.short.5[, i], ])$coefficients[2]
  r2.short.5[i] = summary(lm(richness ~ log10(area),
    # Model fit (r2)
    data = sar.isl[sample.short.5[, i], ]))$adj.r.squared
  p.short.5[i] = summary(lm(richness ~ log10(area),
    # p-value
    data = sar.isl[sample.short.5[, i], ]))$coefficients[8]
}

# Store results in a data frame
sar.short.5 = data.frame(slope = slope.short.5,
  r2 = r2.short.5, p = p.short.5, n.isl = 5)

#####
##### All #####
#####

```

```

# Combine the results
sar.short = rbind(sar.short.25, sar.short.20, sar.short.15, sar.short.10, sar.short.5)

# Indicate the direction of the relationship
sar.short$dir = NA
sar.short$dir[sar.short$p > 0.05] = "non-significant"
sar.short$dir[sar.short$p < 0.05 & sar.short$slope > 0] = "positive"
sar.short$dir[sar.short$p < 0.05 & sar.short$slope < 0] = "negative"

# Colour code
sar.short$col = NA
sar.short$col[sar.short$p > 0.05] = "black"
sar.short$col[sar.short$p < 0.05 & sar.short$slope > 0] = "dodgerblue"
sar.short$col[sar.short$p < 0.05 & sar.short$slope < 0] = "#e41a1c"

```

Figure 4b

```

box.short.slope =
  ggplot(aes(x = as.factor(n.isl),
             y = (atan(sar.short$slope) * 180 / pi) /
                 (atan(sar.74$coefficients[2]) * 180 / pi)),
         data = sar.short) +

  ggtitle("Short range of island size (< 100 ha)") +

  labs(x = "",
       y = "") +

  scale_y_continuous(limits = c(-1.2, 1.2)) +

  geom_boxplot(fatten = NULL, outlier.shape = NA, notch = TRUE) +
  geom_jitter(shape = 19, size = 2, colour = sar.short$col,
             alpha = 0.25, width = 0.25, height = 0) +

  stat_summary(fun.data = data_summary, colour = "#e41a1c", size = 1) +

  theme_bw(base_size = 20) +
  theme(panel.grid = element_blank(),
        panel.border = element_rect(colour = "black"),
        axis.title = element_text(colour = "black", face = "bold"),
        axis.text = element_text(colour = "black"),
        axis.ticks = element_line(colour = "black", size = 0.5)) +
  theme(plot.title = element_text(size = 20, face = "bold", hjust = 0.5)) +

  annotate("text", x = "5", y = 1.2,
         hjust = 1.2, vjust = -0, size = 6, colour = "dodgerblue",
         label = table(subset(sar.short, sar.short$n.isl == 5)$dir)[2]) +
  annotate("text", x = "5", y = 1.2,
         hjust = -0.2, vjust = 0, size = 6, colour = "black",
         label = table(subset(sar.short, sar.short$n.isl == 5)$dir)[1]) +

  annotate("text", x = "10", y = 1.2,

```

```

      hjust = 1.2, vjust = 0, size = 6, colour = "dodgerblue",
      label = table(subset(sar.short, sar.short$n.isl == 10)$dir)[2]) +
annotate("text", x = "10", y = 1.2,
      hjust = -0.2, vjust = 0, size = 6, colour = "black",
      label = table(subset(sar.short, sar.short$n.isl == 10)$dir)[1]) +

annotate("text", x = "15", y = 1.2,
      hjust = 1.2, vjust = 0, size = 6, colour = "dodgerblue",
      label = table(subset(sar.short, sar.short$n.isl == 15)$dir)[2]) +
annotate("text", x = "15", y = 1.2,
      hjust = -0.2, vjust = 0, size = 6, colour = "black",
      label = table(subset(sar.short, sar.short$n.isl == 15)$dir)[1]) +

annotate("text", x = "20", y = 1.2,
      hjust = 1.2, vjust = 0, size = 6, colour = "dodgerblue",
      label = table(subset(sar.short, sar.short$n.isl == 20)$dir)[2]) +
annotate("text", x = "20", y = 1.2,
      hjust = -0.2, vjust = 0, size = 6, colour = "black",
      label = table(subset(sar.short, sar.short$n.isl == 20)$dir)[1]) +

annotate("text", x = "25", y = 1.2,
      hjust = 1.2, vjust = 0, size = 6, colour = "dodgerblue",
      label = table(subset(sar.short, sar.short$n.isl == 25)$dir)[2]) +
annotate("text", x = "25", y = 1.2,
      hjust = -0.2, vjust = 0, size = 6, colour = "black",
      label = table(subset(sar.short, sar.short$n.isl == 25)$dir)[1]) +
annotate("text", x = Inf, y = -Inf,
      hjust = 1.3, vjust = -1, fontface = "bold", size = 10, label = "(b)")

#box.short.slope

```

Figure 4d

```

box.short.r2 =
  ggplot(aes(x = as.factor(n.isl),
            y = r2),
        data = sar.short) +

  ggtitle("") +

  labs(x = "Number of islands",
       y = "") +

  scale_y_continuous(limits = c(-0.35, 1),
                    breaks = seq(-0.2, 1, 0.2)) +

  geom_boxplot(fatten = NULL, outlier.shape = NA, notch = TRUE) +
  geom_jitter(shape = 19, size = 2, colour = sar.short$col,
             alpha = 0.25, width = 0.25, height = 0) +

  stat_summary(fun.data = data_summary, colour = "#e41a1c", size = 1) +

```

```

theme_bw(base_size = 20) +
  theme(panel.grid = element_blank(),
        panel.border = element_rect(colour = "black"),
        axis.title = element_text(colour = "black", face = "bold"),
        axis.text = element_text(colour = "black"),
        axis.ticks = element_line(colour = "black", size = 0.5)) +
  theme(plot.title = element_text(hjust = 0.5)) +

  annotate("text", x = Inf, y = -Inf,
          hjust = 1.3, vjust = -1, fontface = "bold", size = 10, label = "(d)")

#box.short.r2

```

Combine graphs

Figure 4

```

# Combine and save the graphs
ggsave(grid.arrange(box.full.slope, box.short.slope, box.full.r2, box.short.r2,
                    nrow = 2, ncol = 2),
        filename = "fig4.pdf", width = 17*2, height = 17*2, units = "cm")

```

Literature review

We carried out a literature review focused on tropical and temperate frog studies worldwide to assess (1) how prevalent positive ISARs are at a global scale, and (2) the role of the number of patches and range in patch size in detecting ISARs.

Our literature review involved four steps as follows:

1. We searched for data (patch size and observed number of frog species) in all studies listed in Table 1 in Almeida-Gomes, Vieira, Rocha, Metzger, & De Coster (2016) and in Table S1 in Palmeirim, Vieira, & Peres (2017).
2. We updated the search carried out by Almeida-Gomes, Vieira, Rocha, Metzger, & De Coster (2016) in *Web of Science* to include additional studies published since their compilation cut-off in 2015. We searched for the same keywords they used – (fragment size* AND amphibia*) OR (fragment size* AND anura*) OR (fragment size* AND frog*) OR (patch size* AND amphibia*) OR (patch size* AND anura*) OR (patch size* AND frog*). These search terms (in *Topic* on 29 August 2018) resulted in 101 hits.
3. We searched for the keywords (("species-area relation*" OR "species-area curve*") AND (amphibia* OR anura* OR frog*) OR (nestedness AND amphibia*) OR (nestedness AND anura*) OR (nestedness AND frog*)) OR (("nested subset*" AND amphibia*) OR ("nested subset*" AND anura*) OR ("nested subset*" AND frog*) OR ("nested species" AND amphibia*) OR ("nested species" AND anura*) OR ("nested species" AND frog*)) in *Topic* using the *Web of Science* database on 29 August 2018 (*Timespan: All years*). This search resulted in 109 hits.
4. We also screened the titles in the reference list of two recent global syntheses on species-area relationships (Matthews, Guilhaumon, Triantis, Borregaard, & Whittaker, 2016) and vertebrate species responses to habitat fragmentation (Keinath et al., 2017).

Island species-area relationships across studies

Despite of being an expected pattern, the positive island species-area relationship (ISAR) has not held true in some studies (for amphibians, see Almeida-Gomes, Vieira, Rocha, Metzger, & De Coster (2016); for reptiles, see Lion, Garda, Santana, & Fonseca

(2016)). Different than these authors, who presented a summary of the results (negative, non-significant, positive) reported in the previous studies, we reanalysed data from the literature using the logarithmic exponential equation (semi-log model) proposed by Gleason (1922).

$$S = z \times \log_{10}(A) + c$$

where S = number of species, z = regression slope, A = site area (ha), c = regression intercept.

We used the semi-log model to depict the species-area relationships because it allows the inclusion of sites with $S = 0$.

Import data compiled from the literature

```
# Study attributes
studies = read.csv("https://ndownloader.figshare.com/files/15158570")

# Import data of each study (site area and species richness)
sar.lit = read.csv("https://ndownloader.figshare.com/files/15158567")

# Create a list object to store each study separately
sar.list = list()

for (i in 1:length(unique(sar.lit$studyID))) { # number of studies
  sar.list[[i]] = subset(sar.lit, sar.lit$studyID == unique(sar.lit$studyID)[[i]])
}
```

Location of the studies

Figure 5

```
# Draw the map
map =
  ggplot() +

  labs(x = "Longitude", y = "Latitude") +

  borders("world", colour = "#808080", fill = "#808080") +

  geom_point(data = studies,
             aes(x = longitude, y = latitude),
             shape = 21, colour = "black", fill = "dodgerblue", size = 4)

# Save the map
ggsave(map, file = "fig5.pdf", width = 30, height = 20, units = "cm")

# Display the map
#map
```

ISAR – semi-log models

```
# Fit the semi-log model to each dataset and save the results
semilog.out = matrix(nrow = length(sar.list), ncol = 9)
```

```

for (i in 1:length(sar.list)) {
  semilog.out[i, 1] = unique(sar.list[[i]]$studyID)
  semilog.out[i, 2] = as.character(unique(sar.list[[i]]$reference))
  semilog.out[i, 3] = lm(richness ~ log10(area),
    data = sar.list[[i]]$coefficients[1] # regression intercept
  semilog.out[i, 4] = lm(richness ~ log10(area),
    data = sar.list[[i]]$coefficients[2] # regression slope
  semilog.out[i, 5] = summary(lm(richness ~ log10(area),
    data = sar.list[[i]]))$adj.r.squared # model fit (r2)
  semilog.out[i, 6] = summary(lm(richness ~ log10(area),
    data = sar.list[[i]]))$coefficients[8] # p-value
  semilog.out[i, 7] = nrow(sar.list[[i]])
  semilog.out[i, 8] = max(sar.list[[i]]$area) - min(sar.list[[i]]$area)
  semilog.out[i, 9] = max(sar.list[[i]]$richness)
}

# Finish the data frame
# Name columns
colnames(semilog.out) = c("studyID", "reference", "intercept", "slope", "r2", "p",
  "n", "range", "smax")
# Convert the matrix "semilog.out" into a data frame
semilog.out = as.data.frame(semilog.out)

# Convert factor data into numeric data
semilog.out$intercept = as.numeric(as.character(semilog.out$intercept))
# Convert factor data into numeric data
semilog.out$slope = as.numeric(as.character(semilog.out$slope))
# Convert factor data into numeric data
semilog.out$r2 = as.numeric(as.character(semilog.out$r2))
# Convert factor data into numeric data
semilog.out$p = as.numeric(as.character(semilog.out$p))
# Convert factor data into integer data
semilog.out$n = as.integer(as.character(semilog.out$n))
# Convert factor data into numeric data
semilog.out$range = as.numeric(as.character(semilog.out$range))
# Convert factor data into integer data
semilog.out$smax = as.integer(as.character(semilog.out$smax))

# Indicate the direction of the relationship
semilog.out$dir = NA
semilog.out$dir[semilog.out$p > 0.05] = "non-significant"
semilog.out$dir[semilog.out$p < 0.05 & semilog.out$slope > 0] = "positive"
semilog.out$dir[semilog.out$p < 0.05 & semilog.out$slope < 0] = "negative"

```

Figure 6

```

# Draw the graph summarising the results across all 25 datasets
# (number of patches vs. range in size)
graph.semilog =
  ggplot(aes(x = range, y = n,
    fill = as.factor(dir), size = r2),
    data = semilog.out) +

```



```

labs(x = "Range in patch size (ha)",
     y = "Number of patches",
     fill = expression(bold(Effect)),
     size = expression(bolditalic(r)^{2}*"[adj]")) +

scale_x_log10(breaks = c(1, 10, 100, 300, 1000, 10000, 100000),
              labels = c("1", "10", "100", "300", "1,000", "10,000", "100,000")) +
scale_y_continuous(limits = c(5, 25)) +
annotation_logticks(base = 10, sides = "b") +

scale_fill_manual(values = c("#999999", "dodgerblue")) +
scale_size_continuous(range = c(3, 11)) +

geom_vline(xintercept = 300, size = 0.5, alpha = 0.2, linetype = "dashed") +
geom_hline(yintercept = 15, size = 0.5, alpha = 0.2, linetype = "dashed") +
geom_point(shape = 21, colour = "black") +

theme_bw(base_size = 20) +
theme(panel.grid = element_blank(),
      panel.border = element_rect(colour = "black"),
      axis.title = element_text(colour = "black", face = "bold"),
      axis.text = element_text(colour = "black"),
      axis.ticks = element_line(colour = "black", size = 0.5)) +
theme(legend.title = element_text(size = 16),
      legend.text = element_text(size = 14),
      legend.justification = "top",
      legend.background = element_rect(colour = NULL),
      legend.key = element_rect(fill = NA)) +
guides(fill = guide_legend(override.aes = list(size = 5),
                           reverse = TRUE, order = 1)) +

annotate("text", x = 7.82, y = 25,
         hjust = 0.05, vjust = 0, size = 4.5, label = "Many patches, narrow range") +
annotate("text", x = 7.82, y = 14,
         hjust = 0.05, vjust = 0, size = 4.5, label = "Few patches, narrow range") +
annotate("text", x = 400, y = 25,
         hjust = 0, vjust = 0, size = 4.5, label = "Many patches, broad range") +
annotate("text", x = 400, y = 14,
         hjust = 0, vjust = 0, size = 4.5, label = "Few patches, broad range")

# Save the graph
ggsave(graph.semilog, file = "fig6.pdf", width = 25, height = 20, units = "cm")

# Display the graph
#graph.semilog

```